Contents lists available at ScienceDirect

Journal of Structural Biology

journal homepage: www.elsevier.com/locate/yjsbi

Research Article

ColabSeg: An interactive tool for editing, processing, and visualizing membrane segmentations from cryo-ET data

Marc Siggel [a,b], Rasmus K. Jensen [c], Valentin J. Maurer [a,b], Julia Mahamid [c,d], Jan Kosinski [a,b,c,*]

[a] European Molecular Biology Laboratory (EMBL) Hamburg, Notkestrasse 85, Hamburg 20607, Germany
[b] Centre of Structural Systems Biology (CSSB), Notkestrasse 85, Hamburg 20607, Germany
[c] Structural and Computational Biology Unit, European Molecular Biology Laboratory (EMBL) Heidelberg, Meyerhofstrasse 1, Heidelberg 69117, Germany
[d] Cell Biology and Biophysics Unit, European Molecular Biology Laboratory (EMBL) Heidelberg, Meyerhofstrasse 1, Heidelberg 69117, Germany

ARTICLE INFO

Keywords:
Cryo-electron tomography (Cryo-ET)
Electron microscopy
Segmentation
Lipid membranes
Image analysis

ABSTRACT

Cellular cryo-electron tomography (cryo-ET) has emerged as a key method to unravel the spatial and structural complexity of cells in their near-native state at unprecedented molecular resolution. To enable quantitative analysis of the complex shapes and morphologies of lipid membranes, the noisy three-dimensional (3D) volumes must be segmented. Despite recent advances, this task often requires considerable user intervention to curate the resulting segmentations. Here, we present ColabSeg, a Python-based tool for processing, visualizing, editing, and fitting membrane segmentations from cryo-ET data for downstream analysis. ColabSeg makes many well-established algorithms for point-cloud processing easily available to the broad community of structural biologists for applications in cryo-ET through its graphical user interface (GUI). We demonstrate the usefulness of the tool with a range of use cases and biological examples. Finally, for a large *Mycoplasma pneumoniae* dataset of 50 tomograms, we show how ColabSeg enables high-throughput membrane segmentation, which can be used as valuable training data for fully automated convolutional neural network (CNN)-based segmentation.

1. Introduction

In situ cryo-ET has emerged as a powerful method to visualize and analyze the 3D structure of cells and sub-cellular architecture (Mahamid et al., 2016; Pfeffer et al., 2017; Wilfling et al., 2020; Lučić et al., 2013). The 3D volume of a (sub)cellular region, called a tomogram, is reconstructed from 2D projection images acquired on a transmission electron microscope in many different orientations (Pyle and Zanetti, 2021; Volkmann, 2010). Macromolecular complexes can be identified in the tomogram, their spatial arrangement can be analyzed in the native environment, and their structure can potentially be determined to near-atomic resolution (Lučić et al., 2013; Pfeffer et al., 2017; Wilfling et al., 2020; Mahamid et al., 2016; Tegunov et al., 2021). Cryo-ET has also proven particularly useful in tracing the abundant cellular membranes in 3D (Zabeo and Davies, 2022; Lamm et al., 2024; Wietrzynski et al., 2020). Membrane segmentations can be used to analyze a variety of membrane properties such as shape, curvature, or volume to understand morphological changes and biological function. These properties can be analyzed with existing software packages (Salfer et al., 2020; Barad et al., 2023). Membrane segmentations can be also exploited to extract associated membrane proteins (Lamm et al., 2022; Martinez-Sanchez et al., 2020).

However, obtaining these insights comes with numerous challenges. The cryo-ET reconstructions are often noisy or incomplete due to several factors such as (i) the crowded cellular environment, (ii) the missing wedge, (iii) low signal-to-noise ratio, or (iv) artifacts from, i.e., non-vitreous ice or edges of the carbon support film. Due to these complications, segmentation and subsequent analysis of cryo-ET data is still a difficult task and a major bottleneck for automated high-throughput analysis of large datasets (Lučić et al., 2013; Pyle and Zanetti, 2021; Wu et al., 2019). Current membrane segmentation methods are still far from being fully automated and due to the effort required, many tomograms remain unsegmented rendering them not useful for large-scale statistical analyses.

Various tools and algorithms exist to simplify or partially automate the segmentation of membranes from cryo-ET data. Traditional methods include template matching or a watershed algorithm (Lebbink et al., 2007, 2008; Tasel et al., 2016; Luengo et al., 2017). A prevalent tool for membrane segmentation, TomoSegMemTV, uses the tensor voting (TV) method (Martinez-Sanchez et al., 2014; Tong et al., 2004). In the

* Corresponding author.
E-mail address: jan.kosinski@embl.de (J. Kosinski).

https://doi.org/10.1016/j.jsb.2024.108067
Received 12 July 2023; Received in revised form 17 January 2024; Accepted 3 February 2024
Available online 15 February 2024

TomoSegMemTV pipeline, a tomogram is filtered, and ridges are enhanced and filtered based on their surfaceness properties. This pipeline and the algorithms are also available as a parallelized implementation for high-performance computing (HPC) systems (Moreno et al., 2022). While the overall performance of the tool is relatively high, it requires tuning many parameters of the pipeline based on the specifics of the tomogram such as the Gaussian filter widths, the size of the neighborhood considered for the tensor voting algorithm, as well as the intensity and surfaceness thresholds. In all cases, the output membrane segmentation often requires laborious manual cleaning with a tool like Amira (Stalling et al., 2005). More recently, two convolutional neural networks (CNNs) have been developed for particle (Moebel et al., 2021; de Teresa-Trueba et al., 2023) and membrane segmentation (de Teresa-Trueba et al., 2023; Lamm et al., 2024). While these show promise to eventually fully automate the segmentation process, even a state-of-the-art CNN requires large amounts of annotated training data to achieve high-quality segmentations.

To alleviate the above limitations, we developed ColabSeg, an interactive tool for visualizing, editing, and processing membrane segmentations as point clouds from outputs like the tensor voting tool TomoSegMemTV (Martinez-Sanchez et al., 2014) or CNNs (de Teresa-Trueba et al., 2023; Moebel et al., 2021). ColabSeg converts the cryo-ET segmentation into point clouds and enables access to a range of well-established computer-vision algorithms used in processing point clouds, such as statistical and eigenvalue-based edge outlier removal (Bazazian et al., 2015) without the need for scripting. It also provides tools to fit vesicles of various shapes and extended membranes for hole-free geometric calculations. The Jupyter notebook in which ColabSeg can be used enables easy remote visualization of the data and is platform-independent. The tool aims to alleviate the tedious and time-consuming task of generating ground truth data for training neural network-based approaches which can drastically improve performance. Furthermore, simple and commonly used analysis features are provided.

In this manuscript, we first introduce the general framework, and the software workflow and structure. We explain the features and the foundation of the tools provided for processing and editing segmentation files and showcase the different features of the software on four sample tomograms. Next, we show how ColabSeg can be used to generate a large set of training data for a 3D U-Net in DeePiCt (de Teresa-Trueba et al., 2023). Finally, we discuss possible shortcomings and future directions and give an outlook on how this work might serve as a stepping-stone towards a general CNN for segmentation where no further human intervention is required.

2. Methods

2.1. Implementation details

ColabSeg is written in Python with accompanying libraries NumPy (Harris et al., 2020) and SciPy (Virtanen et al., 2020). Features and algorithms of several state-of-the-art point cloud processing libraries, such as Open3D (Zhou et al., 2018) or pyntcloud, are directly used in ColabSeg. The GUI is written using ipywidgets together with py3Dmol (Rego and Koes, 2015) to view the point clouds, which can be rendered in any Jupyter notebook. The platform-independent nature of a Jupyter notebook ensures this software package can be run on any cluster or desktop computer with a working Python and Jupyter notebook installation. It can also be installed and used, in principle, remotely on a JupyterHub instance or with a Google Colab notebook. In this way, a notebook can also be run on an external cluster or server and subsequently viewed on a local desktop computer via an ssh connection on-the-go without copying large amounts of data to a local machine. The usage of the tool and its steps are outlined below.

2.2. Exemplary workflow for segmenting a reconstructed tomogram

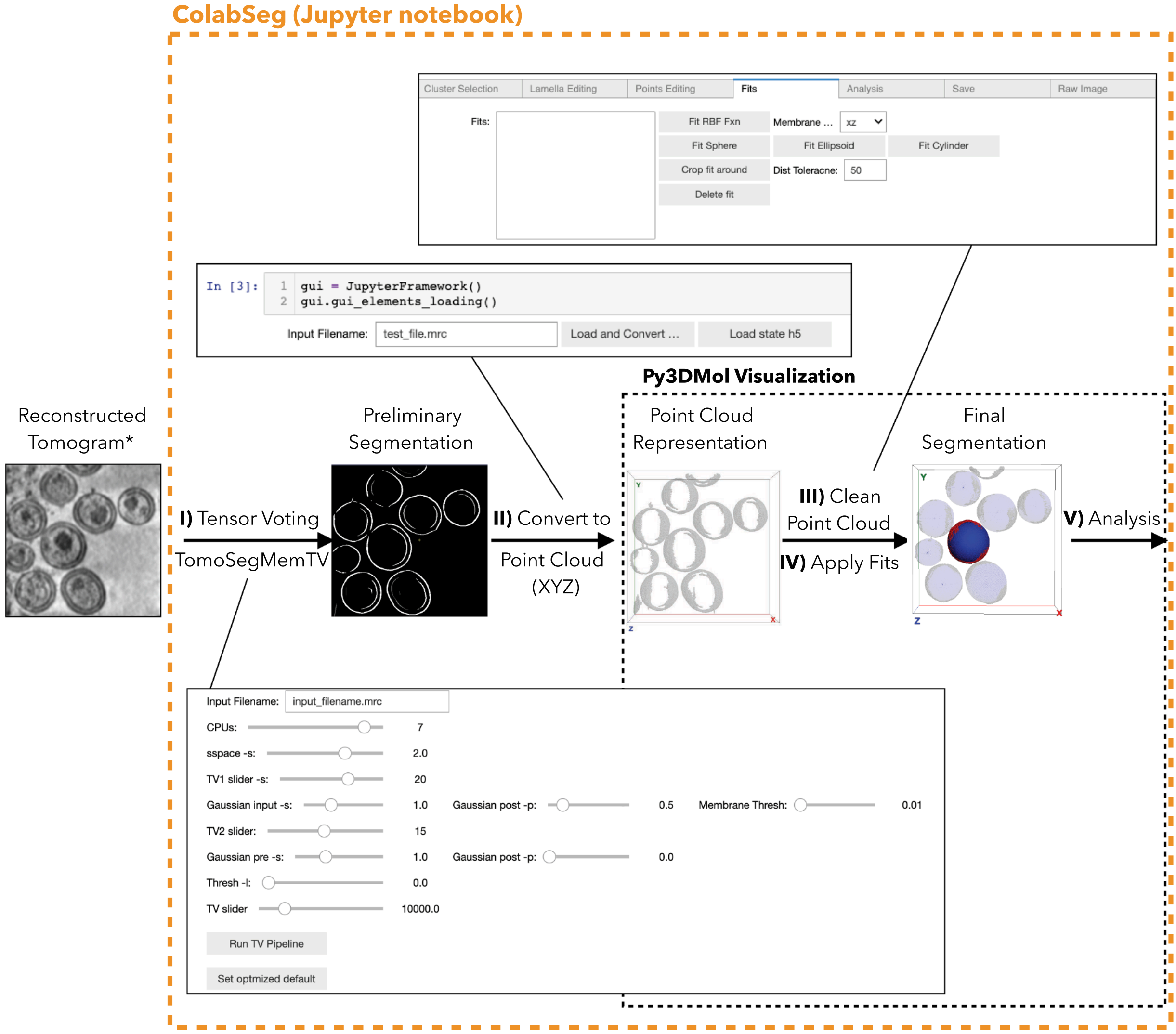
In the following, we describe an exemplary workflow of how to use ColabSeg starting from a tomogram generated using any reconstruction method, e.g. weighted back projection or SIRT (Fig. 1). In step 1, the tomogram is processed using existing segmentation software according to its specifications (Fig. 1, step I). This could be a CNN-based segmentation from a tool like MemBrain-Seg (Lamm et al., 2024) or the output from the well-established tensor voting tool TomoSegMemTV (Martinez-Sanchez et al., 2014; Moreno et al., 2022). ColabSeg provides a wrapper and GUI for TomoSegMemTV with optimized settings, which proved useful for running TomoSegMemTV for a number of example applications (the exact settings are detailed in the SI), and additionally a GUI for the recently developed MemBrain-seg tool (Lamm et al., 2024). Thus, a reconstructed tomogram is the only required input. All reported settings worked best for binned tomograms with a pixel size of approx. 26 Å, but an acceptable performance was achieved 13 Å as well. We note that a 13-26 Å pixel size is preferred to keep the computational load in an acceptable range. We also provide further instructions for tuning the performance of TomoSegMemTV starting from these optimized settings (see SI). Similar settings appear universally useful, as has been reported by Barad et al. (2023) during the preparation of this manuscript. The quality of the overall segmentation and processing with ColabSeg strongly relies on the results of TomoSegMemTV. Therefore optimizing the outputs here has been essential for quality.

We find that permissive thresholds in both the membrane detection and thresholding step in the settings (see SI) are advantageous to achieve more detailed and complete segmentations, even of smaller membrane features. While in this case more noise and false positives are often picked up, many of these excess segments can be removed using the functions of ColabSeg at a later stage. Manual and semi-automated removal of artifacts or false-positive is considerably faster than post-segmentation manual addition of missing features.

In step 2, ColabSeg accepts an MRC file containing a segmentation (binary mask). In this file, background voxels are assigned a value of zero, while foreground voxels are assigned positive integer values. The integer value of each voxel corresponds to the cluster they were assigned to. Such clustered segmentations can be generated using TomoSegMemTV's connected component algorithm (Fig. 1, Step II), or other software following a similar output format. ColabSeg implements a Density-based spatial clustering of applications with noise (DBSCAN) algorithm, which can be used to refine the clusters, or to determine new clusters from binary segmentations at a later stage. Colabseg is also capable of reading precomputed point clouds from text files from alternate sources, as well as the vertices from STL files. Then, based on the segmentation, ColabSeg computes a separate point cloud for each cluster, disregarding voxels that are labeled zero. Point clouds are simply collections of XYZ coordinates in space that describe the surface of an object. The advantage of working with coordinates is the ease of visualization and drastic compression of the information as the zero-valued voxels are no longer stored.

The point clouds and clusters can be directly visualized in the notebook using an interactive py3Dmol viewer (Rego and Koes, 2015). Point cloud coordinates are imported as dummy atoms. Importantly, to avoid memory issues due to a large number of points, the point cloud is downsampled for visualization purposes. The degree of downsampling is adjusted automatically depending on the overall count of points in the point cloud, where the point cloud visualized is limited to $<$ 200.000 by only showing every n-th point to allow for sufficiently smooth visualization in a Web browser. This downsampling procedure does not alter the underlying point cloud data.

In step 3, the point cloud derived from the segmentation can be cleaned and further refined using several utilities (Fig. 1, Step III). Detailed usage is described below and shown with different examples.



Membrane ...: xz
Dist Toleracne: 50

```
In [3]:  gui = JupyterFramework()
         gui.gui_elements_loading()
```

Input Filename: test_file.mrc

Input Filename: input_filename.mrc
CPUs: 7
sspace -s: 2.0
TV1 slider -s: 20
Gaussian input -s: 1.0 Gaussian post -p: 0.5 Membrane Thresh: 0.01
TV2 slider: 15
Gaussian pre -s: 1.0 Gaussian post -p: 0.0
Thresh -l: 0.0
TV slider 10000.0

Fig. 1. Overview of the Colabseg software and workflow. The demo tomogram contains eight HIV virions (EMDB ID: 13079) (Mattei et al., 2018) and is shown on the left (*an image filtered with the Gaussian filter provided in TomoSegMemTV is shown for better visibility). (I) The tensor voting workflow (Martinez-Sanchez et al., 2014) is applied to the reconstructed tomogram. This can be accessed and run in the provided GUI. The segmentations of the virions included included the matrix protein layer next to the membrane which needed further curation. (II) The segmentation in MRC format is then converted to a point cloud file. (III) Within the ColabSeg visualizer and editor, the point cloud can be processed, and any artifacts or undesired clusters can be removed using eigenvalue-based edge detection, statistical outlier removal, trimming, and reclustering tools. (IV) Fits can be applied to the clusters to interpolate spheres or extended planar membranes to fill holes and extrapolate beyond the data for accurate measurements. The final segmentation can be saved to disk from the main menu as XYZ coordinates in text file format or as binary mask in MRC file format. Alternatively, an HDF5 state file of the session can be saved throughout the editing process to resume editing later. (V) The segmentations and fits can be analyzed for some commonly used quantities, such as macromolecule-membrane distances.

This entails rapidly merging and deleting clusters, which is also possible in e.g., Amira, but avoids having to switch the software throughout the workflow and requiring another desktop software (Stalling et al., 2005). Clusters can be further split at this stage using the DBSCAN clustering algorithm. Point clouds can be filtered using a statistical outlier method or eigenvalue-based edge detection. The edges of point clouds can be trimmed to reduce noisy patches, which occur primarily where the edges of the lamellae in the tomograms were originally located. All methods can be combined to achieve a high-quality segmentation.

In step 4, idealized planar and spherical shapes can be fitted to planar membranes and spherical viruses or vesicles (Fig. 1, step IV). These two methods allow the fitting of a broad range of structures in biological samples and can be used to fill holes in existing segmentations, as required for quantitative spatial analysis. The details and usage are described below with the real biological sample data.

Finally, in step 5, it is possible to use the output segmentations for commonly used analysis procedures. These include surface normal estimates for, e.g., constrained particle picking, loading and analyzing macromolecule-membrane distances, and analyzing radii and properties of fitted spheres. Accompanying data and plots can be written to a file. This enables some commonly used properties to be quickly analyzed from within ColabSeg. Additional features, such as membrane curvature can be analyzed with third-party software such as pycurv or the morphometrics toolkit (Barad et al., 2023; Salfer et al., 2020)

ColabSeg allows saving the processed segmentations in various formats. Most commonly, MRC files can be written for use in further analysis pipelines. Additionally, the segmentations can be written as XYZ points or text files for subsequent custom analysis or visualization in other software packages such as VMD (Humphrey et al., 1996). It is also possible to save and load custom ColabSeg state files in HDF5 file format at any time during processing to save intermediate steps. All selections and settings of the GUI are saved and can be reconstructed from this file.

2.3. Advanced features for editing and cleaning point clouds

ColabSeg provides access to numerous point cloud processing algorithms commonly used in computer graphics to refine and correct false positives and split clusters. To demonstrate a possible use of these algorithms, we prepared a synthetic demo dataset consisting of a point cloud with two intersecting planes, representative of membranes, and Gaussian noise added to some of the points (Fig. 2). TomoSegMemTV (Martinez-Sanchez et al., 2014) and other tools often include numerous false positives outside the cellular region, such as ice contaminants or fiducial markers. Additionally, at the edge of the tomogram or cellular slice (in the case of lamellae prepared by focused ion beam milling), the data is ambiguous, resulting in some noise in the corresponding segmentation. Moreover, point clouds are occasionally falsely grouped by the connected component algorithm provided by TomoSegMemTV (Martinez-Sanchez et al., 2014) due to noise.

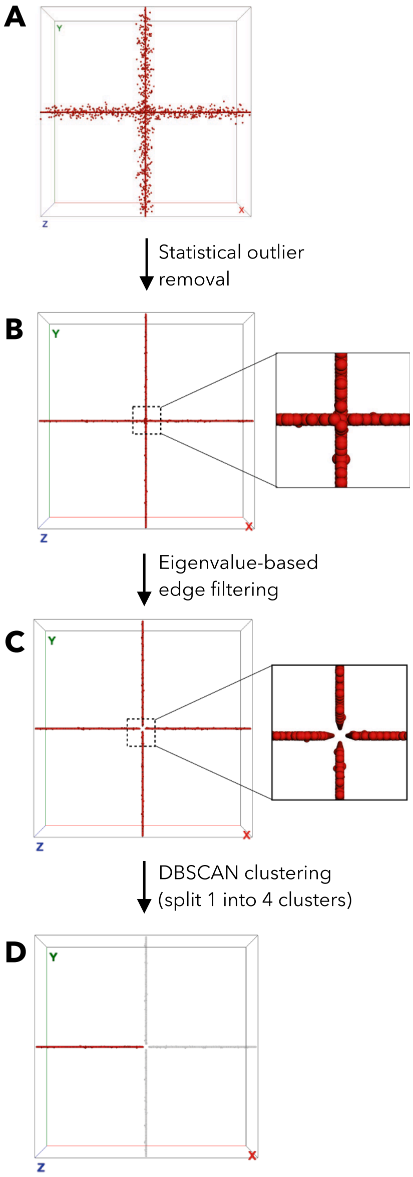

A
Y
Z
X
Statistical outlier removal
B
Y
Z
X
Eigenvalue-based edge filtering
C
Y
Z
X
DBSCAN clustering (split 1 into 4 clusters)
D
Y
Z
X

Fig. 2. Overview of implemented point cloud processing and cleaning features using a synthetic MRC file. Shown are snapshots from the ColabSeg 3D viewer. A selected cluster is shown in red. (A) The initial file consists of two intersecting planes with 10000 points having Gaussian noise added to them. The initial synthetic MRC file has a cross shape. (B) Statistical outlier removal for data cleaning as implemented in the Open3D library. (C) Eigenvalue-based edge filtering example. (D) DBSCAN algorithm for cluster separation. All features can be combined in various order to solve more complex segmentation issues resulting from inaccuracies in the tensor voting. The view render in ColabSeg dynamically downsamples the number of points to avoid speed and memory issues.

To remedy such issues, we provide statistical outlier removal based on the implementation in Open3D (Zhou et al., 2018) (Fig. 2). The free parameters include the number of the k-nearest neighbors, which are used to calculate the average distance for a given point. The threshold is given by the standard deviation from the distribution of distance, where lower values result in stricter cutoffs. Points that constitute background noise are removed.

We also provide a more advanced eigenvalue-based outlier removal methodology. Bazazian et al. (2015) show that the eigenvalues of the covariance matrix that are defined by each point’s k-nearest neighbors are an effective way of filtering points constituting an edge with a sharp change in curvature. For most examples in biological systems which are flat, extended membranes such as plasma membranes, nuclear envelopes, and other large structures, this method works particularly well (see below Fig. 4C). Since membranes are most commonly represented in smooth extended surfaces, most artifacts manifest as unordered, highly distorted point clouds. This analysis outperforms other methods for edge detection such as PCA analysis or triangulation-based methods combined with clustering, as reported in Bazazian et al., 2015. It does not rely on the initial clustering step, which would add additional free parameters to the analysis and depend highly on the point cloud density. In our test case (Fig. 2) the intersection of the two planes is easily filtered with this approach. Good results are also observed for spherical vesicles or tubular structures if large enough continuous pieces of the membrane are captured in the reconstructed volume (Fig. 4).

To enable the re-clustering of clusters initially assigned through the connected-component algorithm in TomoSegMemTV, we provide the DBSCAN algorithm (Ester et al., 1996) which is implemented in the open3D library (Zhou et al., 2018). The DBSCAN algorithm is commonly used for clustering 3D point clouds which can be arbitrary non-convex-shaped clusters, that are in close proximity and contain noise. The algorithm is furthermore agnostic to the number of clusters (Ester et al., 1996), analogous to a connected component algorithm in TomoSegMemTV. In our test case (Fig. 2) the initial single cluster can now be split into four separate clusters. This approach was also chosen for some of the applications shown below (Fig. 4B). In real applications, tuning of the parameters also enables the filtering of outlier points (Ester et al., 1996).

TomoSegMemTV (Martinez-Sanchez et al., 2014) and other tools often include numerous false positives outside of the cellular region. These can easily be removed by trimming the edge using a corresponding tool in ColabSeg (Supporting Information Fig. S4). Removing points at the respective upper and lower bound drastically improves the quality of the segmentation. Finally, in case the cellular or lamellae regions are not corrected for a tilt offset at the reconstruction step, this can be done in ColabSeg by performing a plane fit through all points of the membrane and subsequently aligning with the z-axis (SI Fig. S4).

2.4. Fitting procedures for hole filling

Many analyses of cryo-ET data rely on spatial measurements such as Euclidian distances, e.g., from nearest membranes. Here complete, hole-free membrane representations are required to achieve correct and robust results. However, many segmentations have holes or missing pieces that impede these calculations, which is another reason for many researchers to resort to manual segmentations. Therefore, we provide access to radial basis function (RBF) plane fits for extended membranes and sphere fits for vesicles (Fig. 4B) - two common geometries for membrane structures found in tomograms. These enable extrapolating and filling missing pieces in segmentations for accurate calculations. Within the GUI, one or multiple segments can be selected, and the fit is automatically performed. The SciPy (Virtanen et al., 2020) RBF fit and 'lstsq' function are used to perform the fits for near-planar membranes. To fit spheres, ellipsoids and cylinders to point clouds, an initial estimate of the shape parameters is obtained using the eigenvectors and eigenvalues of the point cloud's covariance matrix. The initial parameters are subsequently numerically optimized using Nelder-Mead optimization implemented in SciPy ((Virtanen et al., 2020), v1.10.1).

2.5. Analysis Functionality

ColabSeg is shipped with several analysis features for the segmentations that are commonly used in biological studies (Fig. 3) The spherical fits can be analyzed to yield distributions of vesicle radii. Such analyses for instance hav been beneficial for the understanding of viral morphology or synaptic vesicles. ColabSeg allows the generalization of estimated membrane normals on the approximated surface of the clusters to enable restrained particle picking, a feature that has been successfully used to identify various membrane coats. ColabSeg also allows loading protein coordinates from other sources, such as template matching, deep-learning methods, or manual annotation, and calculating minimal distance distributions with respect to a target membrane. Such analyses are commonly performed for a broad range of biological systems (Lembo et al., 2023; Ferreira et al., 2023).

2.6. Napari integration and validation

ColabSeg provides integration with Napari (2023), which enables visualizing the point clouds and fits in the context of the input tomograms (see documentation). To this end, ColabSeg provides simple on-click conversion of the data structures directly to Napari-readable objects and automatic loading of the segmentations, fits, and loaded protein positions in Napari. Napari can then be used to visualize the input tomograms overlaid with ColabSeg segmentations. Additionally, users can use Napari internal features, e.g., to manually edit individual points in the point clouds, or to interface with other tomography visualization and analysis tools such as blik (Gaifas et al., 2023). Point clouds edited in Napari can also be imported back to ColabSeg.

2.7. Using ColabSeg to generate training data for the CNN of DeePiCt

The membranes of ten *Mycoplasma pneumoniae* tomograms (Xue et al., 2022) were segmented manually using Amira (version 2022.1,

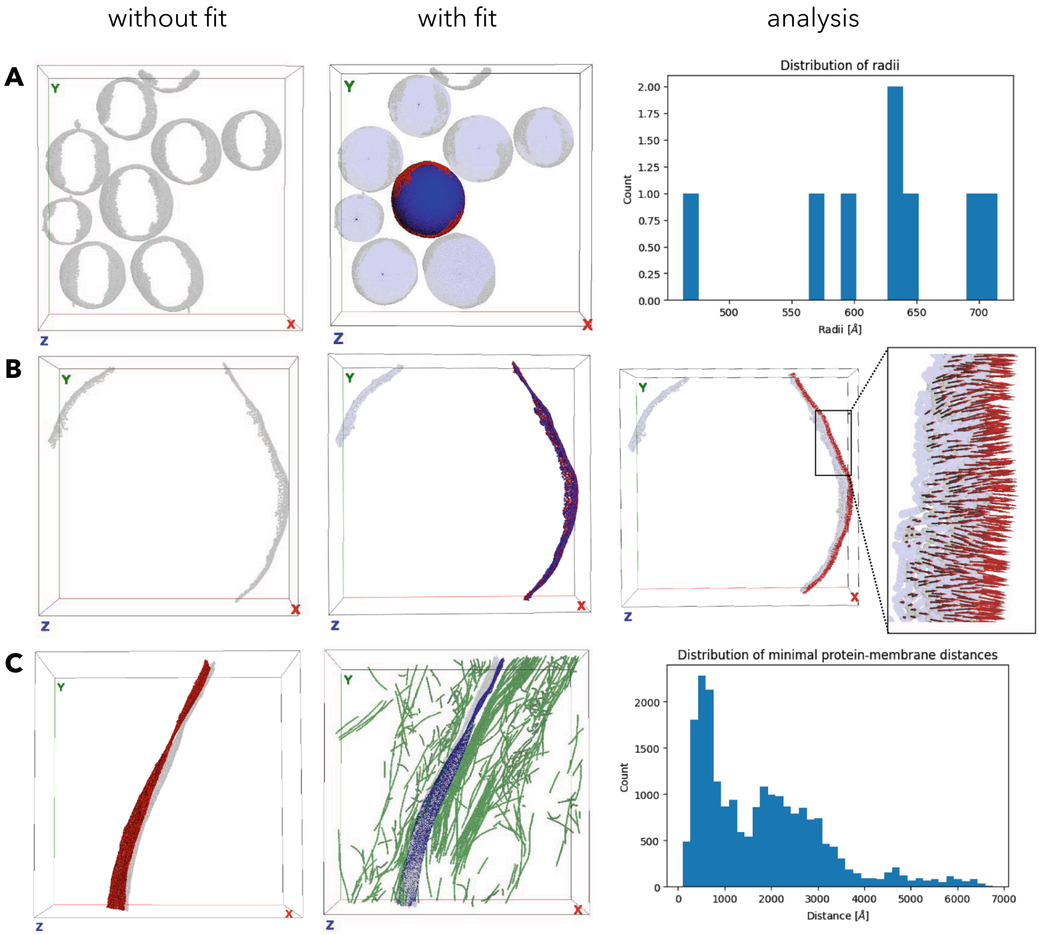


Fig. 3. Examples of fitting and analysis functionalities in ColabSeg. (A) Sphere fits for eight HIV virions. The fits are shown in blue. The original segmentation derived from the cryo-ET data is shown in grey. An example virion cluster and the corresponding fit are selected in dark red and blue, respectively. Top view snapshots from the ColabSeg viewer are shown before and after the processing. The output from the radii plotting analysis feature is shown on the right. (B) Radial basis function fit on two outer membranes of a *Mycoplasma pneumoniae* cell. The initial segmentation is shown in grey and the fits in blue. One example cluster and corresponding fit are shown in dark red and blue. The fit was trimmed to only include points within 50 Å of the input data. The analysis constitutes the estimated normals on the surface of the cluster. The red arrows visualize the calculated normals which can be written to disk. (C) An example of two adjacent plasma membranes as reported in ref. (Lembo et al., 2023). Fits are shown in blue. Actin filaments sampled as equidistant points loaded from the analysis tab are shown in green. The protein-membrane distances were computed using the analysis tab.

Thermo Fischer). In addition, membranes in 50 tomograms were segmented using TomoSegMemTV (Martinez-Sanchez et al., 2014) and cleaned using ColabSeg (detailed below). The DeePiCt 3D U-Net (de Teresa-Trueba et al., 2023) was trained using three different types of training sets and varying amounts of tomograms in the respective set: i) using five manually segmented tomograms from Amira, ii) using 5, 10, 25, or 50 outputs from TomoSegMemTV directly without cleaning or curation, or iii) 5, 10, 25, 50 tomograms semi-automatically curated with ColabSeg. The remaining five manually segmented tomograms not used for training were used for validation. The DeePiCt model was trained at a pixel size of 13.6 Å with box size and overlap of 64 and 6 pixels, respectively. The default U-net convolutional network of DeePiCt (de Teresa-Trueba et al., 2023) was used, with a depth of 2 and 16 initial convolutions. In each epoch, 80% of the data was used for training and validated using the dice coefficient against the remaining 20% of the data. A total of 300 epochs were run for training each model, but training normally converged within the first 50 epochs. The resulting models were used to predict the membrane in the validation set, and the output was filtered to a binarized map using a threshold of 0.4, excluding any cluster containing less than 1000 voxels.

3. Results and Discussion

3.1. Biological examples for editing and fitting segmentations

We tested ColabSeg to refine membrane segmentations from various cryo-ET datasets. Our examples span HIV virions (Mattei et al., 2018), the inner membrane complex (IMC) of Plasmodium falciparum (Ferreira et al., 2023), and *Mycoplasma pneumoniae* cells (Xue et al., 2022; Tegunov et al., 2021). Additionally, an early version of the tool has been used to segment plasma membranes and analyze membrane-actin distance distributions(Lembo et al., 2023). Each example brings diverse challenges in membrane segmentation which are difficult to address with fully automated tools or are too laborious to be performed solely manually due to the number of tomograms in each of the datasets. Starting from reconstructed tomograms, we use ColabSeg together with TomoSegMemTV (Martinez-Sanchez et al., 2014) to arrive at fitted target segmentations (Fig. 4).

First, we explored an HIV virions dataset (EMDB ID: 13079, (Mattei et al., 2018)). We used TomoSegMemTV to segment viral membranes with a low threshold (Fig. 4A). While segmenting viruses is in principle straightforward, in this case, a layer of viral matrix proteins is picked up

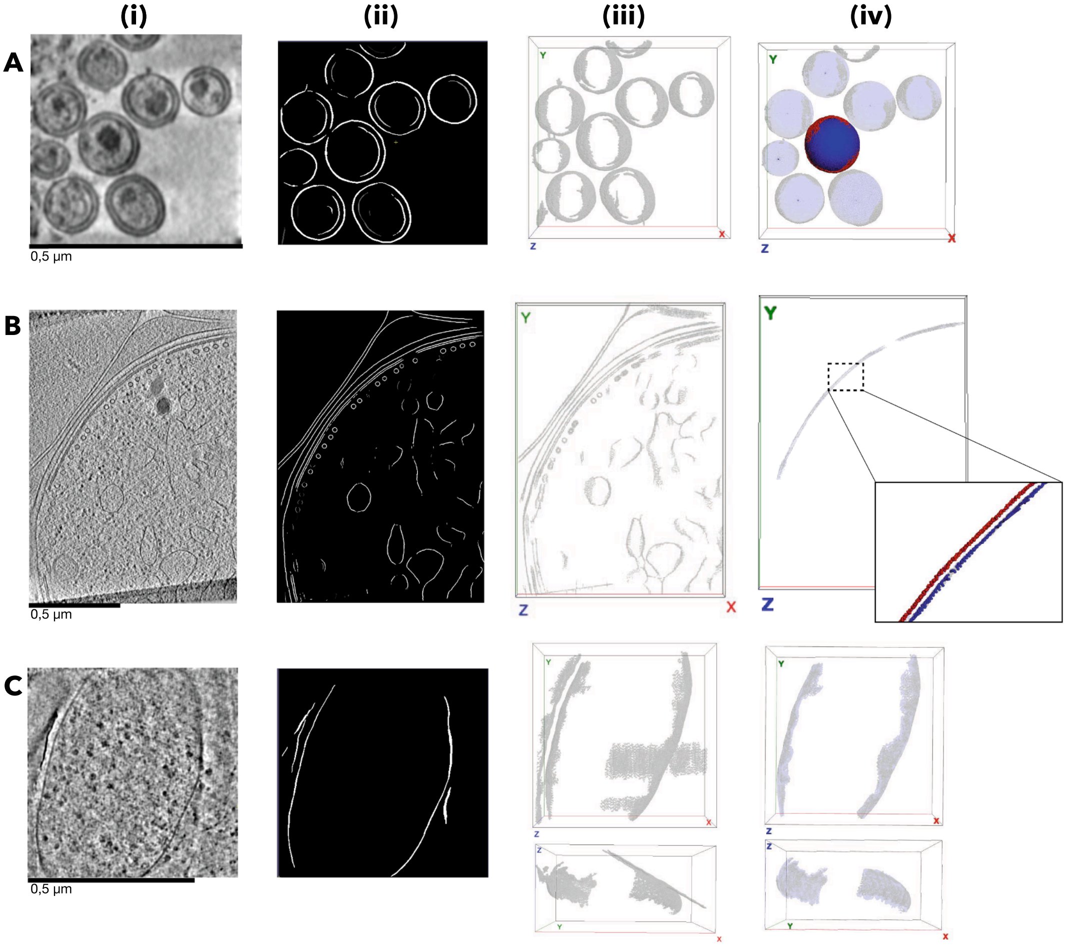


Fig. 4. Application of ColabSeg on biological example systems. (A) the separation and fit of individual HIV virions (EMDB ID: 13079, (Mattei et al., 2018) and removal of a false positive detected matrix protein layer, (B) the extraction of the *Plasmodium falciparum* inner membrane complex (IMC), a narrow double bilayer structure, for further analysis as used in Ferreira et al. (2023), and, (C) the clean-up and artifact removal (carbon edge) from a *Mycoplasma pneumoniae* tomogram. All panels show (from left to right) a central slice from the reconstructed tomogram (shown is the filtered first step of the TomoSegMemTV pipeline) (i), the raw output of the TomoSegMemTV tool without further curation, selection, or cleaning (ii), the TomoSegMemTV output, imported in ColabSeg and visualized as a point cloud (iii), and the final processed and fit point cloud for further use in analysis (iv). Here, grey points indicate the processed data, blue the fits, and red highlights a cluster of interest. Each sample data set combines the use of multiple ColabSeg features including cluster merging, deletion, edge filtering, statistical outlet removal, and various fits to achieve the represented output. The detailed workflow for each of the images is outlined in the provided user guide.

by TomoSegMemTV as membrane, which requires curation and additional processing steps . Also in some instances, the membranes of two independent virions in close proximity touched and were determined as one cluster by the TomoSegMemTV connected component algorithm. Therefore, filtering using the statistical outlier tool and reclustering with the DBSCAN algorithm was necessary. We also used ColabSeg to remove clusters that are not virion membranes. Subsequently, we applied a sphere fit to all virions. These could enable further distance measurements and the use of the fit parameters for spherically constrained 3D subtomogram averaging (Förster et al., 2005).

Next, we used the tools from ColabSeg to extract the inner membrane complex (IMC) of *Plasmodium falciparum* to enable accurate measurements of their distance to microtubules (Fig. 4B) (Ferreira et al., 2023). The IMC is an intricate double membrane structure in *Plasmodium falciparum* which forms below the plasma membrane and plays an important role in the life cycle of the parasite (Ferreira et al., 2021). Here, we first ran TomoSegMemTV to extract the main features (plasma and intracellular membranes, and microtubules), and then used a combination of edge-filtering, edge trimming, statistical outlier removal, and DBSCAN re-clustering to select and segment both bilayers of the IMC (Ferreira et al., 2023). Workflows using solely TomoSegMemTV resulted in either patchy membranes or large amounts of excess noise, which would have interfered with the measurement (Fig. 4B). Moreover, pronounced structures, such as the microtubules, were picked up by TomoSegMemTV, which also had to be removed - a common occurrence with the TomoSegMemTV. In particular, segmentations of tomograms acquired from thick lamella profited from the edge trimming procedure, in most cases removing a bulk of false positive noise. We then used the RBF fit function to fill any holes in the membranes to avoid errors in the distance measurements. We repeated this workflow for a total of 73 tomograms, of which the best were used for further processing. These measurements were then used to show the high degree of consistency in membrane-microtubule distances, indicating the presence of a linker, which would not have been possible without accurate segmentation of the IMC (Ferreira et al., 2023).

We also demonstrate the utility of the eigenvalue-based edge filtering on a difficult-to-segment tomogram from a data set of *Mycoplasma pneumoniae* (Fig. 4C). Occasionally, edges of the grid support film or other high contrast features are falsely picked up in the segmentation process resulting in false positive detection of membranes (Fig. 4C, left (i, ii)). When using TomoSegMemTV, this leads to artifacts where the *Mycoplasma pneumoniae* membrane and carbon support are detected as one continuous cluster and so far had to be separated or segmented manually. Using the eigenvalue edge filtering we can remove such artifacts from the segmentations (Fig. 4C, right (iii, iv)): the edge detection removed points along the distinct edge where the feature intersects with the membrane. Subsequently, we were able to split these two clusters by using the DBSCAN clustering algorithm and using the RBF fit function to fill the resulting hole (Fig. 4C, right). We used Colabseg to segment over 50 tomograms for use as training data for the DeePiCt neural network.

These examples demonstrate the utility and ease of use of ColabSeg for a variety of tasks. By combining different features of the software, we were able to resolve a broad range of issues resulting from the initial segmentations and could enable accurate analysis of many tomograms in a short amount of time. Overall, we note that ColabSeg is strongly dependent on the initial segmentation. The quality of the segmentations such as the output from TomoSegMemTV (Martinez-Sanchez et al., 2014), is a decisive factor for the successful segmentation of membranes. For this reason, we tested various settings for TomoSegMemTV and provided the best ones to the user. We found TomoSegMemTV performs particularly well with a permissive threshold in the membrane 'surfaceness' step to capture all membrane features irrespective of their size. Additionally, thick cellular regions have a particularly low signal-to-noise ratio and usually result in poorer segmentations, which are difficult to clean. In most cases, some manual intervention will be required to clean the remaining segmented clusters, and ColabSeg greatly facilitates this task. More recent deep learning-based tools with a focus on membrane segmentations, such as MemBrain-Seg (Lamm et al., 2024) can also be used for segmentation and do not require parametrization and can handle noisy tomograms given the training dataset is sufficiently varied.

3.2. Training data generated by ColabSeg improve deep learning performance

ColabSeg was developed not only to ease the semi-automated segmentation and directly enable analysis of the data, but also to generate data for training convolutional neural network (CNN) approaches faster and increasing throughput of cryo-ET data analysis. Recently, open-source software solutions for the segmentation of tomograms using 2D and 3D U-Nets have been made available (de Teresa-Trueba et al., 2023; Moebel et al., 2021). The quality of the segmentations from the CNNs strongly depends on the available training data. To date, most CNNs designed for cryo-ET rely on very few (one to ten) segmented tomograms as training data, which are often generated by manual segmentation with software like Amira. The resulting trained models then only perform well for a narrow scope of similar data, if at all. To achieve larger transferability and higher quality segmentation, we propose using the ColabSeg pipeline to quickly generate a sufficiently large training set for CNN-based approaches.

To test this, we used the ColabSeg pipeline in conjunction with the recently developed DeePiCt network (de Teresa-Trueba et al., 2023). First, we use TomoSegMemTV and ColabSeg to segment several hundreds of tomograms from *Mycoplasma pneumoniae*, filtering artifacts and false positive cases with the methods introduced above. By doing so, we prepared several hundred segmentations of tomograms for use as training data. Next, we trained DeePiCt on several different subsets of the raw segmentations, the cleaned segmentationss, and some manual segmentations described above to assess differences in performance. A set of 5 manually segmented tomograms were used as a validation data set.

We found that using a large number of segmented tomograms for training drastically improves the performance of the network for the same validation dataset (Fig. 5). Importantly, a network trained on a dataset consisting of segmentations curated with ColabSeg outperformed a network on a small dataset segmented manually using Amira. Similarly, using the outputs segmented solely with TomoSegMemTV performed less favorably and resulted in more false positives than when the data was inspected and cleaned with ColabSeg. Curation of the best cluster by visual inspection with ColabSeg would improve the quality of the predictions, highlighting the necessity to visually inspect and process the segmentations. With this larger training data set, we were able to reliably filter out artifacts present due to the segmentation of the support film edge, which is visible in many tomograms. The corresponding DeePiCt model could reliably segment membranes from unseen data at a level similar to manual annotation.

Mycoplasma pneumoniae is a comparably straightforward test system. Since it is a small bacterium and contains no complex membrane-bound organelles, the segmentation task is significantly easier than for other cell types. Similarly, viruses such as HIV, influenza A virus, Ebola, or other isolated membrane-bound structures can easily be analyzed using ColabSeg. However, curation of inputs from TomoSegMemTV was still necessary to achieve satisfactory results. Eukaryotic cells pose an even bigger challenge for accurate segmentation, for example, the highly curved membranes such as tubular endoplasmic reticulum or the cristae of mitochondria. Here, a combination of computer-aided and manual segmentation in ColabSeg will enable gathering training data of sufficient quality for accurate CNN-based segmentation of such cellular structures.

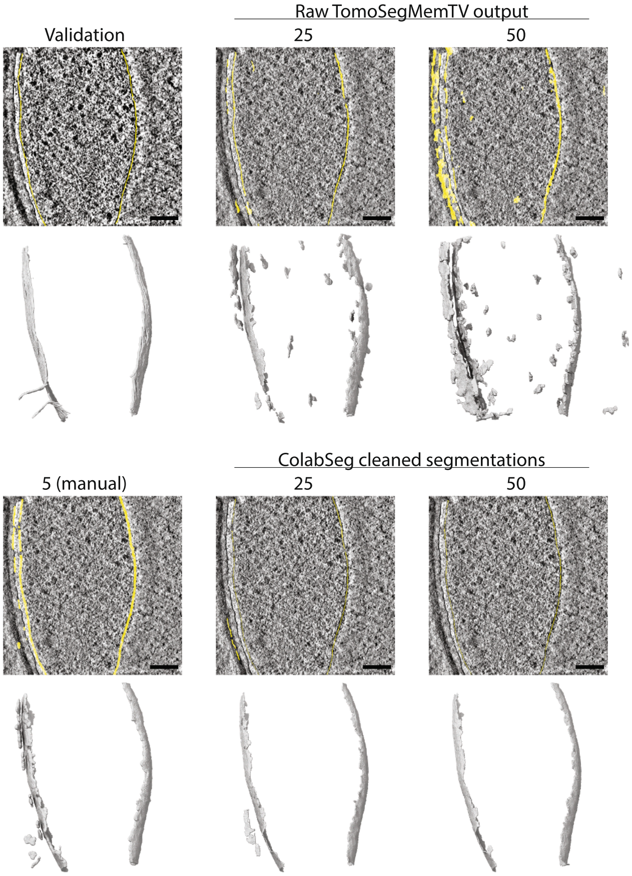


Fig. 5. Use of ColabSeg data to improve CNN training for fully automated membrane segmentation. Respective top rows show a 2D slice of the tomogram of *Mycoplasma pneumoniae* with the segmentation as a yellow overlay. The respective bottom rows show 3D volume renders of the entire segmentation. Shown are a validation dataset (top left), outputs from DeePiCt using 25 or 50 raw TomoSegMemTV output segmentations as the training set (top right), using five manually segmented tomograms as the training set (bottom left), and using 25 or 50 segmentations processed with ColabSeg as the training set. Scale bars correspond to 100 nm.

4. Conclusions and Outlook

We presented ColabSeg, a Jupyter notebook-based GUI for visualizing, editing, and post-processing segmentations of membranes in cryo-ET data. We enable users to generate initial segmentations with TomoSegMemTV or MembrainSeg implemented in our tool, quickly filter relevant segments, remove noise as well as generate fits to fill holes and perform downstream quantitative measurements. We provide numerous algorithms used in point cloud processing accessible in a simple interface without requiring any scripting. Advanced users can build semi-automatic workflows by using the provided classes and the accompanying image-processing libraries. We show that even though some semi- and fully automatic methods for segmentation exist, user intervention and curation are still necessary to arrive at high-quality segmentations. We demonstrated the usefulness of ColabSeg on a range of use cases, that address a broad range of scientific questions (Ferreira et al., 2023; Lembo et al., 2023). Finally, we show how ColabSeg can be used in conjunction with a CNN, DeePiCt, for generating sufficient training data for more accurate automated segmentation.

Currently, ColabSeg is primarily designed and set up to visualize and process membrane segmentations, but it also enables visualization of the coordinates of macromolecules that have been acquired from other sources, such as manual picking or template matching. In the future, the GUI could be extended in a straightforward way to include visualization of the accompanying structural models. Since the underlying 3D visualization software, py3Dmol, was intended to represent proteins and can handle complex representations (Rego and Koes, 2015), it might also become a useful tool to visualize and analyze protein complexes, measure their arrangement, and analyze their properties in segmented cryo-ET data. Our first set of functionalities to alleviate analysis tasks related to membrane segmentation is an initial step in this direction.

The backend functionality can easily be adapted to interface with other visualization software packages with some minor changes to the interface. ColabSeg provides a lightweight Napari integration that enables transferring results between Napari and ColabSeg, and visualizing the segmentations together with the raw tomograms. This also enables interacting with other Napari extensions and using their functionality. Developers can find API documentation and instructions to expand the GUI in a straightforward way in the user guide. The lightweight design of the software and browser-based visualization possibilities might be deployed on web servers or on Google Colab for easy access and analysis. This might make sharing these algorithms, and developing and deploying online servers for the processing of data easier.

In the future, ColabSeg in conjunction with other segmentation methods will hopefully speed up the process of membrane segmentation and as a result, help generate a larger annotated set of cryo-ET data. Although the speed of user-assisted segmentation will always depend on prior user experience with the particular software and the level of familiarity with tomogram analysis in general, we envision this tool as a stepping stone to lower the barrier for experimentalists to annotate their raw tomograms and thus populate a growing body of data for use as training data for deep learning methods such as DeePiCt, DeepFinder or MembrainSeg (de Teresa-Trueba et al., 2023; Moebel et al., 2021; Lamm et al., 2024), which is still lacking to date. Eventually, CNN models will be trained on sufficiently broad data such that near error-free and fully automated segmentation should be available in the future.

5. Software Availability

ColabSeg is available under an Apache License 2.0 license from GitHub (https://github.com/KosinskiLab/colabseg) free of charge and can be installed as a Python package. It can be used from a Jupyter notebook or Colab notebook (setup required by the user). Third-party contributions are welcome as pull requests on GitHub.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code of the software is available at: https://github.com/KosinskiLab/colabseg. Cryo-electron tomography datasets reused in this work are available in the Electron Microscopy Public Image Archive database under the ID: EMPIAR-10499 for M. pneumoniae cells and the Electron Microscopy Data Bank under the ID: EMD-13079. for the HIV particles.

Acknowledgements

The work was supported by EMBL and a research fellowship from the EMBL Interdisciplinary Postdoc (EIPOD) Programme under Marie Curie Cofund Actions MSCA-COFUND-FP (grant agreement number: 847543) to MS. RKJ was supported by a postdoctoral fellowship from the DFF (grant number 0164-00010A). VM and JK acknowledge funding from the CSSB flagship project Plasmofraction. We thank Dr. Josie L. Ferreira for providing multiple tomograms of *Plasmodium falciparum* stages and for helpful discussions. We thank Liang Xue for providing multiple tomograms of *Mycoplasma pneumoniae*. We thank Alba Diz-Muñoz and Dorothy Cheng for tomograms with actin cortex segmentations. We thank Herman Fung, Veijo T. Salo, Sara K. Goetz, and Anastasiia Babenko for testing the software and for critical feedback. We thank Thomas Hoffmann for discussions and, together with the rest of EMBL IT, providing the IT infrastructure.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.jsb.2024.108067.

References

Barad, B.A., Medina, M., Fuentes, D., Wiseman, R.L., Grotjahn, D.A., 2023. Quantifying organellar ultrastructure in cryo-electron tomography using a surface morphometrics pipeline. J. Cell Biol. 222 (4) https://doi.org/10.1083/jcb.202204093.

Bazazian, D., Casas, J.R., Ruiz-Hidalgo, J., 2015. Fast and Robust Edge Extraction in Unorganized Point Clouds. Int. Conf. Digit. Image Comput. Tech. Appl. IEEE, pp. 1–8. https://doi.org/10.1109/DICTA.2015.7371262.

de Teresa-Trueba, I., Goetz, S.K., Mattausch, A., Stojanovska, F., Zimmerli, C.E., Toro-Nahuelpan, M., Cheng, D.W.C., Tollervey, F., Pape, C., Beck, M., Diz-Muñoz, A., Kreshuk, A., Mahamid, J., Zaugg, J.B., 2023. Convolutional networks for supervised mining of molecular patterns within cellular context. Nat. Methods 20 (2), 284–294. https://doi.org/10.1038/s41592-022-01746-2.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD. https://doi.org/10.5555/3001460.3001507.

Ferreira, J.L., Heincke, D., Wichers, J.S., Liffner, B., Wilson, D.W., Gilberger, T.-W., 2021. The Dynamic Roles of the Inner Membrane Complex in the Multiple Stages of the Malaria Parasite. Front. Cell. Infect. Microbiol. 10 (January), 1–12. https://doi.org/10.3389/fcimb.2020.611801.

Ferreira, J.L., Pražák, V., Vasishtan, D., Siggel, M., Hentzschel, F., Binder, A.M., Pietsch, E., Kosinski, J., Frischknecht, F., Gilberger, T.W., Grünewald, K., 2023. Variable microtubule architecture in the malaria parasite. Nat. Commun. 14 (1), 1216. https://doi.org/10.1038/s41467-023-36627-5.

Förster, F., Medalia, O., Zauberman, N., Baumeister, W., Fass, D., 2005. Retrovirus envelope protein complex structure in situ studied by cryo-electron tomography. Proc. Natl. Acad. Sci. 102 (13), 4729–4734. https://doi.org/10.1073/pnas.0409178102.

Gaifas, L., Timmins, J., Gutsche, I., 2023. blik: an extensible napari plugin for cryo-et data visualisation, annotation and analysis, bioRxiv: https://www.biorxiv.org/content/early/2023/12/07/2023.12.05.570263.full.pdf.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585 (7825), pp. 357–362. arXiv:2006.10256, https://doi.org/10.1038/s41586-020-2649-2.

Humphrey, W., Dalke, A., Schulten, K., 1996. VMD: Visual molecular dynamics. J. Mol. Graph. 14 (1), 33–38. https://doi.org/10.1016/0263-7855(96)00018-5.

Lamm, L., Righetto, R.D., Wietrzynski, W., Pöge, M., Martinez-Sanchez, A., Peng, T., Engel, B.D., 2022. MemBrain: A deep learning-aided pipeline for detection of membrane proteins in Cryo-electron tomograms. Comput. Methods Programs Biomed. 224, 106990. https://doi.org/10.1016/j.cmpb.2022.106990.

Lamm, L., Zufferey, S., Righetto, R.D., Wietrzynski, W., Yamauchi, K.A., Burt, A., Liu, Y., Zhang, H., Martinez-Sanchez, A., Ziegler, S., Isensee, F., Schnabel, J.A.,Engel, B.D., Peng, T., 2024. Membrain v2: an end-to-end tool for the analysis of membranes in cryo-electron tomography, bioRxiv: https://www.biorxiv.org/content/early/2024/01/05/2024.01.05.574336.full.pdf.

Lebbink, M.N., Geerts, W.J., van der Krift, T.P., Bouwhuis, M., Hertzberger, L.O., Verkleij, A.J., Koster, A.J., 2007. Template matching as a tool for annotation of tomograms of stained biological structures. J. Struct. Biol. 158 (3), 327–335. https://doi.org/10.1016/j.jsb.2006.12.001.

Lebbink, M.N., Geerts, W.J., van Donselaar, E., Humbel, B.M., Post, J.A., Hertzberger, L. O., Koster, A.J., Verkleij, A.J., 2008. Electron tomography and template matching of biological membranes. In: EMC 2008 14th Eur. Microsc. Congr. 1–5 Sept. 2008, Aachen, Ger., Springer, Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 83–84. https://link.springer.com/chapter/10.1007/978-3-540-85228-5_42.

Lembo, S., Strauss, L., Cheng, D., Vermeil, J., Siggel, M.,Cheng, W.C.D., Vermeil, J., Siggel, M., Toro-Nahuelpan, M., Chan, C.J., Kosinski, J., Piel, M., Du Roure, O., et al., 2023. The distance between the plasma membrane and the actomyosin cortex acts as a nanogate to control cell surface mechanics, bioRxiv 2001–2023. https://doi.org/10.1101/2023.01.31.526409.

Lučić, V., Rigort, A., Baumeister, W., 2013. Cryo-electron tomography: The challenge of doing structural biology in situ. J. Cell Biol. 202 (3), 407–419. https://doi.org/10.1083/jcb.201304193.

Luengo, I., Darrow, M.C., Spink, M.C., Sun, Y., Dai, W., He, C.Y., Chiu, W., Pridmore, T., Ashton, A.W., Duke, E.M., Basham, M., French, A.P., 2017. SuRVoS: Super-Region Volume Segmentation workbench. J. Struct. Biol. 198 (1), 43–53. https://doi.org/10.1016/j.jsb.2017.02.007.

Mahamid, J., Pfeffer, S., Schaffer, M., Villa, E., Danev, R., Kuhn Cuellar, L., Forster, F., Hyman, A.A., Plitzko, J.M., Baumeister, W., 2016. Visualizing the molecular sociology at the HeLa cell nuclear periphery. Science (80-.). 351 (6276), pp. 969–972. https://doi.org/10.1126/science.aad8857.

Martinez-Sanchez, A., Garcia, I., Asano, S., Lucic, V., Fernandez, J.-J., 2014. Robust membrane detection based on tensor voting for electron tomography. J. Struct. Biol. 186 (1), 49–61. https://doi.org/10.1016/j.jsb.2014.02.015.

Martinez-Sanchez, A., Kochovski, Z., Laugks, U., Meyer zum Alten Borgloh, J., Chakraborty, S., Pfeffer, S., Baumeister, W., Lučić, V., 2020. Template-free detection and classification of membrane-bound complexes in cryo-electron tomograms. Nat. Methods 17 (2) (2020) 209–216, https://doi.org/10.1038/s41592-019-0675-5.

Mattei, S., Tan, A., Glass, B., Müller, B., Kräusslich, H.G., Briggs, J.A., 2018. High-resolution structures of HIV-1 Gag cleavage mutants determine structural switch for virus maturation. Proc. Natl. Acad. Sci. U.S.A. 115 (40), E9401–E9410. https://doi.org/10.1073/pnas.1811237115.

Moebel, E., Martinez-Sanchez, A., Lamm, L., Righetto, R.D., Wietrzynski, W., Albert, S., Larivière, D., Fourmentin, E., Pfeffer, S., Ortiz, J., Baumeister, W., Peng, T., Engel, B. D., Kervrann, C., 2021. Deep learning improves macromolecule identification in 3D cellular cryo-electron tomograms. Nat. Methods 18 (11), 1386–1394. https://doi.org/10.1038/s41592-021-01275-4.

Moreno, J.J., Garzón, E.M., Fernández, J.J., Martínez-Sánchez, A., 2022. HPC enables efficient 3D membrane segmentation in electron tomography. J. Supercomput. https://doi.org/10.1007/s11227-022-04607-z.

Napari contributors, napari: a multi-dimensional image viewer for python, 2023, https://doi.org/10.5281/zenodo.3555620.

Pfeffer, S., Dudek, J., Schaffer, M., Ng, B.G., Albert, S., Plitzko, J.M., Baumeister, W., Zimmermann, R., Freeze, H.H., Engel, B.D., Förster, F., 2017. Dissecting the molecular organization of the translocon-associated protein complex. Nat. Commun. 8 (1), 14516. https://doi.org/10.1038/ncomms14516.

Pyle, E., Zanetti, G., 2021. Current data processing strategies for cryo-electron tomography and subtomogram averaging. Biochem. J. 478 (10), 1827–1845. https://doi.org/10.1042/BCJ20200715.

Rego, N., Koes, D., 2015. 3Dmol.js: Molecular visualization with WebGL. Bioinformatics 31 (8), 1322–1324. https://doi.org/10.1093/bioinformatics/btu829.

Salfer, M., Collado, J.F., Baumeister, W., FernándezBusnadiego, R., Martínez-Sánchez, A., 2020. Reliable estimation of membrane curvature for cryo-electron tomography. PLoS Comput. Biol. 16 (8), 1–29. https://doi.org/10.1371/journal.pcbi.1007962.

Stalling, D., Westerhoff, M., Hege, H.C., 2005. Amira: A highly interactive system for visual data analysis. Vis. Handb. https://doi.org/10.1016/B978-012387582-2/50040-X.

Tasel, S.F., Mumcuoglu, E.U., Hassanpour, R.Z., Perkins, G., 2016. A validated active contour method driven by parabolic arc model for detection and segmentation of mitochondria. J. Struct. Biol. 194 (3), 253–271. https://doi.org/10.1016/j.jsb.2016.03.002.

Tegunov, D., Xue, L., Dienemann, C., Cramer, P., Mahamid, J., 2021. Multi-particle cryo-EM refinement with M visualizes ribosome-antibiotic complex at 3.5 Å in cells. Nat. Methods 18 (2), 186–193. https://doi.org/10.1038/s41592-020-01054-7.

Tong, Wai-Shun, Tang, Chi-Keung, Mordohai, P., Medioni, G., 2004. First order augmentation to tensor voting for boundary inference and multiscale analysis in 3d. IEEE Trans. Pattern Anal. Mach. Intell. 26 (5), 594–611. https://doi.org/10.1109/TPAMI.2004.1273934.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., Vijaykumar, A., Bardelli, A.P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C.N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D.A., Hagen, D.R., Pasechnik, D.V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G.A., Ingold, G.-L., Allen, G.E., Lee, G.R., Audren, H., Probst, I., Dietrich, J.P., Silterra, J., Webber, J.T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J.L., de Miranda Cardoso, J.V., Reimer, J., Harrington, J., Rodríguez, J.L.C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N.J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P.A., Lee, P., McGibbon, R.T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T.J., Robitaille, T.P., Spura, T., Jones, T.R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y.O., Vázquez-Baeza, Y., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods 17 (3), 261–272. https://doi.org/10.1038/s41592-019-0686-2 arXiv:1907.10121.

Volkmann, N., 2010. Methods for Segmentation and Interpretation of Electron Tomographic Reconstructions. In: Methods Enzymol, Vol. 483. Elsevier Inc., pp. 31–46. https://doi.org/10.1016/S0076-6879(10)83002-2, 1st Edition.

Wietrzynski, W., Schaffer, M., Tegunov, D., Albert, S., Kanazawa, A., Plitzko, J.M., Baumeister, W., Engel, B.D., 2020. Charting the native architecture of chlamydomonas thylakoid membranes with single-molecule precision. Elife. https://doi.org/10.7554/eLife.53740.

Wilfling, F., Lee, C.-W., Erdmann, P.S., Zheng, Y., Sherpa, D., Jentsch, S., Pfander, B., Schulman, B.A., Baumeister, W., 2020. A selective autophagy pathway for phase-separated endocytic protein deposits. Mol. Cell 80 (5), 764–778.e7. https://doi.org/10.1016/j.molcel.2020.10.030.

Wu, X., Zeng, X., Zhu, Z., Gao, X., Xu, M., 2019. Template-Based and Template-Free Approaches in Cellular Cryo-Electron Tomography Structural Pattern Mining. Comput. Biol., Codon Publications 175–186. https://doi.org/10.15586/computationalbiology.2019.ch11.

Xue, L., Lenz, S., Zimmermann-Kogadeeva, M., Tegunov, D., Cramer, P., Bork, P., Rappsilber, J., Mahamid, J., 2022. Visualizing translation dynamics at atomic detail inside a bacterial cell. Nature 610 (7930), 205–211. https://doi.org/10.1038/s41586-022-05255-2.

Zabeo, D., Davies, K.M., 2022. Studying membrane modulation mechanisms by electron cryo-tomography. Curr. Opin. Struct. Biol. 77, 102464. https://doi.org/10.1016/j.sbi.2022.102464.

Zhou, Q.-Y., Park, J., Koltun, V., 2018. Open3D: A Modern Library for 3D Data Processing. arXiv:1801.09847. https://arxiv.org/abs/1801.09847.