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]MRC-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 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 machine-learning. CLEM-Reg is a fully automated and reproducible vCLEM alignment workflow that requires no prior expert knowledge. When benchmarked against experts on two newly acquired vCLEM datasets, CLEM-Reg achieves near expert-level registration performance. The datasets are made available in the EMPIAR public image archive 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 similar visual appearance, for example by directly converting across modalities (8,9,19) or by constructing a shared modality-agnostic representation (10,28). Once the processed image stacks are sufficiently similar in visual appearance, traditional intensity-based registration techniques, such as those employed in medical imaging (12), 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 (13) and ec-CLEM (14). These tools rely on 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 the 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 (15,16). 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 (14). 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 (28) rely on the availability of large amounts of aligned ground truth data. Due to the low throughput and required expertise of manual 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 objective automation of landmark detection is highly desirable.

To achieve this aim, we have developed a workflow to segment the mitochondria, chosen since they are abundant and typically well-distributed across cells, in both the FM and EM images, in order to find matching structures across the modalities. Various segmentation approaches are widely used in microscopy ranging from classical image processing techniques (9,20,21) to machine learning (22–24). 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 (20), while automatically segmenting EM data typically requires advanced deep learning techniques. Common deep learning architectures such as the U-Net (25) 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 (24). Recently, however, pretrained “generalist” models such as MitoNet (27), based upon a "Panoptic-DeepLab" architecture (29), 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. 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. For this reason, we use an alternative point-based registration method, coherent point drift (CPD) (17), which removes the constraint of strict pairing, instead casting the alignment task as a probability density estimation problem. 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, two benchmark CLEM datasets (EMPIAR-10819 and EMPIAR-11537) 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 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 with an isotropic voxel-size of 5 nm were acquired using a focused ion beam scanning electron microscope (FIB-SEM). The acquired images were preprocessed following standard EM protocols (**Fig. 1a**, **Methods**).

The CLEM-Reg pipeline. CLEM-Reg automatically registers volume CLEM 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 button click. 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, affine 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. 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 for quality control purposes. Intermediate outputs like segmentation masks and sampled point clouds can directly be outputted to the napari viewer to aid troubleshooting (**Fig. 6**). In this study, the performance and user-friendliness of the CLEM-Reg plugin were assessed by an expert electron microscopist from the Electron Microscopy Science Technology Platform at The Francis Crick Institute on two vCLEM datasets (EMPIAR-10819 and EMPIAR-11537).

Segmenting internal landmarks. A promising approach to fully automate the CLEM alignment process is to seek to identify internal landmarks automatically, 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 landmark labels. 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 pretrained 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 approximated by sampling 3D point clouds from the previously obtained segmentation masks. This reduces computational load for large datasets and allows for minor missegmentations to be ignored by using a probabilistic registration algorithm, such as CPD. The number of points in both point clouds can be modified by adjusting a user-defined binning and downsampling factor. Increasing these two parameters drastically speeds up the registration and reduces the required random-access memory (RAM) (**Fig. 3a**).

After sampling, the point clouds are registered using either rigid CPD, affine CPD or non-linear BCPD (**Fig. 3b**). Note that these probabilistic methods are necessary, as opposed to ICP, since the stochastic sampling means there are no explicitly paired points across the modalities. 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 affine CPD, while non-linear BCPD is slower and more expensive. In case of large global displacements and non-linear deformations on a local level between the two imaging volumes, it can be advantageous to first run the point cloud registration with rigid CPD and to then fine-tune the alignment with non-linear BCPD (**Supp. Fig. 1**). This 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 and affine transformations, but slower for non-linear warping. CLEM-Reg implements 3D non-linear thin plate spline warping (32) that uses the initially sampled and registered FM point clouds as control points. Due to the high number of points in the point clouds, the resulting overlay leads to an accurate local alignment (**Supp. Fig 2**). 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 of chunking the data 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 RAM. To obtain overlays from the two benchmark datasets, it was found that rigid alignment was sufficient (**Fig. 4**).

Assessing CLEM-Reg performance against experts. The FM volumes were manually registered to the EM acquisitions by an expert microscopist with a global affine registration using BigWarp. The resulting alignments were then compared to a manual alignment obtained from an expert electron microscopist (**Fig. 5**). CLEM-Reg uses landmarks (mitochondria) identified in both modalities for registration. To accurately assess the registration performance across the whole cell, target structures (lysosomes in EMPIAR-10819 and endosomes in EMPIAR-11537) were manually segmented in EM in various regions of the cell. To quantify the overlay, the fluorescent signal was segmented by thresholding and the agreement compared to the manual segmentations (see Materials and Methods). Usually, metrics such as the Intersection over Union (IoU) or dice coefficient are used to assess the agreement between segmentations by dividing the intersection over the union of both segmentations. However, since the FM signal is non-isotropic in z, for a perfectly aligned FM signal the above-mentioned metrics would underestimate the agreement, as the FM segmentation would be larger than the EM segmentation. The agreement between the fluorescent signal and the manually segmented target structures was thus assessed by computing the true positive rate (TPR) of fluorescent pixels overlaid to the manual segmentations (with respect to the EM segmentation). For a perfectly aligned FM signal, this metric does yield a value of TPR = 1, since the intersection is only divided by the number of pixels in the smaller EM segmentation. CLEM-Reg is able to correctly match the fluorescent signal to all the target structures in EM ($TPR_{lysosomes}$ = 0.48±0.22 and $TPR_{endosomes}$ = 0.74±0.18). However, the fraction of fluorescent signal correctly overlaid to the corresponding manual segmentation is higher for the manually aligned datasets ($TPR_{lysosomes}$ =

0.66±0.12 and $TPR_{endosomes}$ = 0.83±0.12).

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 direct modification of the image viewer, as well as the ability to read and write files (33). The napari-clemreg plugin was developed for this platform due to the increasing adoption of Python in the bioimage analysis community and the seamless integration of state-of-the-art image processing packages, as well as deep learning libraries into napari. Furthermore, napari's support for the visualisation of point clouds and segmentation in the form of the "Points" and "Labels" layers respectively made it an ideal choice for illustrating many steps of 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. 6**). 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. 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 a combination of linear and non-linear state-of-the-art point cloud registration methods, a mapping between the two imaging volumes is found with which the FM acquisition is warped onto the EM volume to obtain a final CLEM overlay. CLEM-Reg achieves expert-level performance on two 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 two 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 (34), DeepImageJ (35) and ImJoy (36) are rapidly removing these barriers by offering frameworks that leverage freely available computing resources for training and enable GUI-driven interaction with a range of deep learning architectures. Open source and community-driven model zoos such as Bioimage.IO (`https://bioimage.io/#/`) which allow easy sharing of pre-trained deep learning models are another important resource in this context.

Further limitations prevail with regard to the scale of the datasets that can be aligned. 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, chunking of the datasets using next generation file formats (37) and tools such as Dask (38) could enable computation of datasets that exceeds available memory, thus allowing users to run the CLEM-Reg workflow without encountering memory issues. Additionally, the CLEM-Reg workflow relies heavily on two packages Scipy (39) and scikit-image (40), 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 (41) 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 expert-level registration, it is important to appreciate that the acquired benchmark CLEM dataset was obtained with high-end microscopes by professional electron microscopists. Other vEM techniques like serial section scanning EM (ssSEM) or serial section transmission EM (ssTEM) introduce more pronounced nonlinear deformations, as the sample is physically cut with a diamond knife and in the case of ssTEM, imaged in ultrathin sections. In FIB-SEM, which was used to acquire the benchmark EM data, the sample is milled by a focused ion beam and imaged in a resin-block which provides stability (43). Thus, further studies are necessary to ascertain the robustness of the proposed algorithm across a range of CLEM datasets acquired under varying conditions in different microscopes. CLEM-Reg is a first step towards potentially making fully automated high-throughput CLEM registration a reality. With recent advances in deep learning enabling the use of ambiguous GT data (44), automatically obtained CLEM overlays could replace the need for manually labelled structures removing the need for human annotators. This in turn would significantly accelerate the speed at which new deep learning models could be trained for segmenting a range of structures in vEM.

While CLEM-Reg was developed to register vCLEM datasets, the framework can be extended to other multi-modal registration tasks which are not amenable to intensity-based registration algorithms via the split registration functionality. This is due to the fact that aligning common landmarks represented as point clouds is a modality agnostic approach, provided that segmentations for both modalities are available. With the rapid development of novel generalist deep learning models (45), it is anticipated that this step will become progressively easier.

Materials and Methods

Cell model. Human cervical cancer epithelial cells (HeLa) were obtained from Cell Services 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). 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 (46). Each step was performed in the Biowave, except for the PB 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, Stansted, UK) with 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 embedding 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 (Zeiss, Cambridge). A segment of the cell monolayer containing the cell of interest was trimmed out, and the coverslip 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 current of 700 pA.

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 highest quality, translation and linear interpolation, with the mitotracker channel as reference. The .czi file was then opened in Fiji (47) and saved as .tif. 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 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 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.

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. Firstly, the FM volume was resampled to obtain 20 nm isotropic voxels. Then, 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. *D. Marr and E. Hildreth* (49) 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 (50) which removes components below a user-defined size threshold with a default value set to remove all segmentations with a size below the 50th percentile.

EM segmentation with MitoNet. The mitochondria within the EM volume were segmented using the pre-trained deep learning model, MitoNet (31). The MitoNet model using its default parameters was applied to the EM volume, resulting in instance segmentation of the mitochondria.

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, a Canny edge filter (51) with a default value of $\sigma = 1.0$ 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 required. The frequency with which the final points were sampled from the found membrane segmentation is based on a user-defined sampling frequency parameter. For a given point in the list of all points, if the random number was below the user-defined sampling frequency then that point was added to the final point cloud, resulting in a list of points representing a point cloud of the EM and FM volumes. A preset sampling frequency of $f = 0.05$ was used.

Point cloud registration with CPD and BCPD. After sampling the point clouds, a downsampling and binning step was applied to both the EM and FM point clouds to reduce memory requirements. Downsampling of the point clouds was performed by a uniform downsampling function which takes point cloud and user defined variable k as the input and would sample every k^{th} point in 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 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 (52) was used, as it implements both the CPD (17) and BCPD (53) 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 (39), 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 (54) 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 used based on (32). To reduce computational overhead, the FM volume was nonlinearly warped as eight individual chunks.

Benchmarking of CLEM-Reg against experts. 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 endosomes in EMPIAR-11537) were manually segmented in 3D in the EM volume. These lysosomes and endosomes were then cropped with a bounding box from the corresponding warped FM volumes and then segmented using the Otsu thresholding method (55) to generate the correlating FM segmentation. The intersection between the FM and EM segmentations is 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 (33). 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.

DATA AND CODE AVAILABILITY

The datasets used in this study have been deposited at EMPIAR and Bioimage Archive. 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).
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 (CC001999), the UK Medical Research Council (CC001999), and the Wellcome Trust (CC001999). 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 Molecular Biology Organization (EMBO) Installation Grant (EMBO-2020-IG4734) and the Chan Zuckerberg Initiative Visual Proteomics Grant (vpi-0000000044). 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

Daniel Krentzel, Romain F. Laine, Ricardo Henriques, Lucy M. Collinson and Martin L. Jones developed the initial idea for this project. Daniel Krentzel conceived CLEM-Reg and directed this project. Daniel Krentzel, Matouš Elphick, Martin L. Jones and Lucy M. Collinson wrote the manuscript. Daniel Krentzel and Matouš Elphick developed the source-code. Romain F. Laine collected preliminary CLEM data and Marie-Charlotte Domart (Electron Microscopy Science Technology Platform, The Francis Crick Institute) and Christopher J. Peddie (Electron Microscopy Science Technology Platform, The Francis Crick Institute) collected the benchmark CLEM datasets. 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. 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
11. Ng, H. P., Ong, S. H., Foong, K. W. C., Goh, P. S. & Nowinski, W. L. Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm. in *2006 IEEE Southwest Symp. Image Anal. Interpret.* 61–65 (2006). doi:10.1109/SSIAI.2006.1633722
12. 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).
13. Pietzsch, T., Saalfeld, S., Preibisch, S. & Tomancak, P. BigDataViewer: visualization and processing for large image data sets. *Nat. Methods* **12,** 481–483 (2015).
14. Paul-Gilloteaux, P. *et al.* eC-CLEM: flexible multidimensional registration software for correlative microscopies. *Nat. Methods* **14,** 102–103 (2017).
15. 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).
16. 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).
17. Myronenko, A. & Song, X. Point-Set Registration: Coherent Point Drift. *IEEE Trans.*

Pattern Anal. Mach. Intell. **32,** 2262–2275 (2010).

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. Heinrich, L. *et al.* Whole-cell organelle segmentation in volume electron microscopy. *Nature* **599,** 141–146 (2021).

20. Malpica, N. *et al.* Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry* **28,** 289–297 (1997).

21. Berg, S. *et al.* ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods* **16,** 1226–1232 (2019).

22. Schmidt, U., Weigert, M., Broaddus, C. & Myers, G. Cell Detection with Star-convex Polygons. *ArXiv180603535 Cs* **11071,** 265–273 (2018).

23. 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

24. 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).

25. 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

26. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods* **18,** 100–106 (2021).

27. 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

28. Pielawski, N. *et al.* CoMIR: Contrastive Multimodal Image Representation for Registration.

29. 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)

30. Ponnambalam, S. *et al.* Primate homologues of rat TGN38: primary structure, expression and functional implications.

31. Conrad, R. & Narayan, K. *Empanada MitoNet model files*. (Zenodo, 2022). doi:10.5281/zenodo.6861565

32. Bookstein, F. L. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* **11,** 567–585 (1989).

33. Nicholas Sofroniew *et al. napari/napari: 0.4.2rc0*. (Zenodo, 2020). doi:10.5281/zenodo.4294808

34. von Chamier, L. *et al.* Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat. Commun.* **12,** 2276 (2021).

35. Gómez-de-Mariscal, E. *et al.* DeepImageJ: A user-friendly environment to run deep learning models in ImageJ. *Nat. Methods* **18,** 1192–1195 (2021).

36. 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).

37. Moore, J. *et al.* OME-NGFF: a next-generation file format for expanding bioimaging data-access strategies. *Nat. Methods* **18,** 1496–1498 (2021).

38. Rocklin, M. Dask: Parallel Computation with Blocked algorithms and Task Scheduling. in 126–132 (2015). doi:10.25080/Majora-7b98e3ed-013

39. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17,** 261–272 (2020).

40. Walt, S. van der *et al.* scikit-image: image processing in Python. *PeerJ* **2,** e453 (2014).

41. Okuta, R., Unno, Y., Nishino, D., Hido, S. & Loomis, C. CuPy: A NumPy-Compatible

Library for NVIDIA GPU Calculations.

42. Haase, R. *et al.* Interactive design of GPU-accelerated Image Data Flow Graphs and cross-platform deployment using multi-lingual code generation.

43. Collinson, L. M., Carroll, E. C. & Hoogenboom, J. P. Correlating 3D light to 3D electron microscopy for systems biology. *Curr. Opin. Biomed. Eng.* **3,** 49–55 (2017).

44. Kohl, S. *et al.* A Probabilistic U-Net for Segmentation of Ambiguous Images. in *Adv. Neural Inf. Process. Syst.* **31,** (Curran Associates, Inc., 2018).

45. Kirillov, A. *et al.* Segment Anything. at http://arxiv.org/abs/2304.02643 (2023)

46. 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>

47. Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9,** 676–682 (2012).

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. Marr, D. & Hildreth, E. Theory of edge detection. *Proc. R. Soc. Lond. B Biol. Sci.* **207,** 187–217 (1980).

50. Silversmith, W. *cc3d: Connected components on multilabel 3D & 2D images.* (Zenodo, 2021). doi:10.5281/zenodo.5719536

51. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8,** 679–698 (1986).

52. Kenta-Tanaka et al. *probreg*. (2019). at <https://probreg.readthedocs.io/en/latest/>

53. Hirose, O. A Bayesian Formulation of Coherent Point Drift. *IEEE Trans. Pattern Anal. Mach. Intell.* 1–1 (2020). doi:10.1109/TPAMI.2020.2971687

54. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585,** 357–362 (2020).

55. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **9,** 62–66 (1979).

Figures

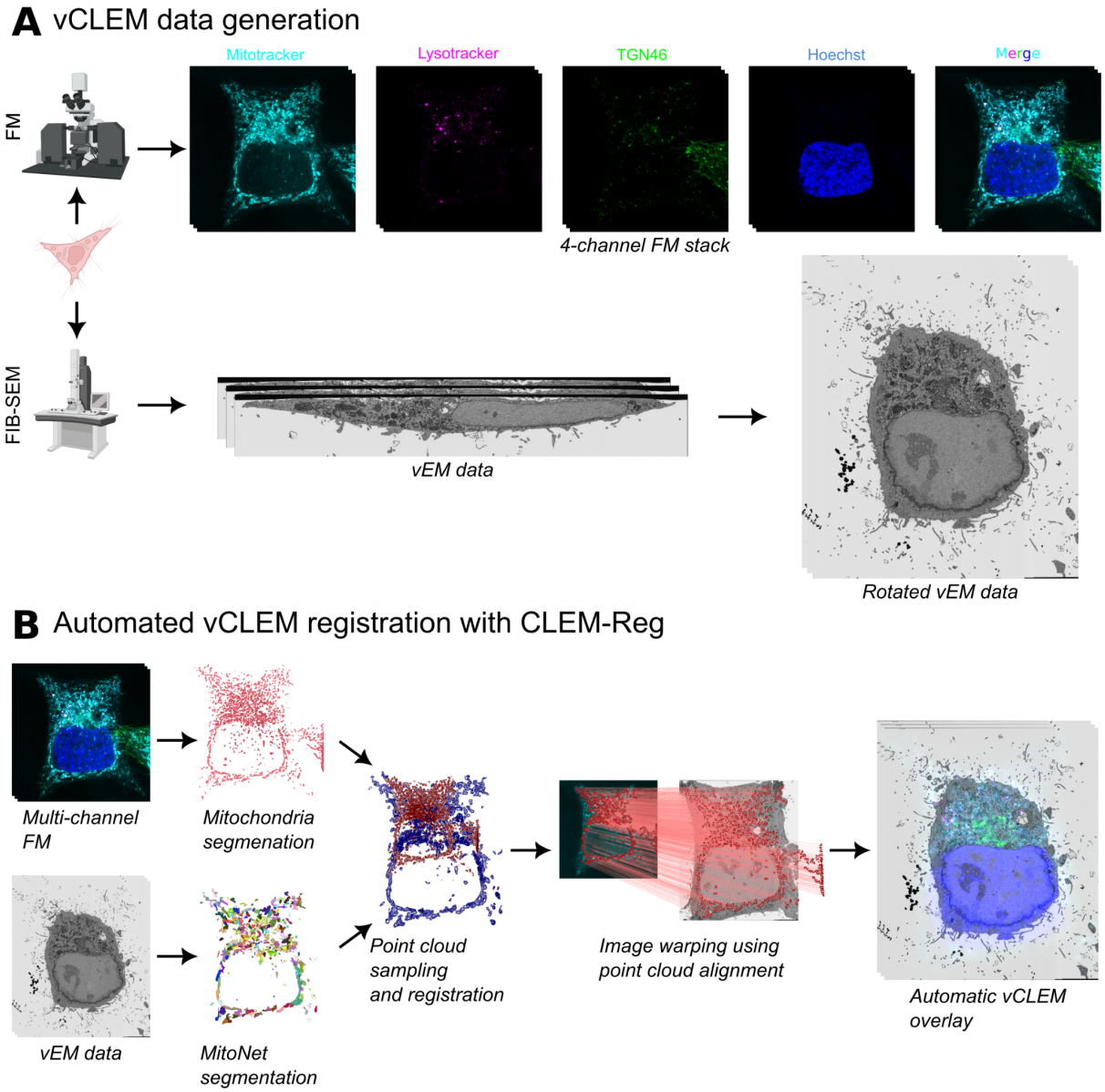

Figure 1. Volume CLEM (vCLEM) data generation and CLEM-Reg algorithm. (A) Obtaining a vCLEM dataset consists of acquiring a FM and vEM image of the same sample. The two image stacks are manually aligned by identifying landmarks in both modalities and computing a transform to warp the LM stack onto the 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 LM stack onto the vEM data. All data visualisations generated with napari.

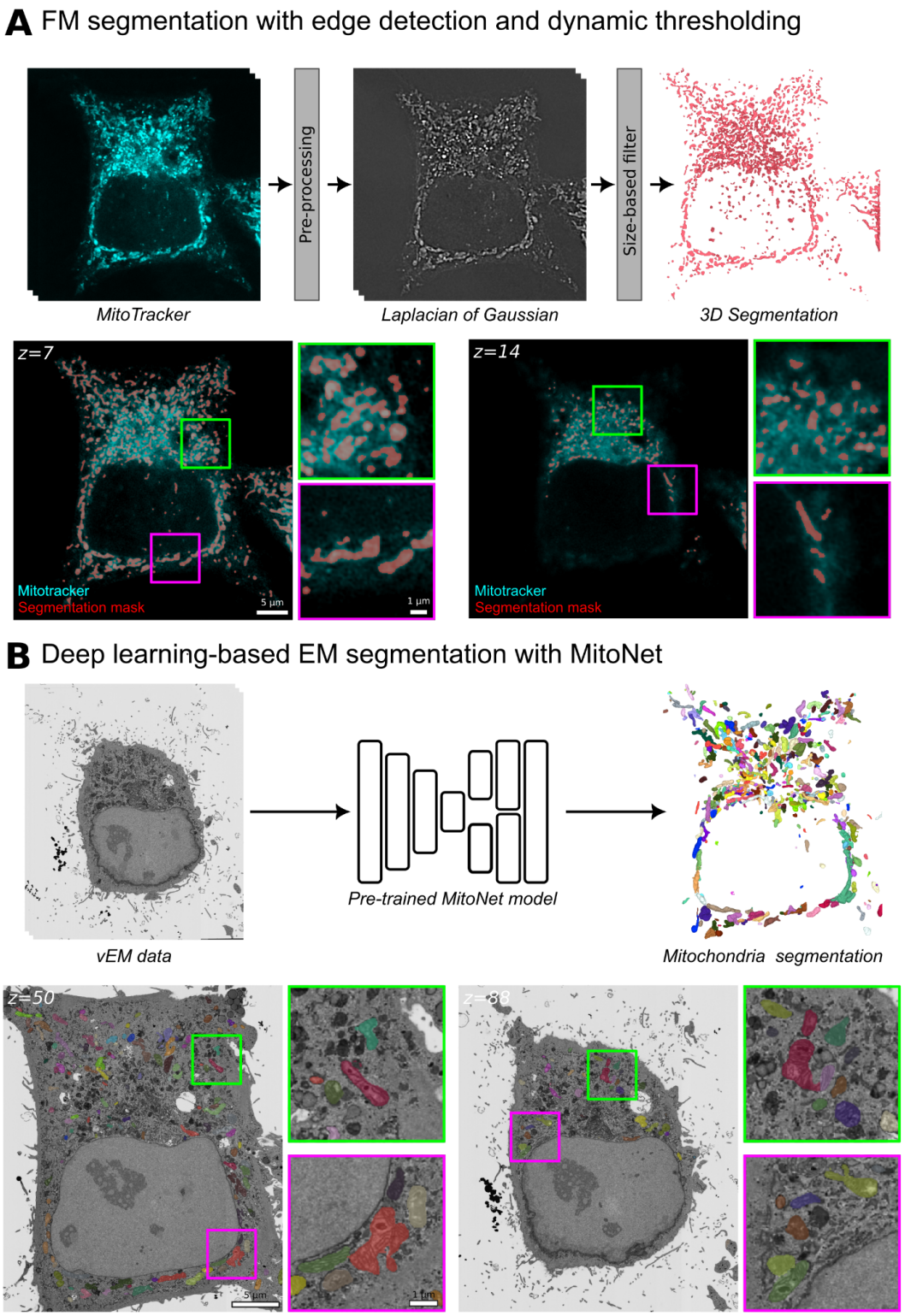

Figure 2. Mitochondria segmentation in LM and EM. (A) Mitochondria in the Mitotracker channel are segmented by applying a 3D Laplacian of Gaussian (LoG) filter to extract edges and dynamically thresholded to account for decreasing pixel intensity values as the imaging depth increases. To remove spurious mitochondria segmentations, a size-based filter is used. (B) Mitochondria in the vEM data are segmented with a pre-trained MitoNet (27) model. All visualisations generated with napari.

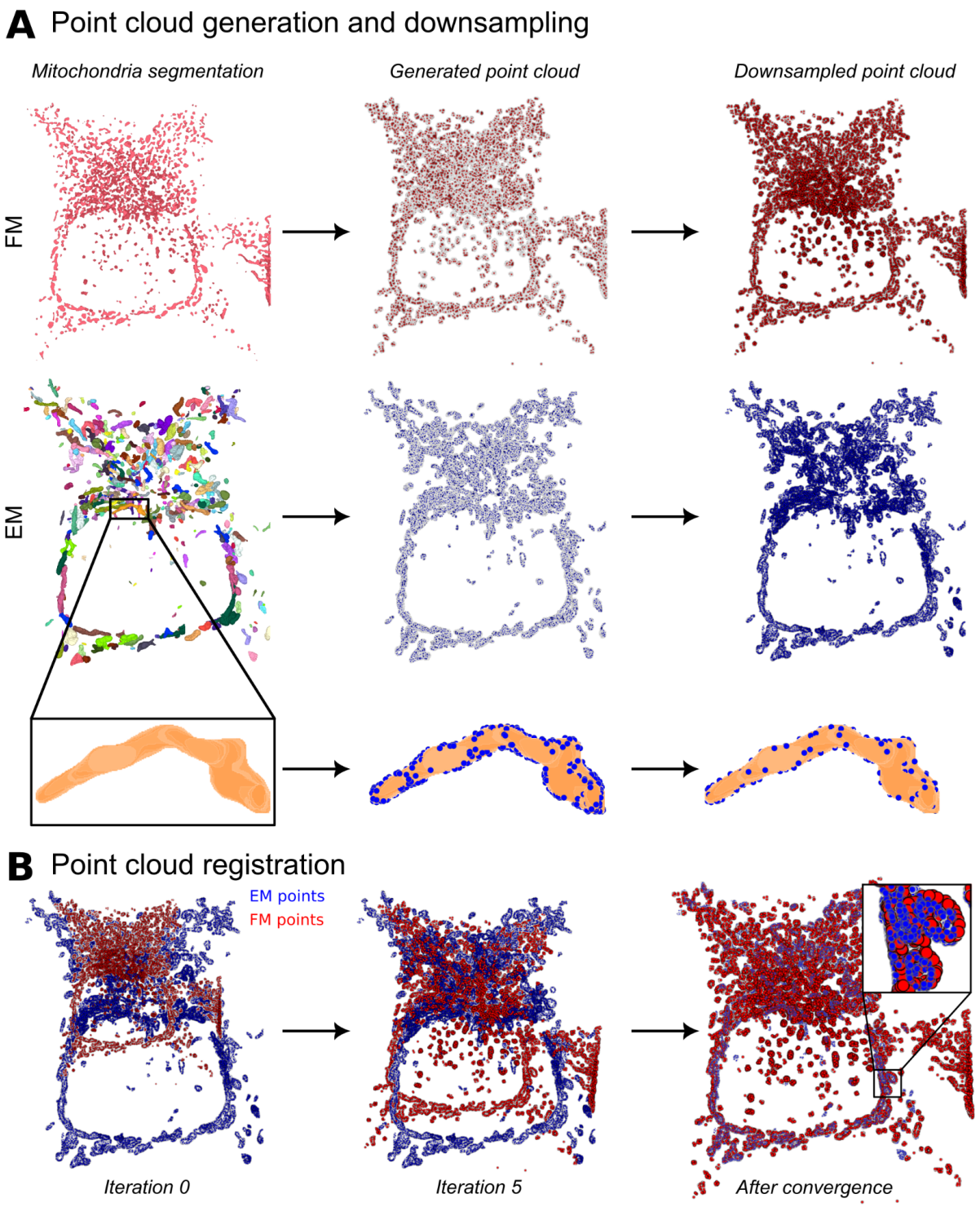

Figure 3. Point cloud generation and registration. (A) Point clouds are sampled on the surface of the 3D mitochondria segmentations in both LM and EM. To reduce the computational load and speed up the alignment time, both point clouds are downsampled. (B) The point clouds are registered using rigid CPD (17) for 50 iterations. All visualisations generated with napari.

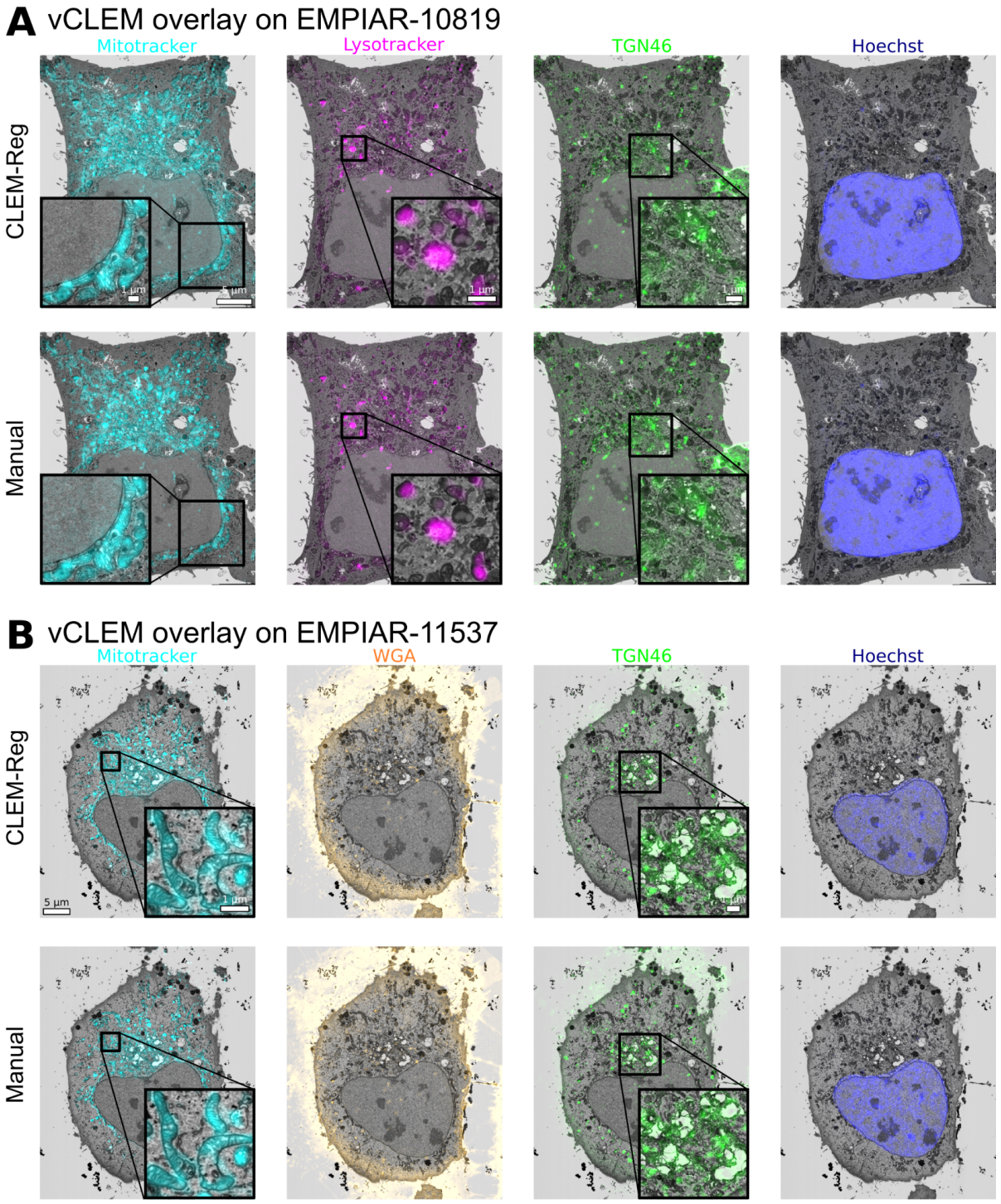

Figure 4. Comparing overlays obtained manually and with CLEM-Reg. CLEM-Reg overlays were obtained with rigid registration, while the manual overlays were obtained with affine registration. Scale bars on the whole EM image are 5 µm and scale bars in insets are 1 µm. (A) CLEM overlays for EMPIAR-10819 dataset showing mitochondria (Mitotracker), lysosomes (Lysotracker), golgi apparatus (TGN46) and nucleus (Hoechst). (B) CLEM overlays for EMPIAR-11537 dataset showing mitochondria (Mitotracker), cell membrane (WGA), golgi apparatus (TGN46) and nucleus (Hoechst). Overlays generated with napari.

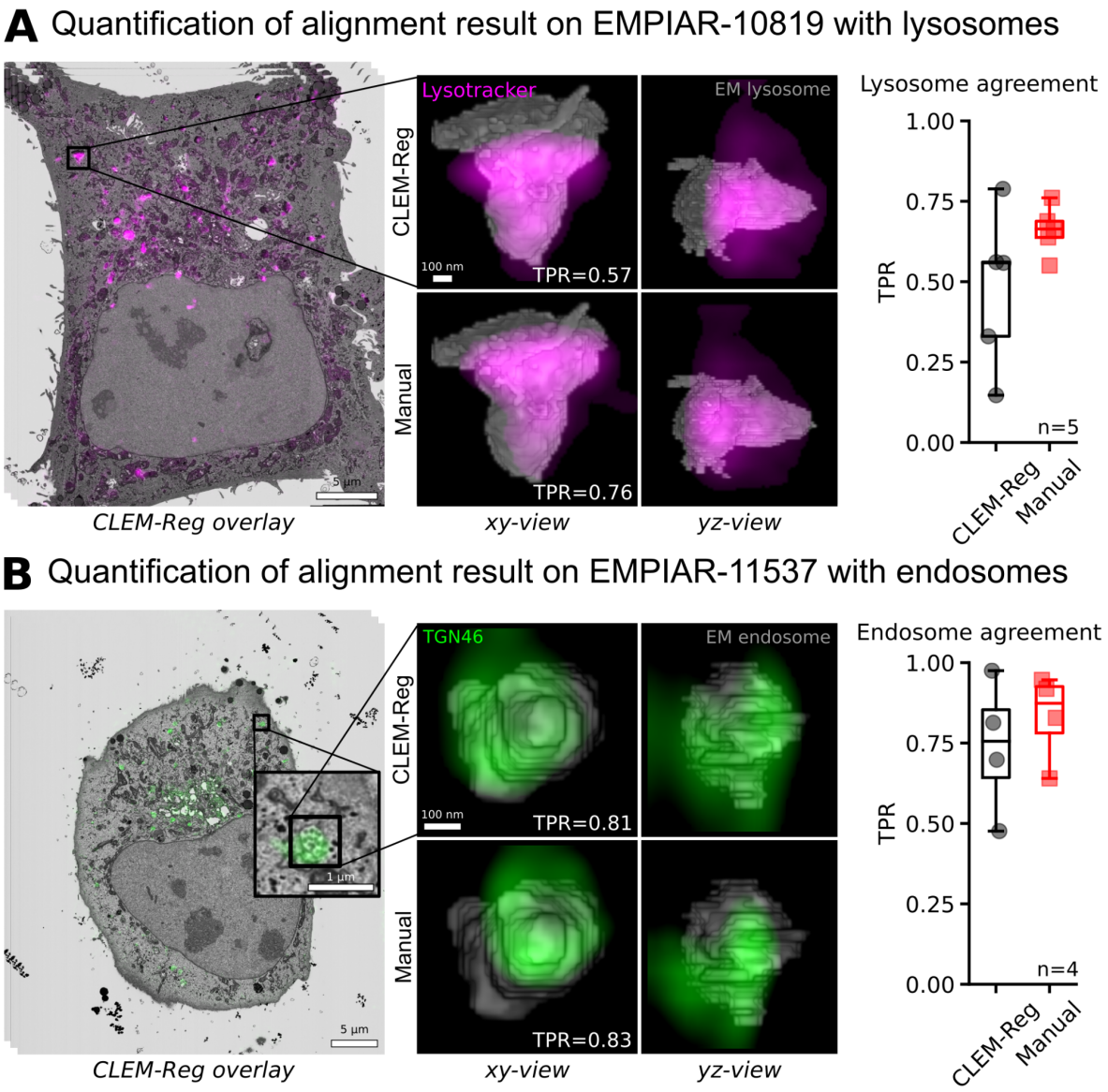

Figure 5. Comparing alignment results between CLEM-Reg and experts. (A) To quantify the vCLEM alignment results for dataset EMPIAR-10819, five lysosomes throughout the EM volume were manually segmented. The corresponding signal in FM was segmented within bounding boxes centred on the five lysosomes segmented in EM with Otsu's method (55). 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. All image visualisations generated with napari.

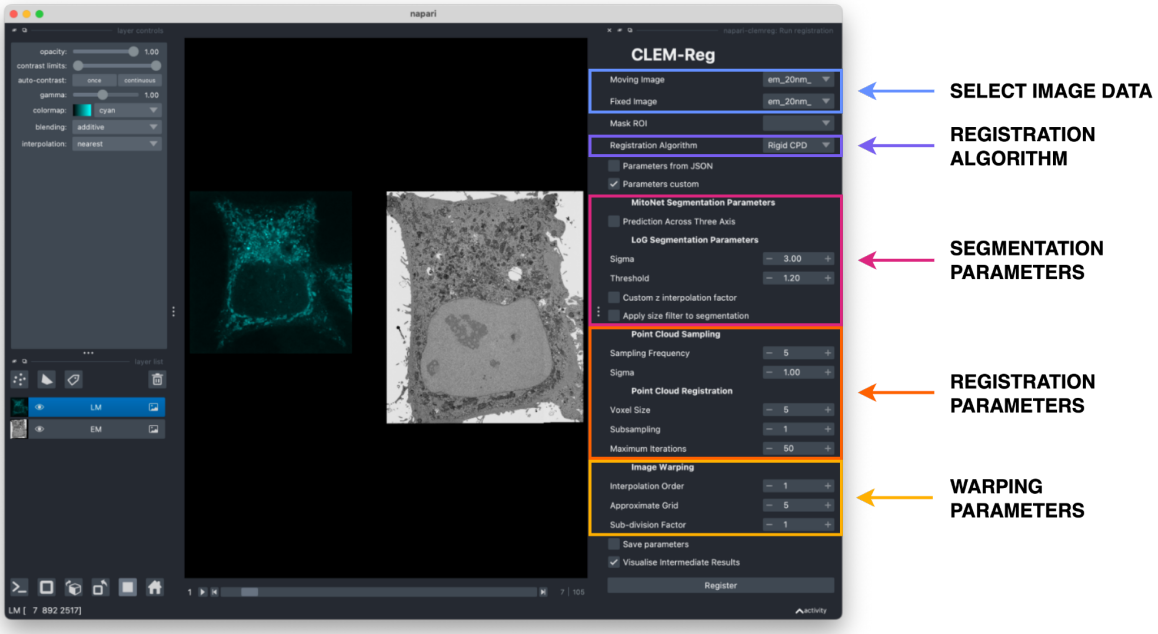

Figure 6. Interface of napari-clemreg plugin. CLEM-Reg can be accessed via a napari plugin called napari-clemreg. The plugin automatically registers vCLEM datasets with a single button click. A ROI can also be drawn on the LM image using the 'Shapes' layer in napari to increase registration performance. The plugin also allows users the flexibility to choose between rigid, affine 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.