CLEM-Reg: An automated point cloud based registration algorithm for correlative light and volume electron microscopy

Daniel Krentzel[1,2], Matouš Elphick[2,3], Marie-Charlotte Domart[2], Christopher J. Peddie[2], Romain F. Laine[4], Ricardo Henriques[4,5], Lucy M. Collinson[2] and Martin L. Jones[2]

[1]Imaging and Modeling Unit, Institut Pasteur, Université Paris Cité, Paris, France
[2]Electron Microscopy Science Technology Platform, The Francis Crick Institute, London, UK
[3]School of Computing, Newcastle University, Newcastle, UK
[4]UCL-Laboratory for Molecular Cell Biology, University College London, London, UK
[5]Instituto Gulbenkian de Ciência, Oeiras, Portugal

R.F.L. current address: Micrographia Bio, London, UK

Correlative light and volume electron microscopy (vCLEM) is a powerful imaging technique that enables the visualisation of fluorescently labelled proteins within their ultrastructural context on a subcellular level. Currently, expert microscopists find the alignment between acquisitions by manually placing landmarks on structures that can be recognised in both imaging modalities. The manual nature of the process severely impacts throughput and may introduce bias. This paper presents CLEM-Reg, a workflow that automates the alignment of vCLEM datasets by leveraging point cloud based registration techniques. Point clouds are obtained by segmenting internal landmarks, such as mitochondria, through a pattern recognition approach that includes deep learning. CLEM-Reg is a fully automated and reproducible vCLEM alignment workflow that requires no prior expert knowledge. When benchmarked against experts on three newly acquired vCLEM datasets using two EM technologies (FIB-SEM and SBF-SEM), CLEM-Reg achieves near expert-level registration performance. The datasets are made available in the EMPIAR and Biostudies public image archives for reuse in testing and developing multimodal registration algorithms by the wider community. A napari plugin integrating the algorithm is also provided to aid adoption by end-users. The source-code for CLEM-Reg and installation instructions can be found at `https://github.com/krentzd/napari-clemreg`.

Introduction

Correlative light and electron microscopy (CLEM) is a powerful imaging technique that seeks to capitalise on the advantages of light microscopy (LM) and electron microscopy (EM) while circumventing the drawbacks of each. This has made CLEM the imaging technique of choice to target rare and dynamic biological events that necessitate structural analysis at high resolution (1,2). Fluorescence microscopy (FM) is an LM imaging modality which generates contrast by tagging macromolecules in living cells and tissues with fluorescent proteins, enabling dynamic observation of their biological interactions. However, due to the diffraction limit of light, traditional FM cannot achieve a resolution better than around 200 nm, hindering fine structural details from being resolved (3). While super-resolution techniques can surpass this diffraction limit, such methods require specialised instruments, as well as specific sample preparation and imaging protocols, imposing additional constraints on the type of biological events that can be imaged (4). Moreover, as FM generally tags specific macromolecules, the surrounding structure cannot be imaged. EM addresses these limitations, achieving orders of magnitude higher resolution while revealing the underlying biological context (5) in exquisite detail, but at the cost of a smaller field-of-view (FOV). By harnessing the complementary information derived from the correlation of LM and EM, CLEM has led to a variety of biological discoveries, such as establishing the structure of tunnelling nanotubes in neurons (6), observing blood vessel fusion events in zebrafish (2) and localising tuberculosis bacteria in primary human cells (1).

One of the most common approaches for performing CLEM experiments is to sequentially image a sample using FM and then EM, necessitating a two-stage sample preparation process. First, relevant structures are

tagged with organic dyes or fluorescent proteins and an image stack through the sample volume is acquired using FM. The sample is immobilised with cross-linkers, either before or after the FM imaging, to conserve structural features. It is then stained with heavy-metal salts to introduce contrast, dehydrated, embedded in resin, and trimmed to the region of interest (ROI) (7). In volume EM (vEM), layers of the embedded sample are physically removed, and either the face of the block or the sections themselves are imaged to create a stack of serial EM images through the volume of the sample (5). This results in two corresponding image stacks, one from FM and one from EM, each containing complementary data from the same physical region of the sample, but typically imaged at non-identical orientations. As well as this orientation mismatch, the sample preparation and imaging processes can introduce both linear and non-linear deformations between the acquired FM and EM image volumes. Thus, to correlate the FM signal to the underlying structures visible in EM, the image volumes need to be brought into alignment. Due to the significant differences in resolution, contrast and FOV between FM and EM, this is a challenging task that cannot be directly approached with intensity-based methods that are routinely used for aligning data from visually similar modalities.

There are two general approaches to solving this problem. The first approach is to process one or both images such that they share a similar visual appearance, for example by directly converting across modalities (8–10) or by constructing a shared modality-agnostic representation (11,12). Once the processed image stacks are sufficiently similar in visual appearance, traditional intensity-based registration techniques, such as those employed in medical imaging (13), can be used to automatically align the two datasets. The second approach is to use a landmark-based method such as those implemented in software tools like BigWarp (14) and ec-CLEM (15). These tools rely on the manual identification of precise spatial regions visible in both modalities, for example, small sub-cellular structures or prominent morphological features. By manually placing a landmark at an identical physical position in each modality, spatial correspondences can be established. A set of these landmarks can then be used to compute the transformation between the image volumes (1), bringing them into alignment via an iterative optimisation process (16,17). Methods of automating the selection of corresponding landmarks across modalities have been investigated, for example determining cell centroids in LM and EM (18), or using semi-automated feature detection (15). However, such methods are often restricted to 2D or limited to relatively coarse alignment, requiring subsequent manual refinement. Moreover, deep learning-based methods that convert across modalities (8) or construct a shared modality-agnostic representation (12) rely on the availability of large amounts of aligned ground truth data. Due to the low throughput and required expertise of manual volume CLEM (vCLEM) alignment, generating such ground truth data is challenging.

A significant advantage of landmark-based approaches is the inherent sparsity of the representation, which substantially reduces memory and computational requirements compared to intensity-based registration techniques that must generally hold both image volumes in working memory. However, this manual landmark selection step is laborious and time-consuming, severely impacting throughput and potentially introducing bias, since the target structure is often directly used for registration. To avoid such biases, landmarks used for registration should be different from the target structures being studied wherever possible, taking care to ensure colour-correction between the channels in FM to avoid spectrally induced shifts in focal depth. Due to these limitations, robust and objective automation of landmark detection is highly desirable.

To achieve this aim, we have developed a workflow to segment mitochondria, chosen since they are abundant and typically well-distributed across cells, in both FM and EM images, in order to find matching structures across the modalities. Various segmentation approaches are routinely used in microscopy ranging from classical image processing techniques (9,19,20) to machine learning (21–23). Depending on the complexity and density of structures in the image at hand, different algorithms are appropriate. For instance, segmentation of objects like mitochondria in FM acquisitions can be performed by filtering and thresholding coupled with further downstream image processing (19), while automatically segmenting EM data typically requires advanced deep learning techniques. Common deep learning architectures like “U-Net” (24) can be trained locally from scratch to very good effect, but the burden of obtaining sufficient ground truth data presents a huge challenge, often requiring significant amounts of expert effort or crowdsourcing of manual annotations (23). Recently, however, pre-trained “generalist” models such as MitoNet (25), based upon a "Panoptic-DeepLab" architecture (26), are able to provide out-of-the-box performance levels for mitochondrial segmentation in EM that are sufficient for

many tasks, with the option to fine-tune on local data where necessary.

After the segmentation step in our workflow, points are then randomly sampled from the resulting mitochondria segmentations in both the FM and EM volumes, resulting in a "point cloud" for each modality. Point clouds are an attractive modality agnostic representation due to their inherent sparsity and the availability of a range of performant registration algorithms (16,27–29). However, unlike manually generated pairs of points, there is no guarantee of a one-to-one precise spatial correspondence between points in the different modalities. We find that the coherent point drift (CPD) (27) algorithm overcomes this limitation by casting the alignment task as a probability density estimation problem, thereby removing the constraint of strict point correspondences. The workflow is deployed as a plugin for the napari image viewer, giving users the option for a single-click end-to-end operation, or to fine-tune or entirely replace individual steps of the workflow, for example importing segmentations from alternative sources.

Results

Benchmark dataset acquisition. To assess the performance of CLEM-Reg against an expert, three benchmark vCLEM datasets (EMPIAR-10819, EMPIAR-11537 and EMPIAR-11666) of HeLa cells were acquired by microscopists from the Electron Microscopy Science Technology Platform at The Francis Crick Institute. Mitochondria (MitoTracker Deep Red), nucleus (Hoechst 33342), Golgi apparatus protein TGN46 (GFP-TGN46) and lysosomes (LysoTracker Red) in EMPIAR-10819 and EMPIAR-11666 and cell membrane (WGA) in EMPIAR-11537 were tagged so that the unbiased registration performance could be assessed using target structures. We also observed GFP-TGN46 labelling endosomes, as part of the trafficking process between the plasma membrane and the trans-Golgi network (30). After imaging the samples with a Zeiss Airyscan LSM900, two corresponding EM volumes (EMPIAR-10819 and EMPIAR-11537) with an isotropic voxel-size of 5 nm were acquired using a focused ion beam scanning electron microscope (FIB-SEM). The EM volume in EMPIAR-11666 was acquired using a serial block-face scanning electron microscope (SBF-SEM) with a voxel-size of 7 nm in XY and 50 nm in Z. The acquired images were pre-aligned following a routine pre-processing workflow in Fiji (**Materials and Methods**, **Fig. 1A**).

The CLEM-Reg pipeline. CLEM-Reg automatically registers vCLEM data by segmenting mitochondria in FM and EM that are then used to generate point clouds which are aligned with CPD, a state-of-the-art point cloud registration technique. The FM volume is then warped onto the EM volume using the found transformation (**Fig. 1B**). CLEM-Reg can be accessed via a napari plugin that allows users to automatically register vCLEM datasets with a single click of a button. The plugin includes a range of features, such as the option to delineate a corresponding ROI using the "Shapes" layer in napari, as well as the option to choose between rigid and non-linear registration. Parameters such as the FM segmentation settings, point cloud sampling density and the maximum number of iterations for the point cloud registration can be tuned (**Supp. Fig. 1**). The chosen parameters can be saved and later re-used to ensure reproducibility. Overlays can be directly exported from napari, as well as the initial and transformed point clouds for quality control purposes. Intermediate outputs like segmentation masks and sampled point clouds can be output directly to the napari viewer to aid troubleshooting. In this study, the performance of the CLEM-Reg plugin was assessed by an expert electron microscopist from the Electron Microscopy Science Technology Platform at The Francis Crick Institute on three vCLEM datasets (EMPIAR-10819, EMPIAR-11537 and EMPIAR-11666).

Segmenting internal landmarks. A promising approach to fully automate the CLEM alignment process is to automatically identify internal landmarks, speeding up the process and minimising opportunities for inadvertent subjective bias to occur. CLEM-Reg relies on the identification and segmentation of these common internal landmarks in both imaging modalities. For the purposes of this study, mitochondria were used as landmarks. To obtain segmentations in FM, an algorithm based on combining a 3D Laplacian of Gaussian (LoG) filter with dynamic thresholding to account for signal degradation at deeper imaging levels was developed. Users can adjust parameters such as the kernel size of the LoG and the relative segmentation threshold to adjust the segmentation output. After obtaining an initial segmentation mask, a user-defined size-based filter can be applied to remove spurious segmentations (**Fig. 2A)**.

The CLEM-Reg plugin allows users to directly generate EM segmentations using the pre-trained MitoNet deep learning model (**Fig. 2B**). It is also possible to run the pipeline on already obtained segmentations with the split registration functionality that implements individual steps of the CLEM-Reg pipeline. This flexibility allows users to choose different landmarks for registration or to make use of custom segmentation techniques that are more suited for a particular dataset.

Generating modality-agnostic point clouds from internal landmarks and registration. The alignment between the FM and EM segmentations can be inferred by sampling 3D point clouds from the previously obtained segmentation masks. This reduces the computational load for large datasets and allows for mistakes in the segmentation to be ignored by using a probabilistic registration algorithm, such as CPD (**Supp. Fig. 2**). The extraction of the coordinates of pixels from the exterior of the segmentation masks results in a 3D point cloud that captures its global architecture with an efficient representation due to the inherent sparsity of point clouds. The number of points in both point clouds can be modified by adjusting a user-defined binning parameter and downsampling factor (**Fig. 3A**). Increasing any of these two parameters speeds up the registration potentially by orders of magnitude. For instance, reducing the point sampling frequency from 1/16 to 1/256 with a fixed voxel size leads to a 19-fold decrease (from 33.7 min to 1.8 min) in registration time with no change in registration error (**Supp. Fig. 1)**.

After sampling, the point clouds are registered using either rigid CPD or non-linear Bayesian Coherent Point Drift (BCPD) (**Fig. 3B**) (27). Note that these probabilistic methods are necessary, as opposed to algorithms like Iterative Closest Point (ICP) (29), since the point sampling across modalities means there are no explicitly paired points. The choice of registration algorithm will depend on the expected deformations between the FM and EM volumes, as well as computational constraints. In general, rigid CPD is faster and computationally less expensive than non-linear BCPD. For certain datasets it might also be beneficial to combine rigid CPD and non-linear BCPD which can be achieved using the split registration functionality in the CLEM-Reg plugin.

Warping FM volume to obtain CLEM overlay. Once registered, the found transformation to register the point clouds is used to warp the FM volume onto the EM volume. This step is fast for rigid transformations but slower for non-linear warping. CLEM-Reg implements 3D non-linear thin-plate spline warping (31) that uses the initially sampled and registered FM point clouds as control points. The runtime of the thin- plate spline warping can be controlled by adjusting the interpolation order and size of the approximate grid directly in the plugin. Since thin-plate spline warping is an expensive algorithm, CLEM-Reg also implements the option to sequentially warp sub-volumes. The number of sub-volumes can be set by the user with the sub-division factor parameter. This extends the runtime of the warping step, but reduces the required random access memory (RAM). To obtain overlays from the three benchmark datasets, it was found that rigid alignment alone was sufficient to obtain good overlays on experimental datasets (**Fig. 4A-C**).

Assessing CLEM-Reg performance against experts. To provide a benchmark for comparison, an expert microscopist manually aligned the FM and EM volumes using a global affine registration in BigWarp. The automated alignments obtained from CLEM-Reg were compared to this gold-standard (**Fig. 5**). The CLEM-Reg alignments were performed by segmenting the mitochondria as landmarks in both the FM and EM volumes. To assess the quality of the alignments, a target organelle not used for registration was identified in each dataset (lysosomes in EMPIAR-10819 and EMPIAR-11666, and endosomes in EMPIAR-11537) and the overlap of these structures was measured. To quantify this overlap, the target organelles were manually segmented in the EM volume and automatically segmented in the FM volume by thresholding, and the overlap of the resulting binary images was measured (**Materials and Methods**).

Typical overlap metrics like Intersection over Union (IoU) and Dice coefficient are implicitly designed for situations where the objects being compared are expected to be of the same size. However, due to the inherently lower resolution of FM imaging in comparison to EM, such overlap metrics tend to underestimate the agreement, in a size-dependent manner, when the objects are of different sizes. For example, a 10 nm object in EM would appear as ~200 nm in FM due to the diffraction limit, resulting in a near-zero IoU even in the case that their centres are perfectly aligned. To combat this effect, we instead use the true positive rate (TPR), effectively measuring the fraction of pixels of the EM segmentation that are within the FM segmentation, with a

maximum value of 1.0 when the EM segmentation lies entirely within the FM segmentation. CLEM-Reg is able to match the fluorescent signal to target structures in EM with $TPR_{EMPIAR-10819}$=0.48±0.22, $TPR_{EMPIAR-11537}$=0.74±0.18 and $TPR_{EMPIAR-11666}$=0.67±0.22. However, the fraction of fluorescent signal correctly overlaid to the corresponding segmentation is higher for the manually aligned datasets ($TPR_{EMPIAR-10819}$=0.66±0.12, $TPR_{EMPIAR-11537}$=0.83±0.12 and $TPR_{EMPIAR-11666}$=0.82±0.08, **Fig. 5**).

CLEM-Reg plugin for napari. Napari is an open-source multi-dimensional image viewer for Python that allows third-parties to develop plugins which enable the addition of custom features and functionality tailored to their specific needs (32). The CLEM-Reg plugin was developed in napari due to the increasing adoption of Python in the bioimage analysis community and its seamless integration of state-of-the-art image processing packages and deep learning libraries. Furthermore, the "Points" and "Labels" layers support visualisation of the point clouds and segmentations used in the CLEM-Reg workflow.

The default operation of the plugin is that, once datasets are loaded, it is possible to initiate an alignment via a single click in the GUI (**Fig. 6A**). However, recognising that the ability for end-users to fine-tune their results can be crucial, the plugin also includes the option to display intermediate steps of the full pipeline, as well as allow each step to be executed individually (**Fig 6B**). By utilising the "Layers" functionality of napari, it is also possible to inject intermediate results from external sources, for example an FM segmentation method from a different plugin, or a pre-existing EM segmentation mask. The resulting "Labels" layer can then be set as an input to subsequent steps in CLEM-Reg. Similarly, several different parameter configurations for a given step can be explored without needing to run the entire pipeline each time.

Discussion

This study introduces CLEM-Reg, an automated and unbiased end-to-end vCLEM registration framework. It can be easily accessed by end-users through a dedicated plugin for napari. The proposed registration approach relies on extracting landmarks with a pattern recognition approach that includes deep learning. After segmenting internal landmarks in FM and EM, point clouds are sampled and preprocessed to obtain a memory-efficient, modality-agnostic representation. By using state-of-the-art point cloud registration methods, a mapping between the two image volumes is found, with which the FM acquisition is warped onto the EM volume to obtain a final CLEM overlay. CLEM-Reg achieves near expert-level performance on three benchmark CLEM datasets.

Nevertheless, certain limitations remain in regard to the automated organelle segmentation in EM which directly impacts the alignment performance. While MitoNet was shown to perform well on the three vCLEM datasets shown in this study, this can not be guaranteed for all EM volumes. One approach to address this potential limitation is to make use of transfer learning, a deep learning method in which already trained models are briefly retrained on a smaller dataset to increase performance. While training deep neural networks previously required access to GPUs and fluency in programming languages such as Python, open-source projects like ZeroCostDL4Mic (33), DeepImageJ (33), Imjoy (35) or DL4MicEverywhere (36) are rapidly removing these barriers by offering frameworks that leverage freely available computing resources for training, enabling GUI-driven interaction with a range of deep learning architectures. Open source and community-driven model libraries such as the Bioimage Model Zoo (37), which allow easy sharing of pre-trained deep learning models, are another important resource in this context.

The size of bioimage datasets can also cause challenges for automated alignment. More specifically, downsampling of the FM and EM datasets is generally required prior to execution. Despite this, downsampling may still be insufficient for very large datasets. To circumvent this limitation, the use of next-generation file formats (38) and tools such as Dask (39) could enable the computation of datasets that exceed available memory by chunking, thus allowing users to run the CLEM-Reg workflow using full resolution data on conventional computers without encountering memory issues. Additionally, the CLEM-Reg workflow relies heavily on two packages Scipy (40) and scikit-image (41), for tasks such as FM segmentation and point sampling of the FM and EM segmentation. However, these packages do not natively provide GPU acceleration, thus not benefitting from available compute resources for users in possession of GPUs. As such, future work to

incorporate GPU acceleration into the CLEM-Reg workflow using Python libraries such as Cupy (42) or clEsperanto (31) could reduce processing time. This optimisation has the potential to significantly accelerate multiple steps within the CLEM-Reg workflow.

While CLEM-Reg performs near expert-level registration, it is important to appreciate that the acquired benchmark CLEM datasets were obtained with vEM techniques that image the block face, which have less nonlinear deformations and sectioning artefacts that disrupt the registration of serial EM images than TEM or SEM-based array tomography techniques. Thus, further development is required for application of CLEM-Reg to datasets acquired by serial section TEM and array tomography in the SEM.

It is anticipated that the ability to automate the accurate alignment of large vCLEM datasets will provide very powerful training data for new and upcoming deep learning approaches. For example, by providing large amounts of functional labelling of ultrastructure, the requirement for dense manual annotations of very large datasets will be reduced.

While CLEM-Reg has been demonstrated to register single-cell scale data via the segmentation of mitochondria, the core of the alignment workflow is agnostic to the landmarks being segmented. As such, the method could be used at different scales or across different modalities as long as certain conditions are met. The main conditions are the existence of a segmentation algorithm for the same structure in both modalities, and that those structures are numerous and distributed throughout the volume to be aligned. For example, nucleus segmentation algorithms in EM and FM could be used to derive point clouds that are capable of registering tissue-scale volumes for vCLEM, or even nucleus segmentation in X-ray microscopy could be aligned to either EM or FM (or both). With the rapid development of novel AI-based segmentation methods (44), it is expected that the range of suitable target structures will continue to grow, broadening the applicability of the method.

Materials and Methods

Cell model. Human cervical cancer epithelial cells (HeLa) were obtained from the Cell Services Science Technology Platform at The Francis Crick Institute and originated from the American Type Culture Collection (ATCC; CCL-2).

CLEM data acquisition. HeLa cells were maintained in Dulbecco's Modified Eagle Medium (Gibco) supplemented with 10% foetal bovine serum at 37°C and 5% CO_2. 150K cells were seeded in 35 mm photo-etched glass-bottom dishes (MaTtek Corp). At 24 h, cells were transfected with 0.1 µg of GFP-TGN46 construct per dish, using Lipofectamine LTX and PLUS reagent (Invitrogen) in Opti-MEM medium (Gibco) as recommended by the manufacturer. 300 µl of transfection mix were added to 2 ml antibiotic-free medium and incubated overnight. At 36 h, cells were stained with LysoTracker red (100 nM) or Wheat Germ Agglutinin (WGA) Alexa Fluor 594 (5 µg/ml) and MitoTracker deep red FM (100 nM) for 10 min in HBSS (WGA) or 30 min in DMEM (lysotracker and mitotracker). Hoechst 33342 1 µg/µl was added in the last 5 min of the tracker incubation. All probes were from Molecular Probes (Thermo Fisher Scientific). Cells were then washed 3 times with 0.1 M phosphate buffer (PB) pH 7.4 and fixed with 4% (v/v) formaldehyde (Taab Laboratory Equipment Ltd) in 0.1 M PB pH 7.4 for 15 min. Cells were washed twice and imaged in 0.1 M PB on an AxioObserver 7 LSM900 with Airyscan 2 microscope with Zen 3.1 software (Carl Zeiss Ltd). Cells were first mapped with a 10x objective (NA 0.3) using brightfield microscopy to determine their position on the grid and tile scans were generated. The cells of interest were then imaged at high resolution in Airyscan mode with a 63x oil objective (NA 1.4). A smart setup was used to set up the imaging conditions. A sequential scan of each channel was used to limit crosstalk and z-stacks were acquired throughout the whole volume of the cells. The samples were then processed using a Pelco BioWave Pro+ microwave (Ted Pella Inc, Redding, USA) and following a protocol adapted from the National Centre for Microscopy and Imaging Research protocol (45). Each step was performed in the Biowave, except for the phosphate buffer and water wash steps, which consisted of two washes on the bench followed by two washes in the Biowave without vacuum (at 250 W for 40 s). All the chemical incubations were performed in the Biowave for 14 min under vacuum in 2 min cycles alternating with/without 100 W power. The SteadyTemp plate was set to 21°C unless otherwise stated. In brief, the samples were fixed again in 2.5% (v/v) glutaraldehyde (TAAB) / 4% (v/v) formaldehyde in 0.1 M PB. The cells were

then stained with 2% (v/v) osmium tetroxide (TAAB) / 1.5% (v/v) potassium ferricyanide (Sigma), incubated in 1% (w/v) thiocarbohydrazide (Sigma) with SteadyTemp plate set to 40°C, and further stained with 2% osmium tetroxide in ddH2O (w/v). The cells were then incubated in 1% aqueous uranyl acetate (Agar Scientific) with a SteadyTemp plate set to 40°C, and then washed in dH2O with SteadyTemp set to 40°C. Samples were then stained with Walton's lead aspartate with SteadyTemp set to 50°C, and dehydrated in a graded ethanol series (70%, 90%, and 100%, twice each), at 250 W for 40 s without vacuum. Exchange into Durcupan ACM® resin (Sigma) was performed in 50% resin in ethanol, followed by 4 pure Durcupan steps, at 250 W for 3 min, with vacuum cycling (on/off at 30 s intervals), before polymerisation at 60 °C for 48 h.

Focused ion beam scanning electron microscopy (FIB-SEM) data was collected using a Crossbeam 540 FIB-SEM with Atlas 5 for 3D tomography acquisition (Carl Zeiss Ltd). A segment of the resin-embedded cell monolayer containing the cell of interest was trimmed out, and the coverslip was removed using liquid nitrogen prior to mounting on a standard 12.7 mm SEM stub using silver paint and coating with a 10 nm layer of platinum. The region of interest was relocated by briefly imaging through the platinum coating at an accelerating voltage of 10 kV and correlating to previously acquired fluorescence microscopy images. On completion of preparation of the target ROI for Atlas-based milling and tracking, images were acquired at 5 nm isotropic resolution throughout the cell of interest, using a 6 µs dwell time. During acquisition, the SEM was operated at an accelerating voltage of 1.5 kV with 1.5 nA current. The EsB detector was used with a grid voltage of 1.2 kV. Ion beam milling was performed at an accelerating voltage of 30 kV and a current of 700 pA.

For serial block face scanning electron microscopy (SBF-SEM), the block was trimmed to a small trapezoid, excised from the resin block, and attached to an SBF-SEM specimen holder using conductive epoxy resin. Prior to the commencement of an SBF-SEM imaging run, the sample was coated with a 2 nm layer of platinum to further enhance conductivity. SBF-SEM data was collected using a 3View2XP (Gatan) attached to a Sigma VP SEM (Carl Zeiss Ltd). Inverted backscattered electron images were acquired through the entire extent of the region of interest. For each of the 133 consecutive 50 nm slices needed to image the cell in its whole volume, a low resolution overview image (horizontal frame width 150.58 µm; pixel size of 75 nm; 2 µs dwell time) and a high resolution image of the cell of interest (horizontal frame width 63.13 µm; pixel size of 7 nm; using a 3 µs dwell time) were acquired. The SEM was operated in a high vacuum with focal charge compensation on. The 30 µm aperture was used, at an accelerating voltage of 2 kV.

Pre-processing for CLEM registration. The Airyscan data was first processed in Zen software using the z-stack alignment tool to correct z-shift. The settings used were the highest quality, translation and linear interpolation, with the mitotracker channel as a reference. The .czi file was then opened in Fiji (46) and saved as .tif. For EMPIAR-11666, the MitoTracker channel was further processed using Yen's thresholding method (47) to remove overexposed punctate artefacts affecting FM segmentation.

After initial registration with template matching by normalised cross-correlation in Fiji (https://sites.google.com/site/qingzongtseng/template-matching-ij-plugin), the FIB-SEM images were contrast normalised as required across the entire stack and converted to 8-bit grayscale. To fine-tune image registration, the alignment to the median smoothed template method was applied (48). The aligned XY 5 nm FIB-SEM stack was opened in Fiji and resliced from top to bottom (i.e., to XZ orientation), and rotated to roughly match the previously acquired Airyscan data. Finally, the EM volume was binned by a factor of 4 in X, Y and Z using Fiji resulting in an isotropic voxel-size of 20 nm.

For the SBF-SEM data, only minor adjustments in image alignment were needed and were done using TrakEM2 in Fiji (49).

Computational resources. The algorithms described here were primarily developed and tested on a portable machine running on a Linux distribution with the following specifications: 64 GB RAM, Intel Core i7-11850H 2.50GHz × 16 CPU and GeForce RTX 3080 GPU. An additional workstation with the following specs was also used for testing: 256 GB RAM, Dual Intel Xeon (14 CPU each) 2.6GHz, Quadro M600 16GB GPU.

FM segmentation with thresholded Laplacian of Gaussian. To ensure the FM pixel values were in the range of 0 and 1, min-max normalisation of the following formula was used, where x is the FM volume, x_{min} is the lowest pixel value in the volume and x_{max} is the largest pixel value in the volume and x_{scaled} is the resulting scaled volume $x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$. In the subsequent step, the scaled FM volume undergoes a LoG filtering process, requiring one tunable parameter σ. This process involves convolving a 1D Difference of Gaussians (DoG) across the X, Y and Z planes of the volume. Marr and Hildreth (51) demonstrated that a reasonable approximation of LoG can be attained by maintaining a ratio of $\frac{\sigma_2}{\sigma_1} = 1.6$ when applying the DoG. A default value of $\sigma = 3.0$ was used, thus after applying the LoG ratio, $\sigma_1 = 3.0$ and $\sigma_2 = 4.8$. A dynamic threshold based on a relative user-defined threshold and the mean intensity of a given slice was then applied to each image in the resulting volume to account for inconsistencies arising due to attenuation of the signal at deeper imaging levels. The default relative threshold was set to $T = 1.2$. Lastly, to reduce the number of spurious segmentations, a size-based filter was then applied. This was achieved by using a 3D connected components algorithm from the Python package cc3d 3.2.1 (52) which removes components below a user-defined size threshold with a default value set to remove all segmentations with a size below the 5th and above the 95th percentile.

EM segmentation with MitoNet. The mitochondria within the EM volume were segmented using the pre-trained deep learning model, MitoNet (53). The MitoNet model using its default parameters was applied to the EM volume, resulting in instance segmentation of the mitochondria.

To improve mitochondrial segmentation of the SBF-SEM data, the aligned stack was processed for contrast-limited adaptive histogram equalisation (50) using the Enhance Local Contrast (CLAHE) plugin in Fiji, using the following parameters: blocksize = 127; histogram bins = 256; maximum slope = 2, 3 and 10; mask = "*None*"; fast. Each CLAHE stack (from 3 maximum slope values) was segmented using MitoNet and the masks were merged by summation before point cloud sampling.

Point cloud sampling of the EM and FM segmentation. The method employed for producing point clouds from the EM and FM volumes uses the same standardised procedure. Firstly, the FM and EM segmentation masks were resampled to obtain isotropic voxels (EMPIAR-10819 and EMPIAR-11537: 20 nm; EMPIAR-11666: 50 nm). Then, a Canny edge filter (54) with a default value of σ = 1.0 was applied to each binary segmentation slice. This process ensures that only points on the edge of the outer mitochondrial membrane are randomly sampled. Then, every pixel along the membrane was identified as a potential point to be used within the point cloud, but due to the computationally expensive nature of point cloud registration algorithms, subsequent downsampling was carried out.

Point cloud downsampling and registration with CPD and BCPD. After sampling, the point clouds were downsampled, binned and filtered by removing statistical outliers to reduce memory requirements and noise. Downsampling of the point clouds was performed by a uniform downsampling function which takes the point cloud and user-defined sampling frequency $1/k$ as the input and samples every k^{th} point in the point cloud, starting with the 0th point, then the kth point, then the $k + k^{th}$ point and so on. Following the initial downsampling, a binning step was used which employs a regular voxel grid to create a downsampled point cloud from the input point cloud. There are two primary steps of the binning starting with an initial partitioning of the point cloud into a set of voxels. Subsequently, each occupied voxel generates exactly one point through the computation of the average of all the points enclosed within it, resulting in a binned version of the original point cloud. For registration, the Python package probreg 0.3.6 (55) was used, as it implements both the CPD (28) and BCPD (27) algorithm with a unified API. After applying either CPD or BCPD the output was a globally registered point cloud of the FM to the EM which is then used for the warping of the image volumes.

Warping of image volumes. The transformation matrix found from the global point cloud registration was applied to the source image volume using the affine transform implementation provided by SciPy 1.10.1 (40), a

powerful open-source Python library that provides a range of functions for mathematics, science, and engineering. This implementation makes use of inverse projection to interpolate the intensity values of the warped image thus requiring an inverted transformation matrix which is found with NumPy's 1.24.2 (56) linear algebra module. Interpolation of the warped image was achieved with higher-order spline interpolation. Non-linear image warping was achieved with thin-plate spline deformation which finds a mapping between a set of control-points $f: (x, y, z) \rightarrow (x', y', z')$ such that the deformed source points match the target points as closely as possible while ensuring that the image between these control-points remains smooth. As there are no open-source implementations for volumetric thin-plate spline warping available for Python, a custom script was implemented based on (31). To reduce computational overhead, the FM volume was nonlinearly warped as eight individual chunks. After warping, the EMPIAR-11666 FM image volume was resampled in XY to obtain a pixel size of 28 nm.

Benchmarking of CLEM-Reg against experts. For expert manual registration, image stacks from FM and EM were manually aligned to each other using the BigWarp plugin of the Fiji framework, with the EM stack set as "target" and the FM stack as "moving" dataset. Corresponding points were placed on mitochondria on both datasets and throughout the whole volume of the cell. An affine transformation was applied to the FM data and the transformed dataset was merged with the EM data to produce the final overlay.

Due to the inherent differences in appearance between the two imaging modalities, direct intensity-based quantification of the registration performance is not possible. Therefore, a metric measuring the target structure agreement was used to assess the resulting overlays. Firstly, target structures (lysosomes in EMPIAR-10819 and EMPIAR-11666 and endosomes in EMPIAR-11537) were manually segmented in 3D in the EM volume in TrakEM2. These lysosomes and endosomes were then cropped with a bounding box from the corresponding manual- or CLEM-Reg-warped FM volumes and then segmented using the Otsu thresholding method (57) to generate the correlating FM segmentation. The intersection between the FM and EM segmentation was computed and divided by the number of pixels in the EM segmentation. This measure is also known as the true positive rate (TPR). This was used in preference to familiar measures such as IoU due to the very different imaging resolutions of the two modalities. To capture volume-wide effects between the imaging volumes, target structures in different areas of the cell were chosen.

Plugin development and deployment. The plugin was developed for Python 3.9 and is compatible with napari 0.4.17 (32). It integrates the CLEM-Reg workflow in the form of a single widget running it end-to-end and as a set of individual widgets, each carrying out an individual step (split registration functionality). To distribute the plugin and make it accessible from the napari interface, the project is packaged using the cookiecutter plugin template provided by napari. The plugin is available for installation via the package installer for Python (pip) in the command line under the project name napari-clemreg. The plugin can also be installed via the built-in plugin menu which lists all plugins currently available on napari-hub The source-code is available as a GitHub repository alongside a detailed project description, installation instructions and user guide. The EMPIAR-10819 dataset can also directly be loaded as sample data from the plug-in.

DATA AND CODE AVAILABILITY

The datasets used in this study have been deposited at EMPIAR and Biostudies. EMPIAR-10819: EM (https://www.ebi.ac.uk/empiar/EMPIAR-10819/) FM (https://www.ebi.ac.uk/biostudies/bioimages/studies/S-BSST707). EMPIAR-11537: EM (https://www.ebi.ac.uk/empiar/EMPIAR-11537/) FM (https://www.ebi.ac.uk/biostudies/bioimages/studies/S-BSST1075). EMPIAR-11666: EM (https://www.ebi.ac.uk/empiar/EMPIAR-11666/) FM (https://www.ebi.ac.uk/biostudies/bioimages/studies/S-BSST1175). Code for CLEM-Reg (under MIT licence) is available on GitHub (https://github.com/krentzd/napari-clemreg).

ACKNOWLEDGEMENTS

This work was supported by The Francis Crick Institute which receives its core funding from Cancer Research UK (CC1076), the UK Medical Research Council (CC1076), and the Wellcome Trust (CC1076). D.K. was funded by The Francis Crick Institute and the Pasteur-Paris University International doctoral program (PPU). R.F.L. was supported by a Medical Research Council Skills development fellowship (MR/T027924/1). R.H. is supported by the Gulbenkian Foundation (Fundação Calouste Gulbenkian) and received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement number 101001332), the European Union through the Horizon Europe program (AI4LIFE project with grant agreement 101057970-AI4LIFE, and RT-SuperES project with grant agreement 101099654-RT-SuperES to R.H.), the European Molecular Biology Organization (EMBO) Installation Grant (EMBO-2020-IG4734) and the Chan Zuckerberg Initiative Visual Proteomics Grant (vpi-0000000044 with doi:10.37921/743590vtudfp). Views and opinions expressed are those of the authors only and do not necessarily

reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. Matt Russell (Electron Microscopy Science Technology Platform, The Francis Crick Institute and Centre for Ultrastructural Imaging, King's College London) collected preliminary CLEM data used in testing.

CONTRIBUTIONS

DK, RFL, RH, LMC and MLJ developed the initial idea for this project. DK conceived CLEM-Reg and directed this project. DK, ME, MCD, MLJ and LMC wrote the manuscript. DK and ME developed the source-code. RFL collected preliminary CLEM data. MCD and CJP collected the benchmark CLEM datasets (MCD: FM and SBF-SEM, CJP: FIB-SEM). MCD performed manual registrations, segmentations and user testing. All authors contributed to the revisions of the manuscript.

Bibliography

1. Russell, M. R. G. *et al.* 3D correlative light and electron microscopy of cultured cells using serial blockface scanning electron microscopy. *J. Cell Sci.* **130,** 278–291 (2017).
2. Armer, H. E. J. *et al.* Imaging Transient Blood Vessel Fusion Events in Zebrafish by Correlative Volume Electron Microscopy. *PLOS ONE* **4,** e7716 (2009).
3. Rino, J., Braga, J., Henriques, R. & Carmo-Fonseca, M. Frontiers in fluorescence microscopy. *Int. J. Dev. Biol.* **53,** 1569–1579 (2009).
4. Culley, S., Pereira, P. M., Laine, R. F. & Henriques, R. in *Imaging Cells Anim. Vivo* (CRC Press, 2020).
5. Peddie, C. J. & Collinson, L. M. Exploring the third dimension: Volume electron microscopy comes of age. *Micron* **61,** 9–19 (2014).
6. Sartori-Rupp, A. *et al.* Correlative cryo-electron microscopy reveals the structure of TNTs in neuronal cells. *Nat. Commun.* **10,** 342 (2019).
7. Zhou, W., Apkarian, R., Wang, Z. L. & Joy, D. in *Scanning Microsc. Nanotechnol.* 1–40 (Springer, 2006).
8. Seifert, R. *et al.* DeepCLEM: automated registration for correlative light and electron microscopy using deep learning. *F1000Research* **9,** 1275 (2020).
9. Cao, T. *et al.* Multi-modal registration for correlative microscopy using image analogies. *Med. Image Anal.* **18,** 914–926 (2014).
10. Heinrich, L. *et al.* Whole-cell organelle segmentation in volume electron microscopy. *Nature* **599,** 141–146 (2021).
11. Acosta, B. M. T., Bouthemy, P. & Kervrann, C. A common image representation and a patch-based search for correlative light-electron-microscopy (CLEM) registration. in *2016 IEEE 13th Int. Symp. Biomed. Imaging ISBI* 257–260 (2016). doi:10.1109/ISBI.2016.7493258
12. Pielawski, N. *et al.* CoMIR: Contrastive Multimodal Image Representation for Registration.
13. Klein, S., Staring, M., Murphy, K., Viergever, M. A. & Pluim, J. P. W. elastix: A Toolbox for Intensity-Based Medical Image Registration. *IEEE Trans. Med. Imaging* **29,** 196–205 (2010).
14. Pietzsch, T., Saalfeld, S., Preibisch, S. & Tomancak, P. BigDataViewer: visualization and processing for large image data sets. *Nat. Methods* **12,** 481–483 (2015).
15. Paul-Gilloteaux, P. *et al.* eC-CLEM: flexible multidimensional registration software for correlative microscopies. *Nat. Methods* **14,** 102–103 (2017).
16. Yang, J., Li, H., Campbell, D. & Jia, Y. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **38,** 2241–2254 (2016).
17. Arun, K. S., Huang, T. S. & Blostein, S. D. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-9,** 698–700 (1987).
18. Nam, D. *et al.* Feature-based registration for correlative light and electron microscopy images. in *2014 IEEE Int. Conf. Image Process. ICIP* 3567–3571 (2014). doi:10.1109/ICIP.2014.7025724
19. Malpica, N. *et al.* Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry* **28,** 289–297 (1997).
20. Berg, S. *et al.* ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods* **16,** 1226–1232 (2019).
21. Schmidt, U., Weigert, M., Broaddus, C. & Myers, G. Cell Detection with Star-convex Polygons. *ArXiv180603535 Cs* **11071,** 265–273 (2018).
22. Weigert, M., Schmidt, U., Haase, R., Sugawara, K. & Myers, G. Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy. *2020 IEEE Winter Conf. Appl. Comput. Vis. WACV* 3655–3662 (2020). doi:10.1109/WACV45572.2020.9093435
23. Spiers, H. *et al.* Deep learning for automatic segmentation of the nuclear envelope in electron microscopy data, trained with volunteer segmentations. *Traffic* **22,** 240–253 (2021).
24. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. in *Med. Image Comput. Comput.-Assist. Interv. – MICCAI 2015* (eds. Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F.) 234–241 (Springer International Publishing, 2015). doi:10.1007/978-3-319-24574-4_28
25. Conrad, R. & Narayan, K. *Instance segmentation of mitochondria in electron microscopy images with a generalist deep learning model*. (Cell Biology, 2022). doi:10.1101/2022.03.17.484806
26. Cheng, B. *et al.* Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. at http://arxiv.org/abs/1911.10194 (2020)
27. Hirose, O. A Bayesian Formulation of Coherent Point Drift. *IEEE Trans. Pattern Anal. Mach. Intell.* 1–1 (2020). doi:10.1109/TPAMI.2020.2971687
28. Myronenko, A. & Song, X. Point-Set Registration: Coherent Point Drift. *IEEE Trans. Pattern Anal. Mach. Intell.* **32,** 2262–2275 (2010).

29. Besl, P. J. & McKay, N. D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14,** 239–256 (1992).
30. Ponnambalam, S. *et al.* Primate homologues of rat TGN38: primary structure, expression and functional implications.
31. Bookstein, F. L. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* **11,** 567–585 (1989).
32. Nicholas Sofroniew *et al. napari/napari: 0.4.2rc0*. (Zenodo, 2020). doi:10.5281/zenodo.4294808
33. von Chamier, L. *et al.* Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat. Commun.* **12,** 2276 (2021).
34. Gómez-de-Mariscal, E. *et al.* DeepImageJ: A user-friendly environment to run deep learning models in ImageJ. *Nat. Methods* **18,** 1192–1195 (2021).
35. Ouyang, W., Mueller, F., Hjelmare, M., Lundberg, E. & Zimmer, C. ImJoy: an open-source computational platform for the deep learning era. *Nat. Methods* **16,** 1199–1200 (2019).
36. Hidalgo-Cenalmor, I., Pylvänäinen, J. W., Ferreira, M. G., Russell, C. T. & Arganda-Carreras, I. DL4MicEverywhere: Deep learning for microscopy made flexible, shareable, and reproducible.
37. Ouyang, W. *et al.* BioImage Model Zoo: A Community-Driven Resource for Accessible Deep Learning in BioImage Analysis.
38. Moore, J. *et al.* OME-NGFF: a next-generation file format for expanding bioimaging data-access strategies. *Nat. Methods* **18,** 1496–1498 (2021).
39. Rocklin, M. Dask: Parallel Computation with Blocked algorithms and Task Scheduling. in 126–132 (2015). doi:10.25080/Majora-7b98e3ed-013
40. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17,** 261–272 (2020).
41. Walt, S. van der *et al.* scikit-image: image processing in Python. *PeerJ* **2,** e453 (2014).
42. Okuta, R., Unno, Y., Nishino, D., Hido, S. & Loomis, C. CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations.
43. Haase, R. *et al.* Interactive design of GPU-accelerated Image Data Flow Graphs and cross-platform deployment using multi-lingual code generation.
44. Kirillov, A. *et al.* Segment Anything. at http://arxiv.org/abs/2304.02643 (2023)
45. Deerinck, T. J., Bushong, E. A., Thor, A. & Ellisman, M. H. A new protocol for preparation of biological specimens for serial block-face SEM. *NCMIR METHODS 3D EM* (2010). at <https://ncmir.ucsd.edu/sbem-protocol>
46. Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9,** 676–682 (2012).
47. Yen, J.-C., Chang, F.-J. & Chang, S. A new criterion for automatic multilevel thresholding.
48. Hennies, J. *et al.* AMST: Alignment to Median Smoothed Template for Focused Ion Beam Scanning Electron Microscopy Image Stacks. *Sci. Rep.* **10,** 2004 (2020).
49. Cardona, A. *et al.* TrakEM2 Software for Neural Circuit Reconstruction. *PLOS ONE* **7,** e38011 (2012).
50. Pizer, S. M., Johnston, R. E., Ericksen, J. P., Yankaskas, B. C. & Muller, K. E. Contrast-limited adaptive histogram equalization: speed and effectiveness. in *1990 Proc. First Conf. Vis. Biomed. Comput.* 337–345 (1990). doi:10.1109/VBC.1990.109340
51. Marr, D. & Hildreth, E. Theory of edge detection. *Proc. R. Soc. Lond. B Biol. Sci.* **207,** 187–217 (1980).
52. Silversmith, W. *cc3d: Connected components on multilabel 3D & 2D images.* (Zenodo, 2021). doi:10.5281/zenodo.5719536
53. Conrad, R. & Narayan, K. *Empanada MitoNet model files*. (Zenodo, 2022). doi:10.5281/zenodo.6861565
54. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8,** 679–698 (1986).
55. Kenta-Tanaka et al. *probreg*. (2019). at <https://probreg.readthedocs.io/en/latest/>
56. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585,** 357–362 (2020).
57. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **9,** 62–66 (1979).

Figures

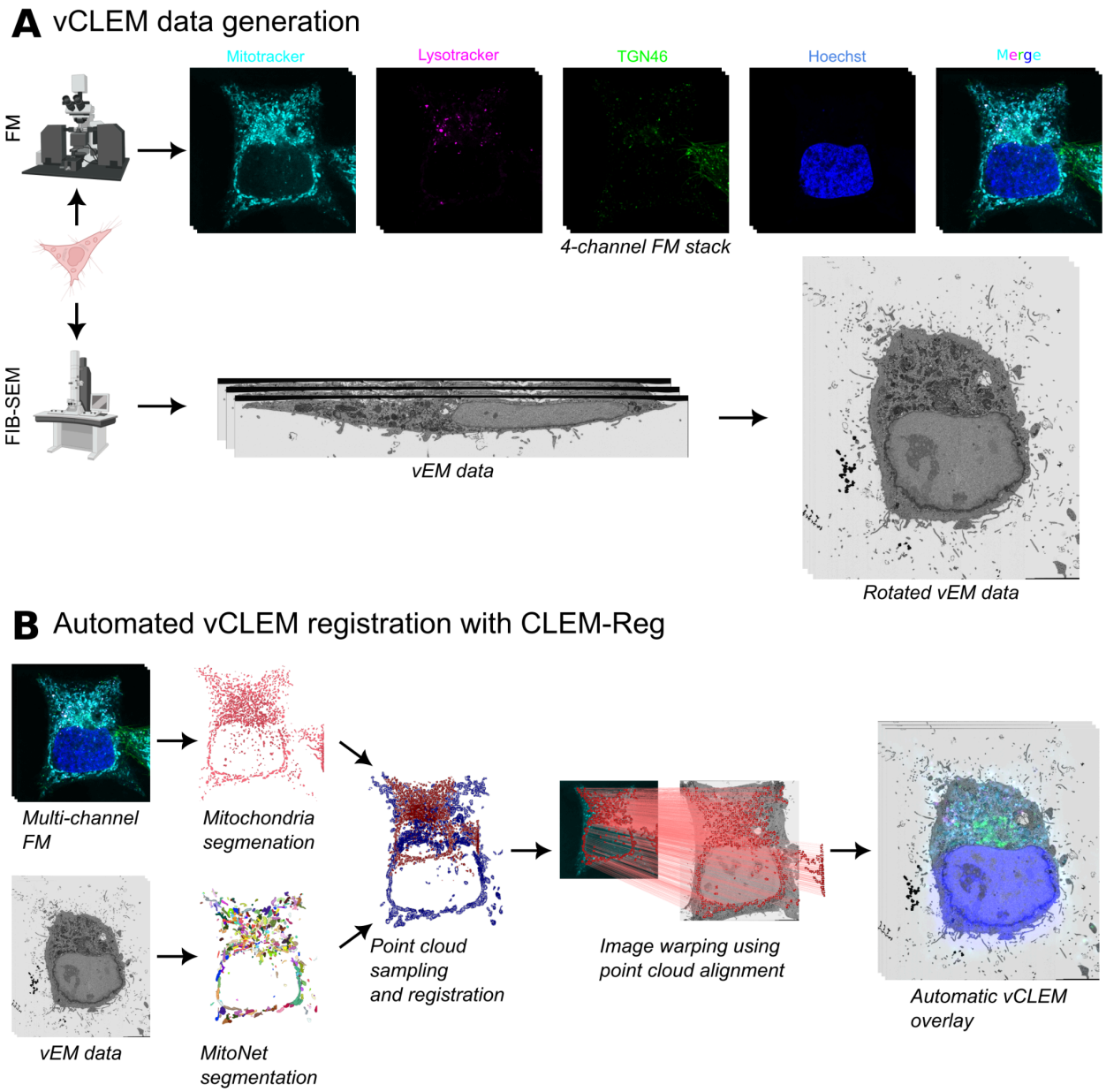

Figure 1. Volume CLEM data generation and CLEM-Reg algorithm. (A) Obtaining a vCLEM dataset consists of acquiring a FM and vEM image of the same sample. The two image stacks are manually aligned by identifying landmarks in both modalities and computing a transform to warp the FM stack onto the vEM data. (B) CLEM-Reg fully automates the registration step for vCLEM datasets by first segmenting mitochondria in both image modalities, sampling point clouds from these segmentations and registering them. Once registered, the point cloud alignment is used to warp the FM stack onto the vEM data. All data visualisations generated with napari.

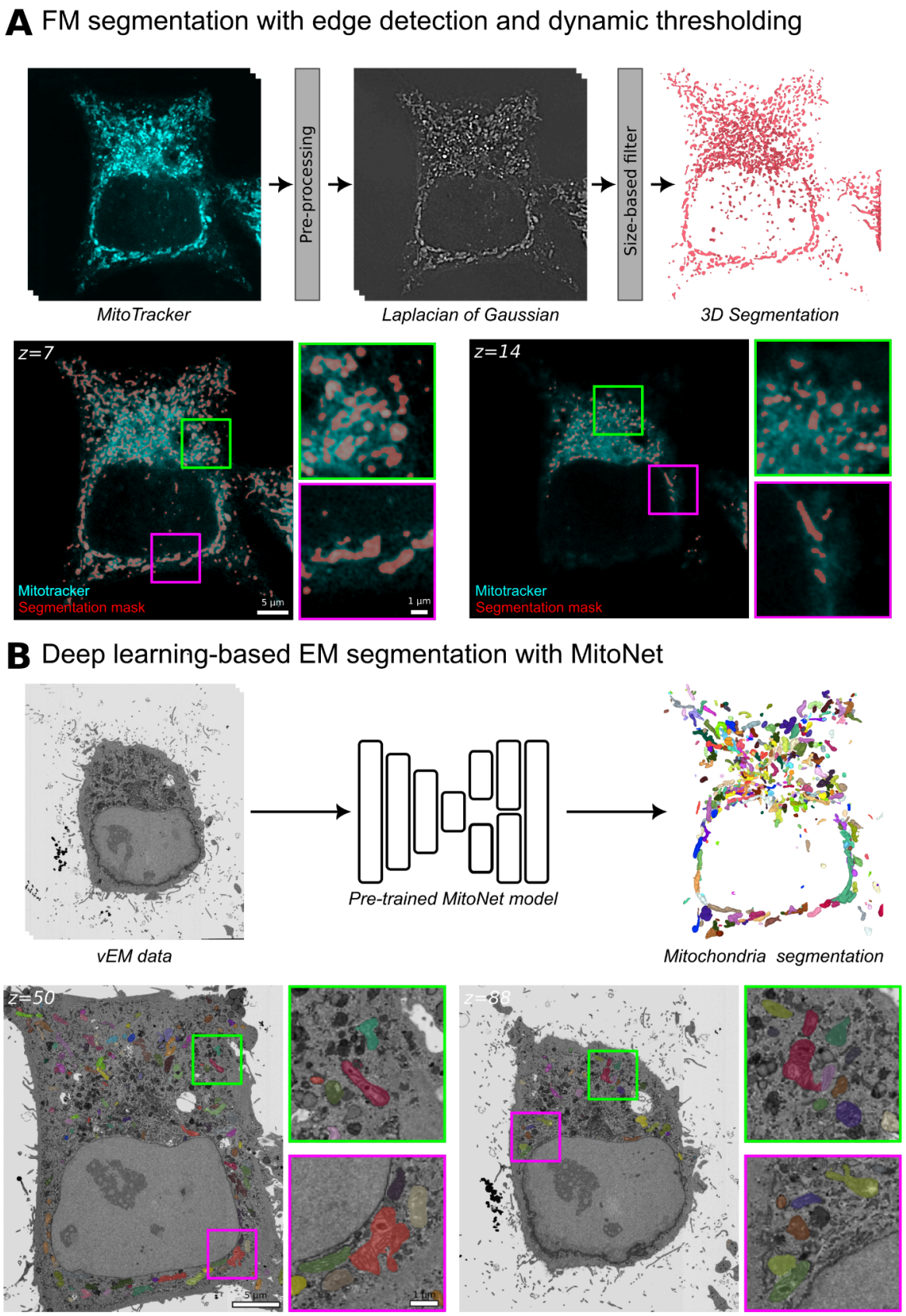

Figure 2. Mitochondria segmentation in LM and EM. (A) Mitochondria in the Mitotracker channel are segmented by applying a 3D Laplacian of Gaussian (LoG) filter to extract edges and dynamically thresholded to account for decreasing pixel intensity values as the imaging depth increases. To remove spurious mitochondria segmentations, a size-based filter is used. (B) Mitochondria in the vEM data are segmented with a pre-trained MitoNet (25) model. All visualisations generated with napari.

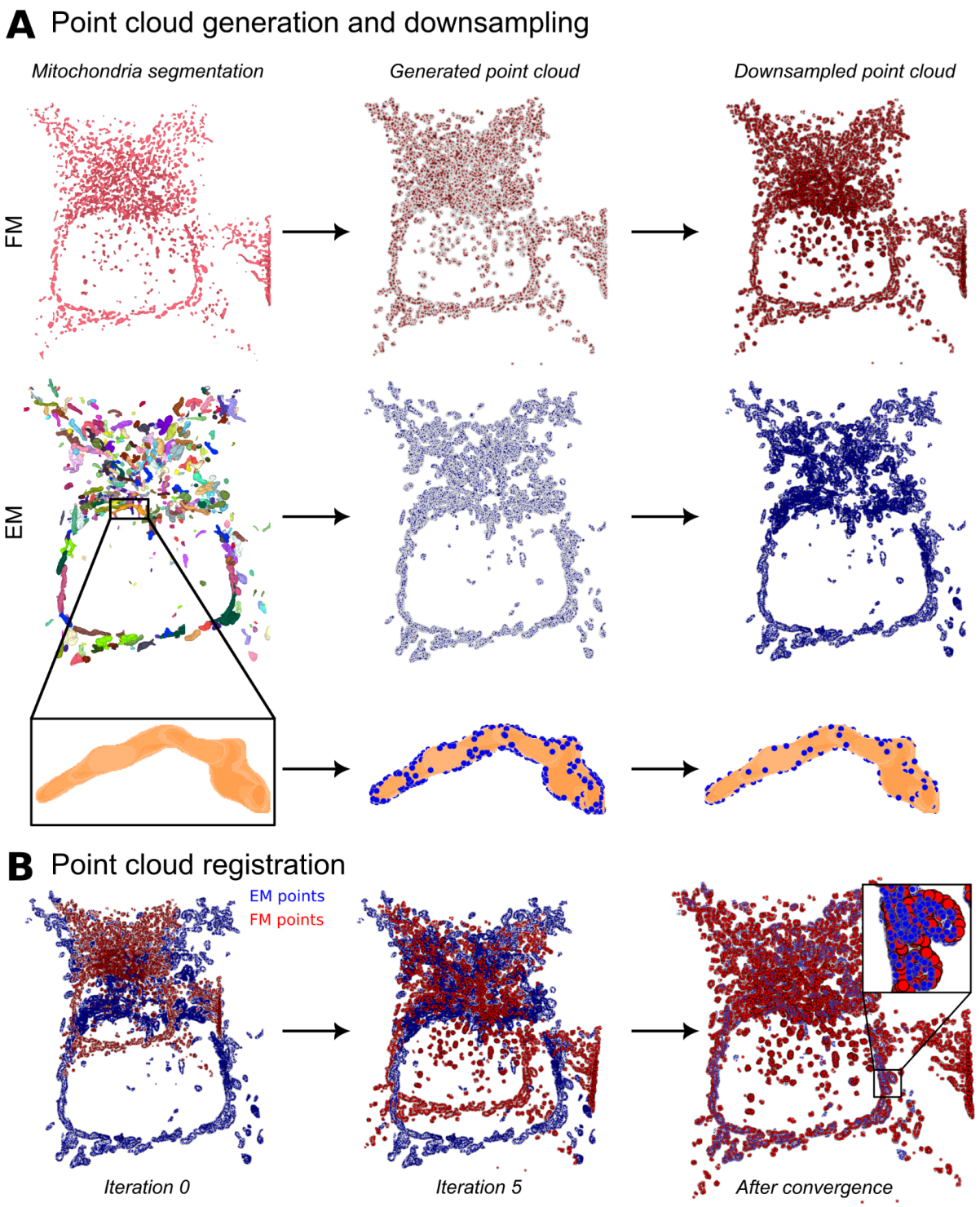

Figure 3. Point cloud generation and registration. (A) Point clouds are sampled on the surface of the 3D mitochondria segmentations in both LM and EM. To reduce the computational load and speed up the alignment time, both point clouds are downsampled. (B) The point clouds are registered using rigid CPD (28) until convergence (50 iterations). All visualisations generated with napari.

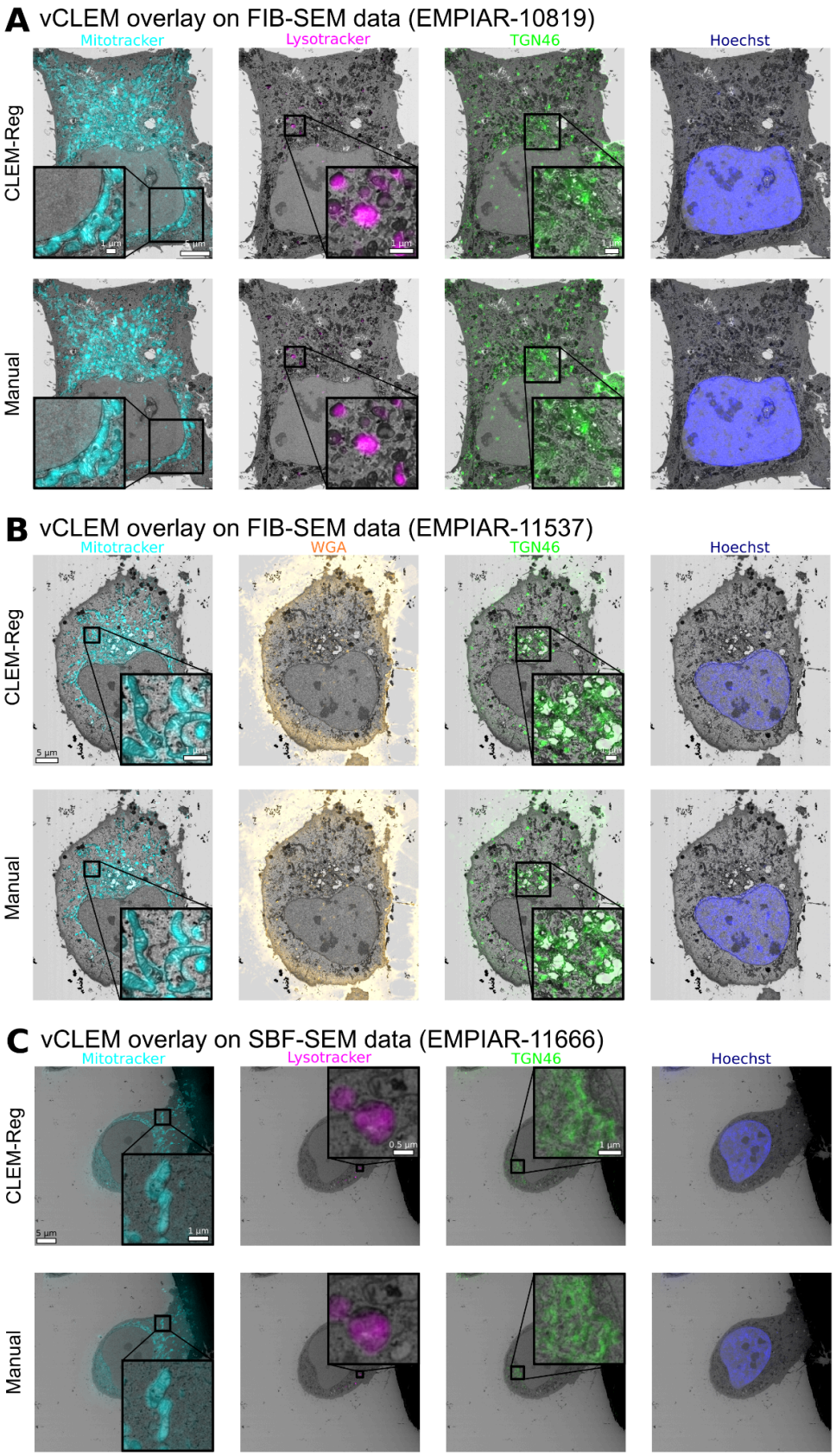

Figure 4. Comparing overlays obtained manually and with CLEM-Reg. CLEM-Reg overlays were obtained with rigid registration using the napari-clemreg plugin, while the manual overlays were obtained with affine registration using the BigWarp Fiji plugin. (A) CLEM overlays for EMPIAR-10819 dataset showing mitochondria (Mitotracker), lysosomes (Lysotracker), golgi apparatus (TGN46) and nucleus (Hoechst) with EM data acquired on FIB-SEM microscope. (B) CLEM overlays for EMPIAR-11537 dataset showing Mitotracker, WGA, TGN46 and Hoechst with EM data acquired on FIB-SEM microscope. (C) CLEM overlays for EMPIAR-11666 dataset showing Mitotracker, Lysotracker, TGN46 and Hoechst staining with EM data acquired on SBF-SEM microscope. Overlays generated with napari.

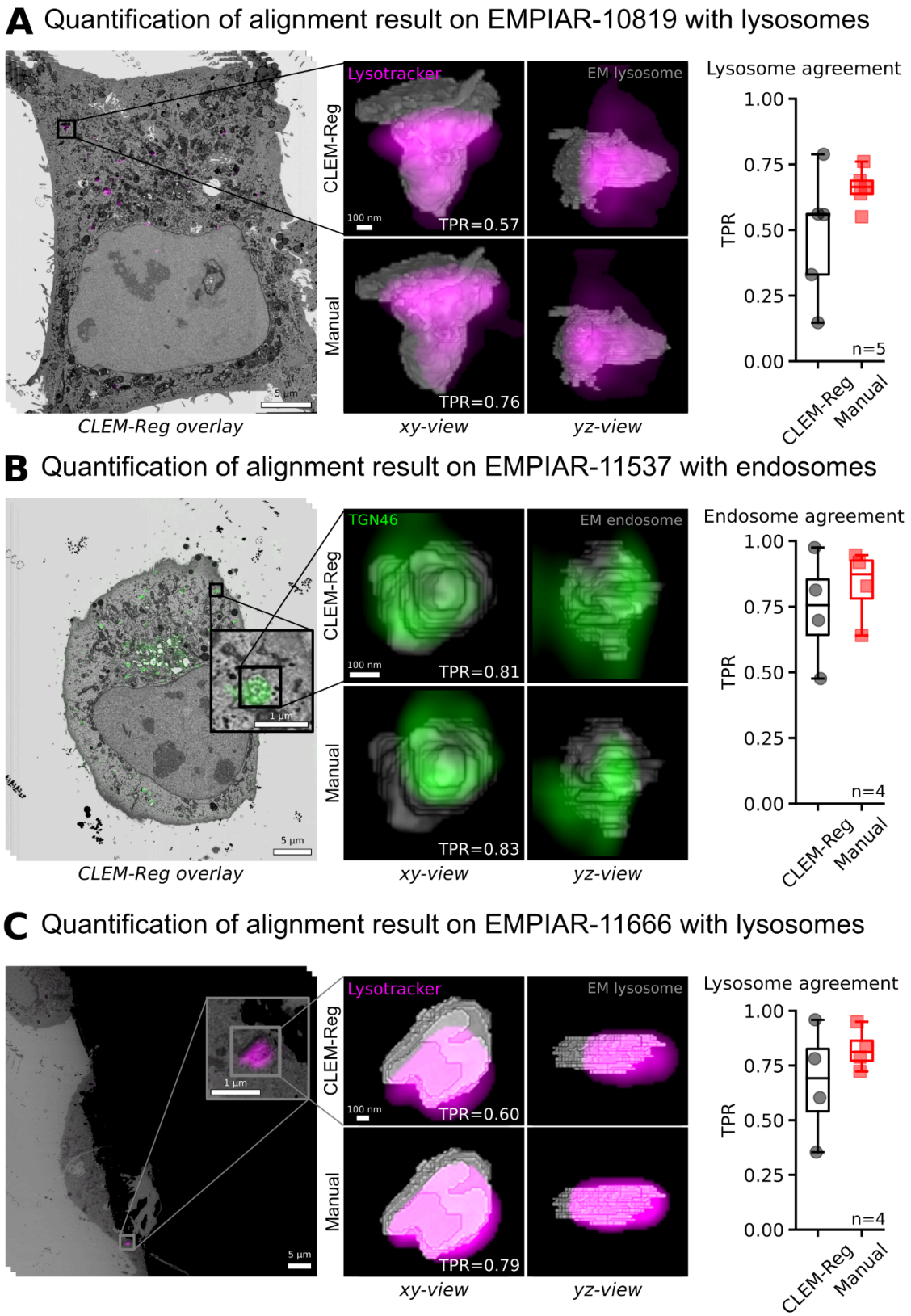

Figure 5. Comparing alignment results between CLEM-Reg and experts. (A) To quantify the vCLEM alignment results for dataset EMPIAR-10819, five lysosomes throughout the EM volume were manually segmented. The corresponding signal in FM was segmented within bounding boxes centred on the five lysosomes segmented in EM with Otsu's method (57). To assess the agreement between the FM signal and the manually segmented lysosomes in EM, the intersection between the FM and EM segmentations was computed and divided by the number of pixels in the EM segmentation. This measure is also known as the true positive rate (TPR). (B) For dataset EMPIAR-11537, four endosomes were manually segmented and the agreement of the FM signal with the EM volume was quantified as described above. C) For dataset EMPIAR-11666, four lysosomes were manually segmented and the agreement of the FM signal with the EM volume was quantified as described above. All image visualisations generated with napari.

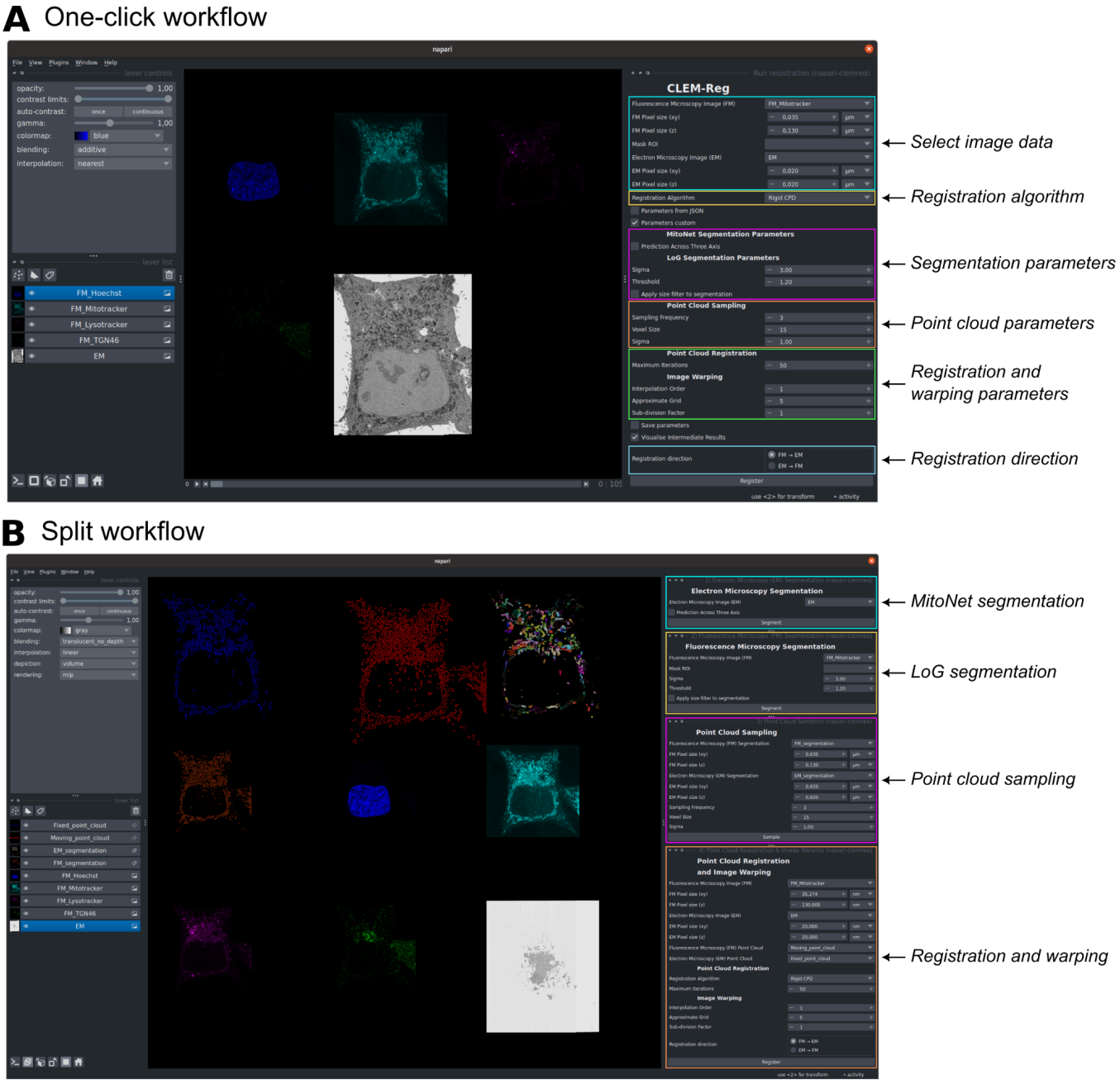

Figure 6. Interface of napari-clemreg plugin. (A) CLEM-Reg can be accessed via a napari plugin called napari-clemreg. The plugin automatically registers vCLEM datasets with a single button click. A ROI can also be drawn on the LM image using the 'Shapes' layer in napari to increase registration performance. The plugin also allows users the flexibility to choose between rigid and non-linear registration and tune parameters such as the LM segmentation settings, point cloud sampling density and the maximum number of iterations for the point cloud registration. The chosen parameters can be saved and later re-used to ensure reproducibility. Overlays can directly be exported from napari, as well as the initial and transformed point clouds. (B) The plugin can also be used as a split workflow thus enabling users to use segmentation masks or point clouds generated with a different algorithm. It can also be used to debug individual steps of the algorithm and to optimise parameters.