Received: 8 December 2023 | Revised: 5 July 2024 | Accepted: 25 July 2024

DOI: 10.1111/jmi.13349

Journal of Microscopy RMS

THEMED ISSUE ARTICLE

Annotation and automated segmentation of single-molecule localisation microscopy data

Oliver Umney[1] | Joanna Leng[1] | Gianluca Canettieri[2,3] | Natalia A. Riobo-Del Galdo[4,5,6] | Hayley Slaney[7] | Philip Quirke[7] | Michelle Peckham[4,6] | Alistair Curd[7,8]

[1]Faculty of Engineering and Physical Sciences, School of Computing, University of Leeds, Leeds, UK

[2]Department of Molecular Medicine, Sapienza University of Rome, Rome, Italy

[3]Institute Pasteur Italy – Cenci Bolognetti Foundation, Sapienza University of Rome, Rome, Italy

[4]Faculty of Biological Sciences, School of Molecular and Cellular Biology, University of Leeds, Leeds, UK

[5]School of Medicine, Leeds Institute for Medical Research, University of Leeds, Leeds, UK

[6]Astbury Centre for Structural and Molecular Biology, University of Leeds, Leeds, UK

[7]Division of Pathology and Data Analytics, Leeds Institute of Medical Research at St James's, University of Leeds, Leeds, UK

[8]Faculty of Engineering and Physical Sciences, School of Physics, University of Leeds, Leeds, UK

Correspondence
Michelle Peckham, Faculty of Biological Sciences, School of Molecular and Cellular Biology, University of Leeds, UK.
Email: m.peckham@leeds.ac.uk

Funding information
UK Engineering and Physical Sciences Research Council (EPSRC), Grant/Award Number: EP/S024336/1; National Institute for Health and Care Research (NIHR) i4i grant, Grant/Award Number: NIHR201643; Wellcome Trust Investigator, Grant/Award Number: 223125/Z/21/Z; Biotechnology and Biological Sciences Research Council, Grant/Award Number: BB/S01716X/1; Fondazione Italiana per la Ricerca sul Cancro (AIRC), Grant/Award Number: IG 25833; EPSRC, Grant/Award Number: EP/R025819/1

Abstract

Single Molecule Localisation Microscopy (SMLM) is becoming a widely used technique in cell biology. After processing the images, the molecular localisations are typically stored in a table as *xy* (or *xyz*) coordinates, with additional information, such as number of photons, etc. This set of coordinates can be used to generate an image to visualise the molecular distribution, for example, a 2D or 3D histogram of localisations. Many different methods have been devised to analyse SMLM data, among which cluster analysis of the localisations is popular. However, it can be useful to first segment the data, to extract the localisations in a specific region of a cell or in individual cells, prior to downstream analysis. Here we describe a pipeline for annotating localisations in an SMLM dataset in which we compared membrane segmentation approaches, including Otsu thresholding and machine learning models, and subsequent cell segmentation. We used an SMLM dataset derived from dSTORM images of sectioned cell pellets, stained for the membrane proteins EGFR (epidermal growth factor receptor) and EREG (epiregulin) as a test dataset. We found that a Cellpose model retrained on our data performed the best in the membrane segmentation task, allowing us to perform downstream cluster analysis of membrane versus cell interior localisations. We anticipate this will be generally useful for SMLM analysis.

KEYWORDS
deep-learning, dSTORM, segmentation, SMLM

1 | INTRODUCTION

The initial development of superresolution fluorescence imaging included a number of single molecule localisation microscopy (SMLM) approaches such as photoactivated localisation microscopy (PALM[1]), stochastic optical reconstruction microscopy (STORM[2]) and dSTORM (direct optical reconstruction microscopy).[3,4] A range of superresolution approaches since developed all use the basic principle of 'blinking' fluorophores, in which a subset of fluorophores in the sample fluoresce briefly at any one time, and the positions of the fluorophores can be precisely determined to nanometre precision (reviewed in Refs. 5 and 6). In turn, this has led to a wide range of methods to quantitatively assess SMLM data (reviewed in Ref. 7) as well as the development of deep learning approaches to improve the rate and accuracy of SMLM imaging.[8–10]

To analyse the organisation of protein complexes in SMLM data, a useful first step is to segment the dataset. Segmentation of an image extracts the boundaries of an object, such as the nucleus or plasma membrane, and allows the analysis of that specific region.[11] Segmentation of an SMLM dataset labels each molecular localisation (a data point with *xyz* position, channel identifier, other properties) as belonging to a particular target structure (nucleus, etc.) or not. This enables the nanoscale analysis of protein organisation in those structures, vital for understanding sub-cellular structures and function.

Several methods have been developed for image segmentation, such as U-Net,[12,13] Cellpose,[14] reviewed in Ref. (11), and Ilastik.[15] U-Net is a widely used neural network developed for biomedical segmentation, with many pretrained models available to nonexperts through plug-and-play style interfaces. Cellpose provides U-Net style models for cell segmentation and is pretrained on a very large and diverse set of cell images that have been manually annotated, avoiding the need for retraining.[14] Ilastik is a semi-automated machine learning software that provides a GUI (graphic user interface) with access to multiple workflows including segmentation.[15] However, as image processing methods, these approaches cannot currently be used directly on an SMLM dataset (point cloud). In particular, although SMLM data can be rendered as an image, these methods cannot output labelled localisations for downstream analysis of segmented sets of localisations.

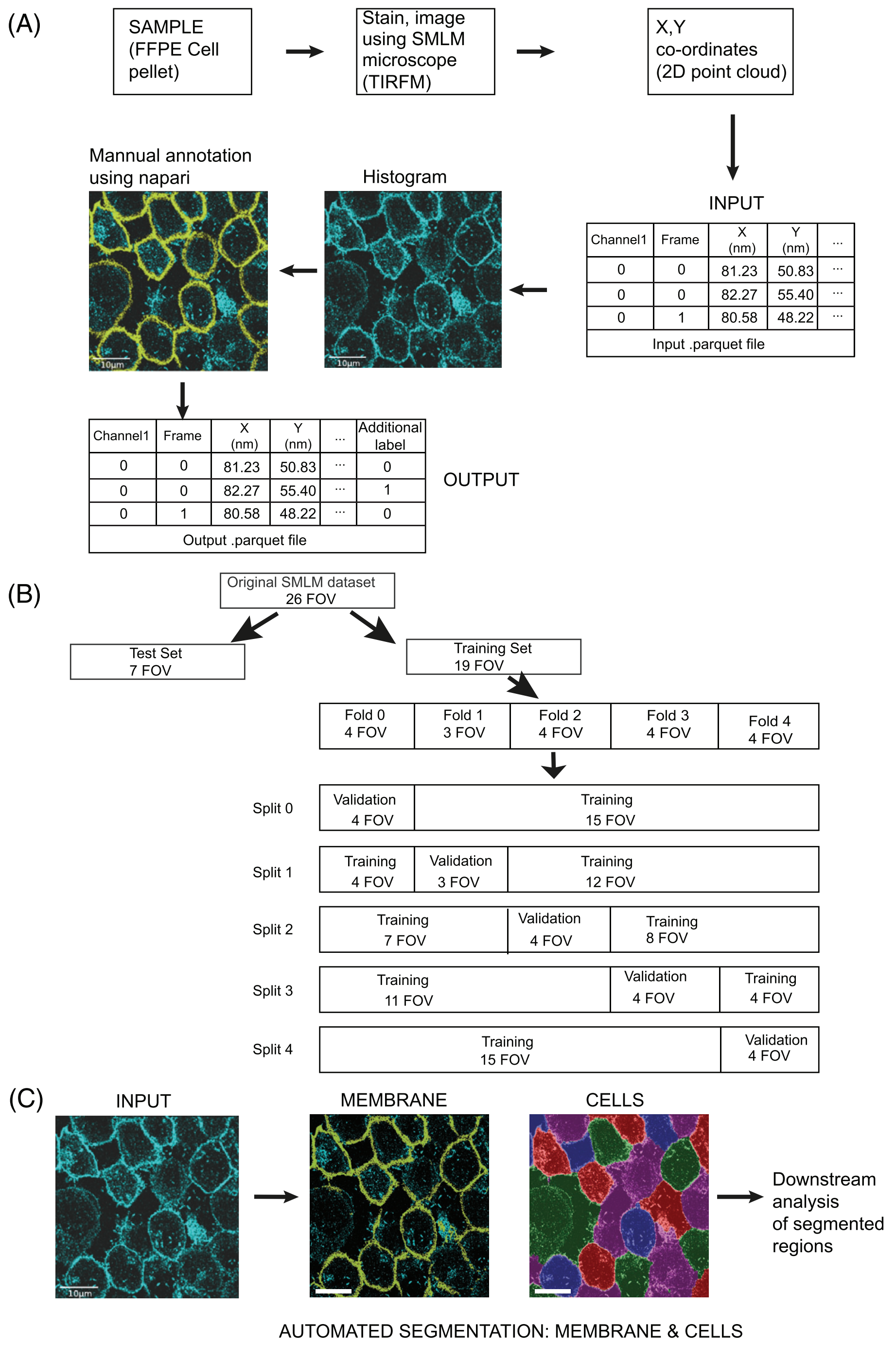
Methods for segmenting cells and subcellular regions directly from SMLM data are not as well established.[16–18] A range of software has been developed to segment clusters from *xy* or *xyz* localisations in SMLM data.[19] However, these approaches are generally not designed for high throughput processing of many images, require careful parameter tuning for each FOV and require a subsequent processing step to combine the segmented clusters into whole cells. Alternatively, a widefield image of the FOV, automatically thresholded, can be overlaid onto an SMLM point cloud, to segment and extract the localisations in this region.[20] However, thresholding can struggle to deal with images that are noisy or that have large variations in the intensity of the background or object, which in turn leads to poor segmentation. Nanowrap is a relatively new approach that extracts subcellular membrane surfaces, approximating the SMLM data using a coarse, density-based isosurface or density-thresholded mesh, but requires the user to set many parameters.[21]

To allow the nanoscale analysis of protein distributions in many new subcellular features and in cells, we have developed a pipeline for automatic segmentation of localisations in structures of interest. This pipeline automatically makes downstream analysis possible on specific structures in SMLM datasets over many fields of view. It is available in *locpix*, a publicly available Python package (https://github.com/oubino/locpix). *locpix* also provides a means of manually annotating SMLM data rendered as an image. This process labels individual localisations in the SMLM point cloud, giving ground-truth data for both training (in machine learning methods) and testing of segmentation. Here, we automatically classify protein localisations into sets of plasma membrane and nonmembrane localisations, testing several segmentation algorithms, and subsequently segment localisations into groups belonging to different cells. Finally, we analyse protein localisations from the segmented datasets.

2 | MATERIALS AND METHODS

2.1 | Pipeline overview

Following sample preparation and SMLM imaging and data preprocessing (including drift correction and



Channel1	Frame	X (nm)	Y (nm)	...
0	0	81.23	50.83	...
0	0	82.27	55.40	...
0	1	80.58	48.22	...
Input .parquet file				

Channel1	Frame	X (nm)	Y (nm)	...	Additional label
0	0	81.23	50.83	...	0
0	0	82.27	55.40	...	1
0	1	80.58	48.22	...	0
Output .parquet file					

FIGURE 1 *locpix* analysis pipeline for segmentation of SMLM data. (A) The point cloud data is converted from tabular style data to an image (2D histogram). Images are manually annotated in *napari* (yellow lines are manual annotations of plasma membrane). The output tabular data has an additional ground-truth label for each localisation (zero for nonmembrane and one for membrane in this case). (B) Partitions of the dataset for training and testing of a segmentation algorithm (26 FOVs in this case, using 5-fold cross-validation in training). (C) Automated segmentation obtained on data from FOVs previously unseen by the segmentation algorithm (membrane segmentation followed by cell segmentation here).

localisation filtering), we developed *locpix* (https://github.com/oubino/locpix) for manual annotation and segmentation of single-molecule localisation data (Figure 1). First, the distribution of localisations is rendered as an image (a 2D histogram). Next, it is manually annotated to identify the structures of interest (e.g. plasma membrane). This manual annotation step adds a label to all localisations, depending on whether they are within annotated pixels of the image or not. Multiple layers of annotation (multiple labels for different structures of interest) can be added at this stage, or just one. Next, a segmentation algorithm is trained or otherwise optimised (if not a learning algorithm) on a portion of the data. Then it is tested on data from fields of view (FOVs) previously unseen by the algorithm. The original annotation step provides a per-localisation ground truth label for characterising segmentation performance on the test data. The algorithm may then be applied later to segment further, unannotated data. Finally, this segmentation of localisations into different labelled subsets may be used to analyse the organisation of localisations in a particular structure or structures, for instance in the plasma membrane or distinct cells.

2.2 | Sample preparation and SMLM data acquisition

To test our pipeline, we used SMLM data obtained from imaging sections of cell pellets, taken from formalin fixed and paraffin embedded (FFPE) samples. The cell pellets were generated from an engineered single cell derived clone (C15) of a metastatic colorectal cancer cell line, SW620, which contained a targeted mutation in exon 22 of PTCH1 (Patch1).[22] This mutation upregulated expression levels of EGFR and EREG in these cells (personal communication, 2022). The sections were labelled using antibodies to the plasma membrane protein EGFR (epidermal growth factor receptor) using an anti-EGFR antibody (5B7: rabbit monoclonal, Roche) and to its ligand EREG (epiregulin) using an anti-EREG antibody (SP326: rabbit monoclonal, Roche), followed by donkey anti-rabbit Alexa 647 and goat anti-rabbit CF568 secondary antibodies. A goat anti-rabbit Fab antibody was used for blocking before adding the second primary anti-EGFR antibody. Staining for these two proteins generated a high-density membrane localisation, often visible as cell outlines in sections through the cell pellet.

To image the samples, we performed TIRFM (total internal reflection fluorescence microscopy) dSTORM (direct stochastic optical reconstruction microscopy) imaging using a commercial system (Nanoimager (ONI)) and 100 × 1.4 NA oil-immersion objective lens. Samples were bathed in STORM buffer (B-cubed buffer (ONI, BCA0017)). Using an exposure time of 30 ms, 5000 frames per channel were acquired sequentially. The 640 nm laser was set to 60% power and the 561 nm laser to 20% power of the maximum excitation output of the Nanoimager. 2D localisation of fluorescence emission events was performed while imaging using NimOS (ONI, UK).

We obtained 26 FOVs from four samples with approximately 1.5×10^7 localisations per FOV. In 12 of the 26 FOVs, EREG/EGFR were imaged in the 568/647 nm channels, respectively. In the remaining 14 FOVs, EREG/EGFR were imaged in the 647/568 nm channels respectively.

2.3 | Data preprocessing

Drift correction, filtering and temporal grouping for each FOV was performed using CODI (COllaborative Discovery platform from ONI, UK; https://oni.bio/nanoimager/software/codi-software/). The filtering step removed localisations in the 647 nm channel in frames 5000-10,000 (while imaging at 568 nm) and in the 568 nm channel in frames 0–4999 (while imaging at 647 nm). In addition, localisations with >30,000 photons, with a standard deviation of the fitted point spread function (PSF) <75 nm or >200 nm, with a p-value for the fitted PSF above 0.01 or with a localisation precision >25 nm were removed. Localisations within 60 nm and no more than two frames apart were grouped, removing those that existed for longer than five frames. This resulted in ~ 250,000 localisations per FOV.

Each FOV was then reconstructed into one 2D histogram (image) per channel. The data in its proprietary format was first converted into an Apache Parquet file, a column-orientated data format which can be more efficient for querying and storing than .csv files (https://parquet.apache.org/). For each localisation, the channel, frame number and xy coordinates were stored. For each FOV, the point cloud data for the EGFR and EREG channels were binned into separate 2D histograms and rendered as images, with pixel grey levels equal to the bin values. Each histogram consisted of 500 × 500 pixels over the x and y range of the FOV. Since the range varied between FOVs, each pixel was between 99–100 nm wide and 157–160 nm tall. For the analysis presented here, we merged the data from the two channels, to obtain one overall ‘membrane’ dataset.

2.4 | Software development

We created *locpix*, a Python 3 package installable via the Python Package Index. *locpix* prepares tabular SMLM data for analysis (https://github.com/oubino/locpix) and includes conversion of the SMLM data to 2D histograms using *numpy* (https://pypi.org/).[23] It enables manual annotation of the dataset via *napari*, an image viewer implemented in Python 3. For membrane segmentation, *locpix* includes Otsu thresholding, a standard U-Net model implemented in PyTorch using code adapted from https://github.com/milesial/Pytorch-UNet and Cellpose (version 2.0).[14,24] Cell segmentation using the watershed algorithm was implemented via scikit-image.[25] We also include input and output of data to and from the Ilastik GUI for membrane and cell segmentation.[15] Finally, we include evaluation of performance metrics for the segmentation. For more details on the commands used to produce the analysis below, please see the README in the *locpix* repository https://github.com/oubino/locpix.

2.5 | Manual annotation

The EGFR and EREG preprocessed localisation distributions (point clouds) were binned into 2D histograms, rendered as images, and loaded into separate channels in *napari*.[26] In *napari*, the cell membranes were manually traced using the freehand drawing tool to generate a ground-truth labelled image, in which each pixel had an integer value of either zero for nonmembrane or one for membrane. The ground-truth label for each pixel was then assigned to all localisations within the corresponding 2D histogram bin. The localisations were then exported into a new Apache Parquet file, with an additional column for the ground-truth label (Figure 1A). This manual annotation step is available as an open-source (OS) napari plugin at https://www.napari-hub.org/plugins/napari-locpix.

2.6 | Dataset partitions

Separate datasets for training and evaluation were created (Figure 1B). First, the entire dataset was divided into a training set (70%) and a test set (30%). The test set was generated from the FOVs with the highest percentage of membrane localisations according to the manual annotations and was not used until performance analysis. The training set was then divided into five subsets (hereafter referred to as folds). Five different splits of the training dataset were then generated, each with a different fold for validation and the remaining folds for training (Figure 1B). For each of the five methods used for membrane and cell segmentation detailed below, we developed a model for each split of the training dataset, using the training folds for training the model where relevant (standard U-Net, Cellpose (retrained) and Ilastik) and the validation set for comparing performance. Each model was then evaluated on the test set for the final comparison.

2.7 | Segmentation algorithms: probability map generation

We developed several methods to predict the probability for each pixel in a FOV that it was located within the plasma membrane (probability map). These probabilities were then assigned to the localisations belonging to each pixel. First, for each method except Ilastik, the EGFR and EREG images (2D localisation histograms) were summed into a single channel for processing. For Ilastik, the input was a two-channel EGFR and EREG image. The pixel values in the resulting images (and each channel in the Ilastik method) were scaled by $\log_2$ to reduce skew, thresholded above zero and scaled to between 0 and 255. We obtained membrane probability maps from these images using the following approaches.

- Otsu thresholding: A binary probability map (0 or 1 for each pixel and underlying localisations) is obtained after Otsu thresholding of the transformed and scaled image.
- U-Net: Standard U-Net architecture, with four encoder blocks and decoder blocks with skip connections between them, and a final sigmoid function to convert the raw output values to normalised probabilities for image pixels (Supplementary Information: Architectures). Images were normalised by subtracting the mean and dividing by the standard deviation of all pixel values in the images in the training folds. Further, random augmentation including rotations, horizontal and vertical flips, erasing, and perspective shifting was applied to the training folds. The model was trained for 1000 epochs using a dice loss function and Adam optimiser with a learning rate of 0.01 and weight decay of 0.0001.[27] The model was saved when the loss on the validation fold was lowest.
- Cellpose (pretrained): We used the 'LC1' model in Cellpose, pretrained on phase-contrast images of cells with only a single channel for cytoplasm. The LC1 model was considered the most appropriate, given we expect the edge of the cytoplasm and EGFR/EREG proteins to define a similar boundary for the cell. Brief experimentation on a training image also indicated this was the best-performing Cellpose model. LC1 is a modification of the standard U-Net (Supplementary Information: Architectures). Cell diameter in LC1 was manually set to 100 pixels as determined from training FOVs. Cellpose (pretrained) assigned to each pixel an unnormalised probability that it belonged to a cell, which we reassigned as the probability it belonged to a membrane, followed by scaling to between 0 and 1, as usually performed by Cellpose.
- Cellpose (retrained): We modified the Cellpose training script to change the loss function to calculate binary cross-entropy logits loss between the ground-truth label image from manual annotation and the output membrane probability map, manually set the cell diameter and mean cell diameter for all images to 100 pixels and allow training to run over multiple splits of the data. The pretrained Cellpose LC1 model was retrained for 1000

epochs, with a weight decay of 0.0001 and a learning rate of 0.01. We performed limited tuning for these hyperparameters by training with a small partition of split zero's training folds (Figure 1B), without evaluation on the test dataset. The output probabilities were scaled as in Cellpose (pretrained). The model was saved when the loss on the validation fold was lowest.

- Ilastik: The two-channel images and ground-truth membrane annotation images were used to perform the Ilastik pixel classification workflow. We trained the model using the Ilastik GUI with all possible image features. No further annotations were made to the ground-truth images once they were loaded into the GUI. We chose a label of 2, rather than 0 for nonmembrane pixels (in Ilastik a label of zero means no label is present) and randomly removed ∼80% of these nonmembrane ground-truth pixel labels to reduce the computational overhead.

2.8 | Membrane segmentation

The point cloud datasets were then segmented by classifying each localisation as membrane (positive class) or nonmembrane (negative class). Performance was evaluated at this per-localisation level, as this is most relevant to downstream analysis of nanoscale protein organisation. Supplementary Information: Performance metrics (together with Tables S1 and S2, and Figure S1) provides more information on the metrics used to evaluate performance.

Following the membrane probability map generation with each method, the probabilities assigned to the localisations underlying each pixel were used together with the ground-truth annotation of those pixels (membrane or not membrane) to plot precision–recall (PR) curves. We prefer this approach to the commonly-used ROC (receiver operating characteristic) curve based on its sensitivity to changes in false positives (FP), despite a large number of true negatives (TN).[28,29] First, for each split of the training data, the localisations from all FOVs in the training folds were aggregated into one table. In this table, every localisation now had both its ground-truth label from manual annotation (0: nonmembrane or 1: membrane) and its model-generated probability of belonging to a membrane. PR curves were generated by calculating the precision and recall for different values of a classification threshold, τ, that increased from zero to one, where localisations with a probability above each τ were assigned to the membrane (Supplementary Information: Performance Metrics). PR curves for the validation folds and the test set were generated using the same method. The normalised area under the PR curve (AUCNPR: area under curve normalised PR) was then calculated for the test set.[30] For the Cellpose (pretrained) and Otsu models, which were not trained on our data, the probability maps on the test set did not depend on our training dataset splits, resulting in only one PR curve and AUCNPR in each case.

The F_1 score was chosen as the primary performance metric as it accounts for class imbalance, as is present in our dataset (more nonmembrane than membrane localisations).[31,29] For each segmentation method, the probability threshold τ that maximised the F_1 score for localisation classification was determined for each training fold. These values of τ were then applied to the probability maps for the test dataset, used to calculate the accuracy, precision, recall and the F_1 score for each model on previously unseen data. For the Otsu method, the membrane probability map was already binary, and setting different probability classification thresholds, τ, between 0 and 1 did not affect results. Therefore, there was a single result on the test set for each metric, independent of the training dataset splits.

2.9 | Cell segmentation

The probability maps for each method were used as the starting point for cell segmentation, except for the Otsu method, which operated directly on the 2D localisation histogram. For all 26 histograms in the original SMLM dataset, the approximate location of the cell centre was identified manually, giving seed locations for the watershed algorithm for the Otsu, standard U-Net and Cellpose methods. Seeds were also placed across the background, to prevent cell segmentations extending into the background and to avoid assigning the same label to well-separated membrane localisations on different cells. Ilastik on the other hand, automatically calculated seed locations for the watershed algorithm.

For the Otsu method, the EGFR and EREG 2D localisation histograms were summed, transformed by $\log_2$, thresholded above zero and scaled to between zero and 255 as performed for membrane segmentation. The result was used as input to the watershed algorithm. The decision to use this thresholded and scaled histogram rather than the raw histogram was based on a visual analysis of the performance on one training histogram. For standard U-Net and both Cellpose methods, the watershed algorithm was applied to the probability maps from membrane segmentation. For Ilastik, the input 2D localisation histograms and output probability maps from the Ilastik membrane segmentation were used to perform the Ilastik multicut workflow. We trained the model using the Ilastik GUI with only five histograms from the training folds to save time. Ilastik then batch-processed the remaining histograms from the entire dataset using these trained parameters.

2.10 | Downstream analysis

We performed an example of downstream analysis of segmented membranes as follows (https://github.com/oubino/locpix/blob/main/examples/c15_data_ds_analysis/analysis.ipynb). First, we manually picked well-segmented cells from the test set. Next, comparing localisations at the plasma membrane and interior of each cell, we calculated the 2D radial distribution function and clustered the localisations using DBSCAN with epsilon and minimum points set to 75 nm and 5 respectively. Then, aggregating over the cells for the test dataset, we calculated the overall 2D radial distribution function, localisations per cluster, and cluster length (using the convex hull) for the cell interiors and plasma membranes.

2.11 | Statistical analyses

The distributions of the cluster parameters for interior and membrane localisations were tested for normality using the Shapiro–Wilk test. As the distributions failed the test for normality, we used the two-tailed nonparametric Mann–Whitney U rank test to test the null hypothesis that the distributions for interior and membrane cluster parameters come from the same population, with the null hypothesis rejected for $p \leq 0.05$. Both tests were implemented in the Python package SciPy.[32]

3 | RESULTS

3.1 | Manual annotation

All 26 FOVs from the original SMLM dataset were manually annotated. Each localisation that belonged to a membrane was first manually annotated using the custom image annotation script (Figures 1A and 2A–D). It was not always possible to clearly differentiate between the membrane, and the cell interior or general background (defined here as nonmembrane). The manual membrane annotations thus did not always delineate the entirety of a cell (Figure 2A–D). This contributed to a small imbalance in the dataset, with ~1.5 times more nonmembrane than membrane localisations for the dataset.

3.2 | Membrane segmentation

Multiple methods for membrane segmentation were developed, trained and validated on five splits of the training dataset. These included methods that learnt from our training dataset (Standard U-Net, Cellpose (retrained), Ilastik): one that had been pretrained on a different dataset (Cellpose (pretrained)) and one without any machine learning (Otsu method). The results of these methods were compared considering both the quantitative and qualitative performance on the test set (Table 1 and Figures 3 and 2). We used AUCNPR (area under the curve, normalised precision–recall) as the key measure of performance as it balances precision and recall and accounts for class imbalance, while the F_1 score was most useful when evaluating the performance on the final segmentation.[29,31,33] We were mostly confident that the annotated membranes were true membrane regions but cannot rule out that some may have been missed; therefore, true-positive and false-negative counts were the most reliable. This made recall the most reliable metric (Table 1), despite the pitfall of predicting all localisations as membrane giving the maximum recall (1.0).

We found that Otsu was the second worst-performing model for membrane segmentation (AUCNPR: 0.63, Table 1). It overestimated the membrane localisations, as shown by the high membrane recall (0.902, Table 1) and low membrane precision (0.543, Table 1). Furthermore, the false positives were in areas unlikely to be regions of membrane that were mislabelled during annotation, such as cell interiors (Figure 2E and G, μ).

Standard U-Net significantly outperformed Otsu and was the second best of all approaches (AUCNPR: 0.810 ± 0.003, Table 1). It had higher nonmembrane recall and membrane precision but lower membrane recall than Otsu (Table 1), demonstrating that it predicted more membrane localisations as nonmembrane but made fewer false-positive membrane predictions. In particular, it predicted fewer localisations in cell interiors than membranes compared to Otsu (Figure 2E and G, μ).

Cellpose (pretrained) performed the worst of all (AUCNPR: 0.36, Table 1). Despite outperforming Otsu in all metrics apart from AUCNPR (Table 1), the predictions appeared visually similar (Figure 2E–H). Further, like Otsu and unlike standard U-Net, it mistook cell interiors as membranes (Figure 2G, μ). It did have higher membrane recall than standard U-Net (Table 1), but it also made more false-positive mistakes (lower membrane precision, Table 1), predicting more of the nonmembrane localisations as belonging to the membrane.

Cellpose (retrained) was the best-performing model (AUCNPR: 0.853 ± 0.008, Table 1). Retraining Cellpose demonstrated a clear performance improvement (rise in the PR curve vs. pretrained, Figure 3). This model predicted fewer localisations as belonging to a membrane (decrease in the membrane recall, Table 1), noticeably predicting fewer cell interiors as membranes compared to Otsu, Ilastik and Cellpose (pretrained) (Figure 2G, μ). Compared to standard U-Net, Cellpose (retrained) gave

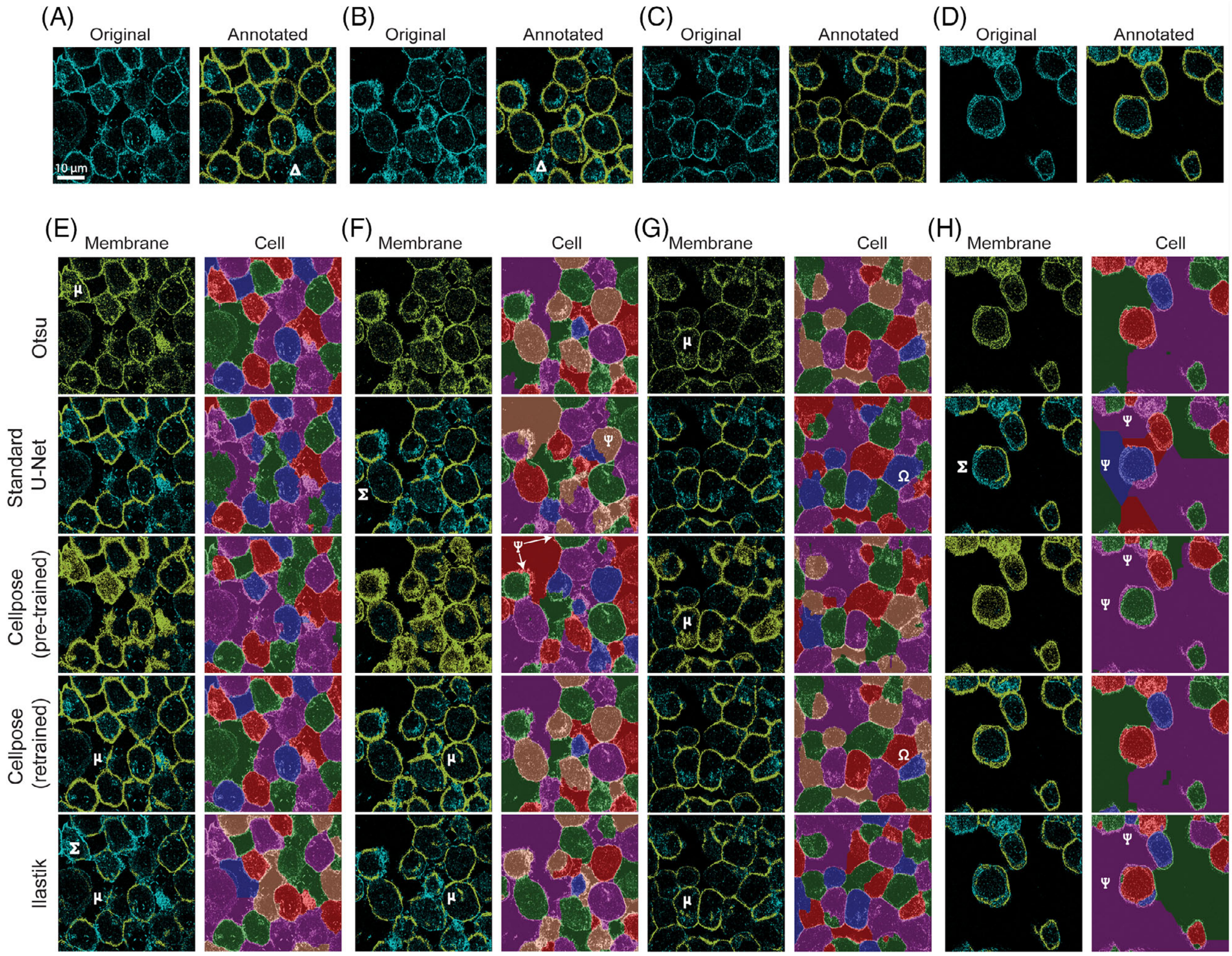


FIGURE 2 Manual annotation, membrane segmentation and cell segmentation results. (A–D) The original 2D histogram (sum of EGFR and EREG) together with the annotated 2D dataset (yellow: membrane annotation) for four FOVs from the test set (unseen to all trained methods), where all results are from the same split (zero). (E–H) The membrane and cell segmentations for each of the five methods. For cell segmentation, each colour represents a different cell label and nonbordering segments of the same colour represent different cell labels. KEY: Δ: nonannotated region that could be membrane; μ: nonmembrane localisations predicted as membrane; Σ: membrane localisations predicted as nonmembrane; Ψ: error in cell segmentation; Ω: instability of the watershed algorithm.

TABLE 1 Performance metric scores (Supplementary Information: Performance metrics) for each method.

	Recall nonmembrane	Recall membrane	Precision nonmembrane	Precision membrane	F_1 score membrane	Accuracy membrane	AUCNPR membrane
Otsu	0.205	0.902	0.666	0.543	0.678	0.561	0.63
Standard U-Net	**0.948 ± 0.001**	0.608 ± 0.006	0.698 ± 0.003	**0.924 ± 0.001**	0.734 ± 0.004	0.774 ± 0.003	0.810 ± 0.003
Cellpose (pretrained)	0.273 ± 0.036	**0.943 ± 0.013**	0.824 ± 0.019	0.576 ± 0.008	0.715 ± 0.003	0.616 ± 0.011	0.36
Cellpose (retrained)	0.783 ± 0.023	0.842 ± 0.009	**0.826 ± 0.005**	0.803 ± 0.015	**0.822 ± 0.004**	**0.813 ± 0.007**	**0.853 ± 0.008**
Ilastik	0.911 ± 0.012	0.539 ± 0.033	0.654 ± 0.014	0.864 ± 0.010	0.663 ± 0.023	0.721 ± 0.012	0.784 ± 0.011

Note: Scores are presented as the mean ± standard deviation over the five splits evaluated on the test set. Recall is given by $\frac{TP}{TP+FN}$, precision (prec.) by $\frac{TP}{TP+FP}$, F_1 score by $\frac{2\times precision\times recall}{precision+recall}$ and accuracy (acc.) by $\frac{TP+TN}{TP+TN+FP+FN}$,, where TP, TN, FP, and FN are the number of true-positive, true-negative, false-positive and false-negative predictions, respectively, and either nonmembrane or membrane is the positive class. AUCNPR is the normalised area under a curve that plots precision against recall for different thresholds applied to the probability map. For Otsu and Cellpose (pretrained), there is no variance for AUCNPR as the split has no impact on the probability map for the test set (Section 2). For Otsu, the remaining metrics have no variance as changing the threshold for each split has no impact on the probability maps (Section 2). The best scores for each metric are highlighted in bold.

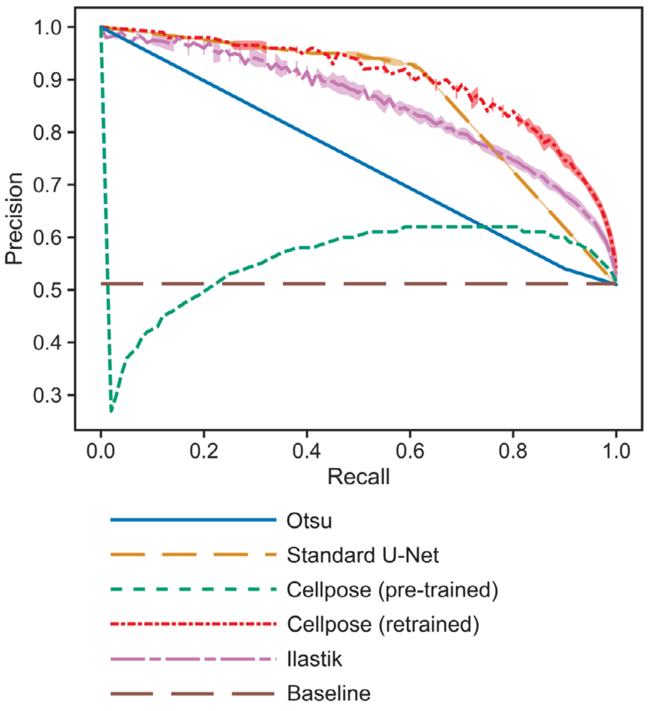


FIGURE 3 Precision–recall curves for each method evaluated on the test set. The curves show the mean over the five training dataset split models with ±1 standard deviation shaded where appropriate. No variance values for Otsu and Cellpose (pretrained) are shown as the algorithm applied to the test data did not depend upon the training data (Section 2). Baseline performance was a model that predicted all localisations as belonging to a membrane.

more extensive segmentations (higher membrane recall, Table 1) but at the cost of more false positives (lower membrane precision, Table 1). At points standard U-Net seemed to better reflect our annotations, omitting an edge that Cellpose (retrained) predicted (Figure 2E, μ), which, despite looking membranous, was not manually annotated. Further, standard U-Net correctly segmented localisations in cell interiors that were mislabelled as membrane by Cellpose (retrained) (Figure 2F, μ). However, standard U-Net also omitted regions that were clearly membrane, which Cellpose (retrained) correctly segmented (Figure 2F and H, Σ). Cellpose (retrained) also identified regions that may have been membrane and that we were not confident enough to annotate (Figure 2A and B, Δ).

Finally, we found that Ilastik performed worse than retrained Cellpose and standard U-Net (AUCNPR: 0.784 ± 0.011, Table 1). Like standard U-Net, it predicted more membrane as nonmembrane than Cellpose (retrained) (higher nonmembrane recall and lower membrane recall, Table 1). This included missing a significant proportion of the membranes (Figure 2E, Σ), which Cellpose (retrained) correctly segmented. Further, it made similar mistakes to Cellpose (retrained) with interiors (Figure 2F and G, μ) and trailing edges (Figure 2E, μ), the latter of which were not manually annotated despite looking membranous. Ilastik overfit the training data, as evidenced by the poorer performance for the validation and test set compared to the training set (Figures 3 and S2). One likely reason for this is the small size of the dataset. A second is that we did not monitor for overfitting during training by evaluating the performance on the validation set. Therefore, the probability threshold determined using the F_1 score on the training set was also likely to be suboptimal when applied to the test set.

3.3 | Cell segmentation

Multiple methods for cell segmentation were developed, trained and validated on five splits of the training dataset and results compared using a qualitative analysis on the test set (Figure 2). Quantitative performance metrics on the localisations could not be evaluated as the cells were not manually annotated.

All methods segmented some cells correctly but generally performed poorly. This was most evident in the examples shown in Figure 2H, where we were most confident in the ground truth, reflected in the extensive manual annotations (Figure 2D: ∼7 identified cells). Cellpose (retrained) did not make the same mistakes that Cellpose (pretrained), standard U-Net and Ilastik made for cell segmentation (Figure 2H, Ψ). This was expected as these methods relied on the quality of the membrane segmentation, which was best for Cellpose (retrained). Further, as Cellpose (retrained) provided more extensive annotations than the other high-performing model, standard U-Net, it was less likely to divide cells in two because there were gaps in the membrane annotation (Figure 2F, standard U-Net, Ψ). Despite this, the performance of Cellpose (retrained) was almost identical to the much simpler Otsu method (Figure 2E–H).

There were problems using the watershed algorithm for cell segmentation across all methods. Firstly, localisations from the exterior of different cell membranes were incorrectly assigned the same label, despite being far apart (Figure 2F, Cellpose (pretrained), Ψ). Even though the same markers were used for all methods (apart from Ilastik), small differences in the membrane segmentation caused large differences in the cell segmentation (Figure 2G, Ω).

3.4 | Downstream analysis

Once the data is segmented, it can then be used in downstream analysis, as we show here for membrane and cell segmentation results, by exploring EGFR

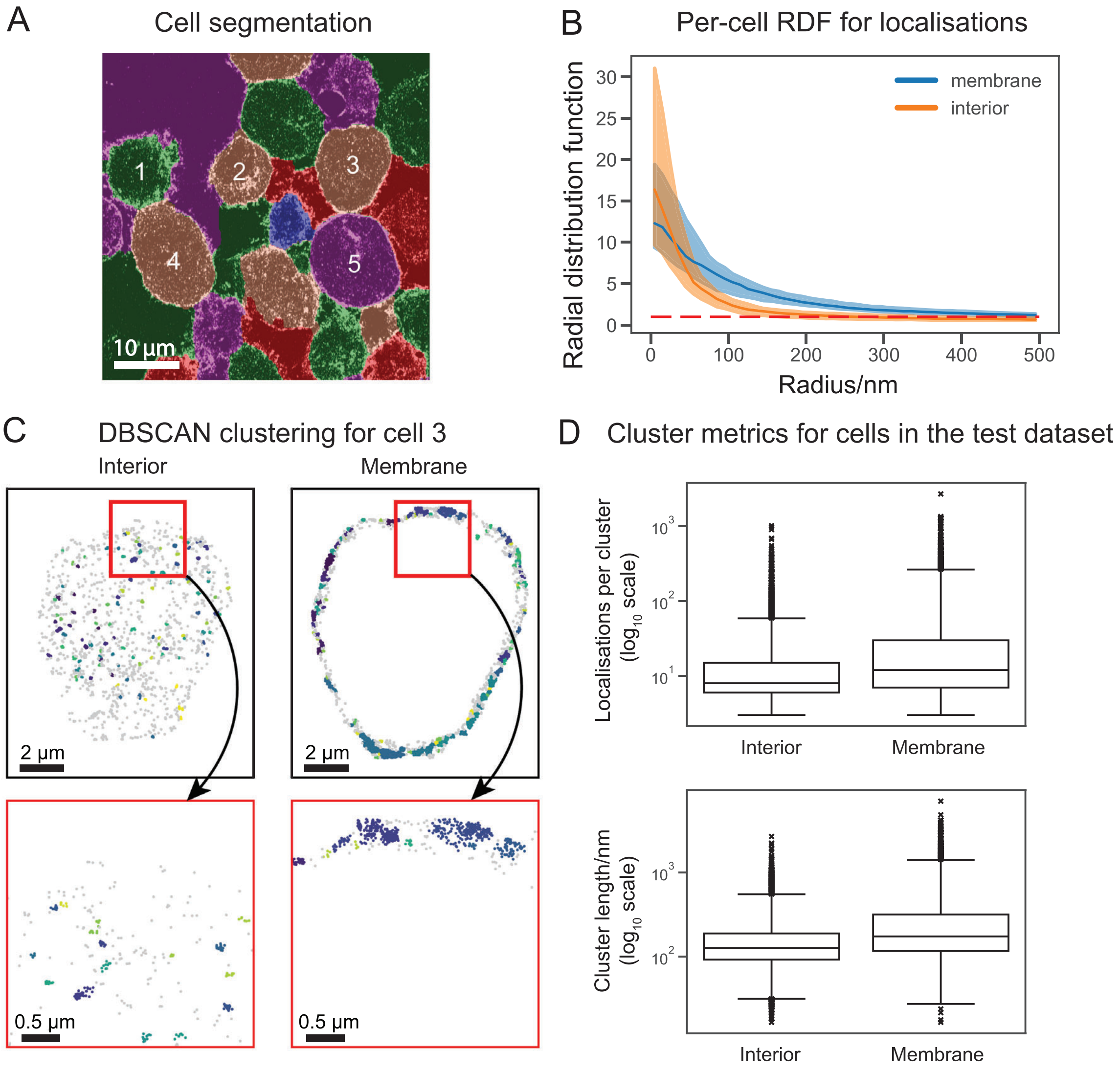


FIGURE 4 Analysis of segmented data. (A) Part of an FOV from the test dataset that shows five of the eight cells that were manually selected following cell segmentation. (B) Radial distribution function (RDF) for the membrane and cluster localisations calculated for each cell and aggregated over the whole test set. The median value for each radius is plotted with interquartile range shaded. (C) Interior localisations and plasma membrane localisations for cell 3 from panel A. Each colour represents a different cluster from DBSCAN (epsilon = 75 nm). Grey localisations do not belong to a cluster. Localisations within the red box are shown at higher magnification below as indicated. (D) Cluster metrics for the cells in the test dataset. Box plots for the number of localisations per cluster (top) and cluster length (bottom) for each cell aggregated over the whole test dataset. The three central horizontal lines are the first (Q1), second (Q2) and third quartile (Q3) from bottom to top; the whiskers are Q1–1.5 × (Q3–Q1) and Q3–1.5 × (Q3–Q1); and the outliers are plotted as crosses. Median localisations per cluster: 8 (interior), 12 (membrane), Mann–Whitney $U = 1.9 \times 10^7$, $n_{\text{membrane}} = 6183$, $n_{interior} = 8141$, $p \leq 0.05$. Median cluster lengths: 127 nm (interior), 174 nm (membrane), Mann–Whitney $U = 1.8 \times 10^7$, $n_{\text{membrane}} = 6183$, $n_{\text{interior}} = 8141$, $p \leq 0.05$.

distribution and clustering in the cell membrane and interior (Figure 4). From the automated segmentation from Cellpose (retrained), we manually selected well-segmented cells from all FOVs in the test dataset (60 cells) and separately calculated the 2D radial distribution function and clustering of their localisations (DBSCAN: epsilon = 75 nm, minimum points = 5) predicted as 'membrane' or 'nonmembrane'. Localisations within the interior of the cells were characteristically found in close proximity (≤approx. 100 nm), while typical distances between those at the membrane extended over a longer distribution (Figure 4B). Localisations at the cell membrane formed clusters with significantly higher localisations per cluster and cluster length than the cell interior (Figure 4D). These clusters included repeated localisations of the same fluorescent dye molecule, multiple dye molecules per secondary labelling antibody and any clustered instances of EGFR. Although many membrane localisations are grouped into large clusters, there is a smaller number of these large clusters, and the difference between the distributions appears to reveal a more subtle difference between the arrangement of EGFR in the interior and at the membrane. The cluster parameter distributions in the interior likely include a major contribution from monomeric EGFR, with multiple dye molecules per labelling antibody. The increase in the median length and number of localisations per cluster at the membrane may be a result of the known dimerisation of EGFR at the membrane, or a larger number close together, although results are confounded by the dense packing at the membrane and the multiple fluorescent molecules per labelling antibody.

4 | DISCUSSION

We have demonstrated a pipeline for annotating and automatically segmenting cells and membranes from SMLM point-cloud data, which is important for downstream subcellular analysis. Using the normalised area under the precision–recall curve (AUCNPR) to compare methods, we found that Cellpose (retrained) performed the best for membrane segmentation. This type of approach is useful for identifying molecular localisations as belonging to specific cells and regions of cells, to enable directed analysis of localisation data specific to those regions.

The trained models outperformed traditional methods for membrane segmentation (Otsu method).[34] This was expected, as the heterogeneity in membrane staining between cells makes it challenging to set a threshold for the entire FOV. Further, standard thresholding techniques fail to consider both local (cells) and global (entire FOV) context and can struggle to deal with small objects and images that are noisy or show significant variation in the background or object intensity.[35,36]

The inability of these trained models to outperform the Otsu method in cell segmentation points to issues with applying the watershed algorithm to this task. Small changes in the membrane segmentation led to large differences in cell segmentation, and localisations from different cell membranes that were large distances apart could be assigned the same label. This is expected based on the known disadvantages of watershed that it is sensitive to noise and inhomogeneity of background and object intensity, and it does not consider the global context.[37,38]

When comparing membrane segmentation models, one should consider if it is more important for downstream analysis to minimise the number of missing membranes (false negatives), or the number of nonmembrane localisations predicted as belonging to a membrane (false positives). Cellpose (retrained) had higher membrane recall but lower membrane precision than standard U-Net. If it is more important to avoid false positives while allowing more false negatives, then standard U-Net would be better, and vice versa for Cellpose (retrained). The accuracy of these metrics should also be considered; for example, some of the false positives from Cellpose (retrained) were in regions likely to contain membrane that we were not confident enough to annotate.

Using an ensemble of metrics can give a fairer indication of model performance across both positive and negative classes, rather than focusing on a single metric. For example, membrane recall could be misleading in isolation, because it can be maximised by classifying all localisations as membrane and giving no true negatives. Combining PR curves and AUCNPR with *k*-fold cross-validation measures how robust the methods are to changes in threshold and changes to training and evaluation data respectively. The performance of Cellpose (retrained) was less variable across a range of thresholds, evidenced by the highest AUCNPR (Table 1), which is important if setting the threshold is challenging.

5 | CONCLUSION

The workflow we have developed demonstrates how membrane and cell segmentation can be incorporated into the analysis pipeline for SMLM data in a range of applications. SMLM data segmentation allows the analysis of high-precision molecular distributions in specific subcellular regions, for instance using one of the many preexisting algorithms.[17,39,40] This analysis can be extended to annotate, segment and analyse SMLM images from a broad variety of samples, including clinical samples. Eventually, this may enable the automated evaluation of biomarkers with SMLM, with the potential of predicting patient response to treatment.

SOFTWARE AVAILABILITY

locpix can be found at https://github.com/oubino/locpix and is installable via the Python Package Index (https://pypi.org/project/locpix/). The manual annotation tool is available as a napari plugin from https://www.napari-hub.org/plugins/napari-locpix. The modified Cellpose training script is also available at https://github.com/oubino/cellpose. The downstream analysis is available as a Jupyter notebook in the *locpix* GitHub repository at https://github.com/oubino/locpix/blob/main/examples/c15_data_ds_analysis/analysis.ipynb.

ACKNOWLEDGEMENTS

OU was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/S024336/1 for the University of Leeds Centre for Doctoral Training. PQ and AC acknowledge funding from a National Institute for Health and Care Research (NIHR) i4i grant (Award ID:

NIHR201643) in collaboration with ONI. PQ is an emeritus NIHR Senior Investigator, partly supported by NIHR Leeds Biomedical Research Centre. JL was funded by an EPSRC fellowship: EP/R025819/1. MP is a Wellcome Trust Investigator (223125/Z/21/Z). The views expressed are those of the authors and not necessarily those of the NHS, the NIHR or the Department of Health and Social Care. Engineering of the SW620 cells was funded by the Biotechnology and Biological Sciences Research Council, grant number BB/S01716X/1 (NRD-G), and Fondazione Italiana per la Ricerca sul Cancro (AIRC) IG 25833 (GC). We also thank Roche for the generous donation of the EGFR and EREG antibodies.

ORCID

Gianluca Canettieri https://orcid.org/0000-0001-6694-2613
Michelle Peckham https://orcid.org/0000-0002-3754-2028
Alistair Curd https://orcid.org/0000-0002-3949-7523

REFERENCES

1. Betzig, E., Patterson, G. H., Sougrat, R., Lindwasser, O. W., Olenych, S., Bonifacino, J. S., Davidson, M. W., Lippincott-Schwartz, J., & Hess, H. F. (2006). Imaging intracellular fluorescent proteins at nanometer resolution. *Science*, *313*, 1642–1645.
2. Rust, M. J., Bates, M., & Zhuang, X. (2006). Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nature Methods*, *3*, 793–795.
3. Heilemann, M., van de Linde, S., Schuttpelz, M., Kasper, R., Seefeldt, B., Mukherjee, A., Tinnefeld, P., & Sauer, M. (2008). Subdiffraction-resolution fluorescence imaging with conventional fluorescent probes. *Angewandte Chemie (International ed in English)*, *47*, 6172–6176.
4. Heilemann, M., van de Linde, S., Mukherjee, A., & Sauer, M. (2009). Super-resolution imaging with small organic fluorophores. *Angewandte Chemie (International ed in English)*, *48*, 6903–6908.
5. Jacquemet, G., Carisey, A. F., Hamidi, H., Henriques, R., & Leterrier, C. (2020). The cell biologist's guide to super-resolution microscopy. *Journal of Cell Science*, *133*, jcs240713.
6. Liu, S., Hoess, P., & Ries, J. (2022). Super-resolution microscopy for structural cell biology. *Annual Review of Biophysics*, *51*, 301–326.
7. Hugelier, S., Colosi, P. L., & Lakadamyali, M. (2023). Quantitative single-molecule localization microscopy. *Annual Review of Biophysics*, *52*, 139–160.
8. Nehme, E., Freedman, D., Gordon, R., Ferdman, B., Weiss, L. E., Alalouf, O., Naor, T., Orange, R., Michaeli, T., & Shechtman, Y. (2020). DeepSTORM3D: Dense 3D localization microscopy and PSF design by deep learning. *Nature Methods*, *17*, 734–740.
9. Ouyang, W., Aristov, A., Lelek, M., Hao, X., & Zimmer, C. (2018). Deep learning massively accelerates super-resolution localization microscopy. *Nature Biotechnology*, *36*, 460–468.
10. Speiser, A., Muller, L. R., Hoess, P., Matti, U., Obara, C. J., Legant, W. R., Kreshuk, A., Macke, J. H., Ries, J., & Turaga, S. C. (2021). Deep learning enables fast and dense single-molecule localization with high accuracy. *Nature Methods*, *18*, 1082–1090.
11. Gogoberidze, N., & Cimini, B. A. (2024). Defining the boundaries: Challenges and advances in identifying cells in microscopy images. *Current Opinion in Biotechnology*, *85*, 103055.
12. Falk, T., Mai, D., Bensch, R., Cicek, O., Abdulkadir, A., Marrakchi, Y., Bohm, A., Deubner, J., Jackel, Z., Seiwald, K., Dovzhenko, A., Tietz, O., Dal Bosco, C., Walsh, S., Saltukoglu, D., Tay, T. L., Prinz, M., Palme, K., Simons, M., … Ronneberger, O. (2019). U-Net: Deep learning for cell counting, detection, and morphometry. *Nature Methods*, *16*, 67–70.
13. Siddique, N., Paheding, S., Elkin, C., & Devabhaktu, V. (2021). U-Net and its variants for medical image segmentation: Theory and applications. *arXiv*.
14. Pachitariu, M., & Stringer, C. (2022). Cellpose 2.0: How to train your own model. *Nature Methods*, *19*, 1634–1641.
15. Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., Eren, K., Cervantes, J. I., Xu, B., Beuttenmueller, F., Wolny, A., Zhang, C., Koethe, U., Hamprecht, F. A., & Kreshuk, A. (2019). ilastik: Interactive machine learning for (bio)image analysis. *Nature Methods*, *16*, 1226–1232.
16. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 226–231). AAAI Press.
17. Nieves, D. J., & Owen, D. M. (2020). Analysis methods for interrogating spatial organisation of single molecule localisation microscopy data. *International Journal of Biochemistry & Cell Biology*, *123*, 105749.
18. Pike, J. A., Khan, A. O., Pallini, C., Thomas, S. G., Mund, M., Ries, J., Poulter, N. S., & Styles, I. B. (2020). Topological data analysis quantifies biological nano-structure from single molecule localization microscopy. *Bioinformatics*, *36*, 1614–1621.
19. Nieves, D. J., Pike, J. A., Levet, F., Williamson, D. J., Baragilly, M., Oloketuyi, S., de Marco, A., Griffie, J., Sage, D., Cohen, E. A. K., Sibarita, J. B., Heilemann, M., & Owen, D. M. (2023). A framework for evaluating the performance of SMLM cluster analysis algorithms. *Nature Methods*, *20*, 259–267.
20. Sieben, C., Banterle, N., Douglass, K. M., Gonczy, P., & Manley, S. (2018). Multicolor single-particle reconstruction of protein complexes. *Nature Methods*, *15*, 777–780.
21. Marin, Z., Fuentes, L. A., Bewersdorf, J., & Baddeley, D. (2023). Extracting nanoscale membrane morphology from single-molecule localizations. *Biophysical Journal*, *122*, 3022–3030.
22. Caballero-Ruiz, B., Gkotsi, D. S., Ollerton, H., Morales-Alcala, C. C., Bordone, R., Jenkins, G. M. L., Di Magno, L., Canettieri, G., & Riobo-Del Galdo, N. A. (2023). Partial truncation of the C-terminal domain of PTCH1 in cancer enhances autophagy and metabolic adaptability. *Cancers (Basel)*, *15*, 369.
23. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.

H., Brett, M., Haldane, A., Del Rio, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*, 357–362.

24. Stringer, C., Wang, T., Michaelos, M., & Pachitariu, M. (2021). Cellpose: A generalist algorithm for cellular segmentation. *Nature Methods*, *18*, 100–106.
25. van der Walt, S. J., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). Scikit-image: Image processing in Python. *arXiv:1407.6245*.
26. Ahlers, J., Althviz Moré, D., Amsalem, O., Anderson, A., Bokota, G., Boone, P., Bragantini, J., Buckley, G., Burt, A., Bussonnier, M., Can Solak, A., Caporal, C., Doncila Pop, D., Evans, K., Freeman, J., Gaifas, L., Gohlke, C., Gunalan, K., Har-Gil, H., … Yamauchi, K. (2022). *Napari: A multi-dimensional image viewer for Python (v0.4.18)*. Zenodo.
27. Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980v9*.
28. He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*, 263–1284.
29. Tharwat, A. (2021). Classification assessment methods. *Applied Computing and Informatics*, *17*, 168–192.
30. Boyd, K., Santos Costa, V., Davis, J., & Page, C. D. (2012). Unachievable region in precision-recall space and its effect on empirical evaluation. In *ICML'12: Proceedings of the 29th International Conference on International Conference on Machine Learning* (Vol., *2012*, pp. 349). Omnipress.
31. Maxwell, A. E., Warner, T. A., & Guillén, L. A. (2021). Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—Part 1: Literature review. *Remote Sensing*, *13*.
32. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy, C. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*, 261–272.
33. Bender, S. W. B., Dreisler, M. W., Zhang, M., Kaestel-Hansen, J., & Hatzakis, N. S. (2024). SEMORE: SEgmentation and MORphological fingErprinting by machine learning automates super-resolution data analysis. *Nature Communications*, *15*, 1763.
34. Otsu, N. A. (1979). Threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*, 62–66.
35. Lee, S. U., Yoon, C. S., & Park, R. H. (1990). A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics, and Image Processing*, *52*, 171–190.
36. Poletti, E., Zappelli, F., Ruggeri, A., & Grisan, E. (2012). A review of thresholding strategies applied to human chromosome segmentation. *Computer Methods and Programs in Biomedicine*, *108*, 679–688.
37. Zhang, M., Zhang, L., & Cheng, H. (2010). A neutrosophic approach to image segmentation based on watershed method. *Signal Processing*, *90*, 1510–1517.
38. Beucher, S. (1979). Use of watersheds in contour detection. In *Proceedings of the International Workshop on Image Processing*. CCETT.
39. Curd, A. P., Leng, J., Hughes, R. E., Cleasby, A. J., Rogers, B., Trinh, C. H., Baird, M. A., Takagi, Y., Tiede, C., Sieben, C., Manley, S., Schlichthaerle, T., Jungmann, R., Ries, J., Shroff, H., & Peckham, M. (2021). Nanoscale pattern extraction from relative positions of sparse 3D localizations. *Nano Letters*, *21*, 1213–1220.
40. Khater, I. M., Nabi, I. R., & Hamarneh, G. (2020). A review of super-resolution single-molecule localization microscopy cluster analysis and quantification methods. *Patterns*, *1*, 100038.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Umney, O., Leng, J., Canettieri, G., Galdo, N. A. R.-D., Slaney, H., Quirke, P., Peckham, M., & Curd, A. (2024). Annotation and automated segmentation of single-molecule localisation microscopy data. *Journal of Microscopy*, 1–13. https://doi.org/10.1111/jmi.13349