Biological Imaging (2023), 3: e10
doi:10.1017/S2633903X23000107

SOFTWARE REPORT

Ot2Rec: A semi-automatic, extensible, multi-software tomographic reconstruction workflow

Neville B.-Y. Yee[1], Elaine M. L. Ho[1], Win Tun[2,3], Jake L. R. Smith[4,5], Maud Dumoux[4], Michael Grange[4,5], Michele C. Darrow[1,6] and Mark Basham[1,3]

[1]Artificial Intelligence and Informatics, Rosalind Franklin Institute, Didcot, United Kingdom
[2]Faculty of Medical Sciences, Newcastle University, Newcastle upon Tyne, United Kingdom
[3]Diamond Light Source Ltd., Didcot, United Kingdom
[4]Structural Biology, Rosalind Franklin Institute, Didcot, United Kingdom
[5]Division of Structural Biology, Wellcome Centre for Human Genetics, University of Oxford, Oxford, United Kingdom
[6]SPT Labtech, Melbourn, United Kingdom
Corresponding author: Neville B.-Y. Yee; Email: neville.yee@rfi.ac.uk

Received: 25 July 2022; **Revised:** 10 February 2023; **Accepted:** 07 March 2023

Keywords: Cryo-electron microscopy; cryo-electron tomography; image processing

Abstract

Electron cryo-tomography is an imaging technique for probing 3D structures with at the nanometer scale. This technique has been used extensively in the biomedical field to study the complex structures of proteins and other macromolecules. With the advancement in technology, microscopes are currently capable of producing images amounting to terabytes of data per day, posing great challenges for scientists as the speed of processing of the images cannot keep up with the ever-higher throughput of the microscopes. Therefore, automation is an essential and natural pathway on which image processing—from individual micrographs to full tomograms—is developing. In this paper, we present Ot2Rec, an open-source pipelining tool which aims to enable scientists to build their own processing workflows in a flexible and automatic manner. The basic building blocks of Ot2Rec are plugins which follow a unified application programming interface structure, making it simple for scientists to contribute to Ot2Rec by adding features which are not already available. In this paper, we also present three case studies of image processing using Ot2Rec, through which we demonstrate the speedup of using a semi-automatic workflow over a manual one, the possibility of writing and using custom (prototype) plugins, and the flexibility of Ot2Rec which enables the mix-and-match of plugins. We also demonstrate, in the Supplementary Material, a built-in reporting feature in Ot2Rec which aggregates the metadata from all process being run, and output them in the Jupyter Notebook and/or HTML formats for quick review of image processing quality. Ot2Rec can be found at https://github.com/rosalindfranklininstitute/ot2rec.

Impact Statement

The field of cryo-electron tomography has grown substantially in recent years, bringing about new advances in hardware and software which enable visualization of cell and tissue architecture and proteins found in their native context. These same advances have, in some ways, stratified the field into those with access and those without. On the software side, this has emphasized the need for open-source options that do not require high levels of computational literacy to access. Additionally, it has highlighted the need for ways to both mix-and-match software for easy prototyping and comparisons between parameters and methods. Ot2Rec addresses these needs through a simple, unified plugin structure allowing the addition of existing software or the development of new and does so in a way which democratizes access.

1. Introduction

1.1. Electron cryo-tomography and data acquisition

Electron cryo-tomography (cryo-ET), is an imaging technique that produces three-dimensional outputs in the nanometer resolution range(1,2). In the biomedical field, it has been used to study heterogeneous purified proteins using subtomogram averaging (STA)(3,4), structures within thin or smaller cells(5), the edges of larger cells(6), and more recently in conjunction with cryo focused ion beam milling (cryoFIB), subcellular structures and proteins(7–9).

The sample must first be vitrified using either traditional plunge freezing methods(10) or high-pressure freezing for larger samples(11). In some cases, the frozen sample is subsequently thinned by cryosectioning(12) or cryoFIB milling(13–15). These cryogenic specimen preparation techniques alleviate the need for chemical fixatives and stains which are commonly used to preserve biological structures and bolster contrast in room temperature volume electron microscopy (vEM) techniques(16,17).

During data collection images are taken at a series of angles by tilting the specimen, typically in the range of approximately −60° to +60° with a 2° to 5° step between tilts, though many tilt acquisition schemes exist to ration exposure of the specimen(18). In all cases, a portion of information is not collected due to mechanical constraints of the microscope stage and increasing specimen thickness during tilting. This leads to a missing wedge or cone of information in Fourier space and an elongation of the resultant reconstructed data in the direction of the electron beam in real space(19).

1.2. Current image processing workflow for cryo-ET

Several stages of image processing tasks are required to produce tomograms from raw micrograph movies (Figure 1). Many software tools exist for each stage, which have been reviewed recently by Pyle and Zanetti(20). Raw micrograph movies at each tilt angle are first motion-corrected to compensate for specimen movement during acquisition. Individual frames of the movies recorded at each angle are aligned and averaged to produce a single image for each tilt angle. These motion-corrected images are sorted in order of their tilt angles to produce a tilt series. Tilt series projections are then aligned to ensure the rotation axis is consistent between projections. This can be performed with or without fiducial markers. At this stage, the contrast transfer function (CTF) can be optionally estimated from the aligned tilt series, and deconvolved with the tomogram at later stages to improve the image quality. Aligned projections are then reconstructed into tomograms, which provide the 3D representation of the specimen. Post-processing strategies can then be applied to the tomograms, which will be dependent on the specific research question. This often involves STA to obtain high-resolution structures of repeating structures in the tomogram, or segmentation to determine spatial relationships between structures.

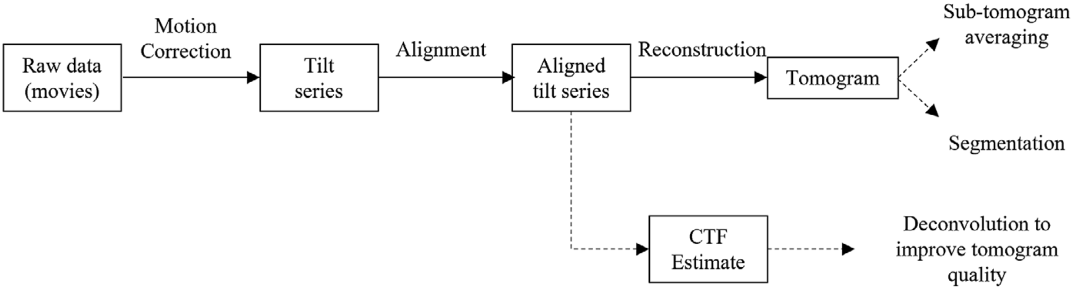

Figure 1. *Typical image processing tasks for reconstructing tomograms from raw micrograph movies. Key: boxes: datasets; solid arrows: processes necessary for tomogram reconstruction; dotted arrows: optional processes.*

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

1.3. Existing cryo-ET pipelines

The image processing tasks required to reconstruct tomograms from the movies are generally consistent for different experiments, though the post-processing stages afterward often require more customization. However, many software tools exist for each stage of the reconstruction process, each with their own disparate data structures, standards, and user interfaces. Users must learn how to use each individual tool and adapt their data structures to fit the specific software package, increasing the barrier to trying alternative tools which may have better performance or functionality. Therefore, pipelines which manage the data flow and provide a unified interface for several software tools would enable easier combination and trialing of the different software tools to optimize reconstruction outcomes.

A few pipelines already exist to automate the reconstruction process within a single framework, to varying degrees of customization (Table 1). Generally, these pipelines were developed to automate reconstruction for datasets collected within a single research group, and thus, do not enable substitution of different tools developed elsewhere. These pipelines include TomoBear(21), TomoRobot(22), EMAN2(23), and Warp(26). Scipion gives users a wide range of software tools to choose in their framework and offers a user-friendly workflow builder in their graphical user interface (GUI). However, accessing the Scipion tools programmatically is not straightforward, so automation of the process is more challenging.

1.4. Motivation

Here, we present Ot2Rec, a pipeline for reconstruction of cryo-ET tilt series which allows users to combine different software tools within a single framework. Ot2Rec can be used via a GUI but can also be automated programmatically if required, as demonstrated through a Jupyter Notebook in the Supplementary Material(27). Ot2Rec is easily extensible through its plugin architecture and published open-source under the Apache v2.0 license. Ot2Rec is developed as an open-source project with a strong focus on user involvement. We welcome contributions from users and developers alike, our contributors guide is available on our GitHub Wiki. A written tutorial guiding users through reproducing one of our case studies is available in the Supplementary Material, in addition to a guide to writing Ot2Rec plugins for developers. These documents are also online on our wiki at https://github.com/rosalindfranklininstitute/ot2rec/wiki.

2. Development Philosophy and Features

Ot2Rec was developed to achieve three key characteristics of a tomography reconstruction pipeline, with the ultimate aim of fully automating tomography reconstruction to obtain high-quality tomograms. Firstly, the pipeline had to offer different routes to combine tools for each stage of tomogram reconstruction. The pipeline had to be user-friendly and accessible to users without programming experience. Finally, a mechanism to evaluate all tilt series at a glance was also required. These functions together provide the framework for which a completely automated tomogram reconstruction pipeline can be built, where the optimum combination of software tools and parameters can be automatically selected and applied to several tilt series at once, with minimal user intervention. The following sections describe the design choices that were made toward achieving this functionality in Ot2Rec.

2.1. Different routes to tomogram reconstruction

2.1.1. Plugin architecture

In Ot2Rec, the basic program infrastructure is designed such that each feature is a standalone plugin, which has minimal interaction with other plugins. Such design is beneficial for developers as a fault in one plugin does not directly cause other plugins to fail, making further development and maintenance easier. This plugin architecture also facilitates the extension of Ot2Rec to cover other image processing tools, whilst maintaining easy integration with existing workflows.

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

Table 1. *Comparison of cryo-ET reconstruction pipelines.*

Software/Feature	TomoBear	TomoRobot	EMAN2	ScipionTomo	Warp	Ot2Rec
Open-source?	Yes	Yes	Yes	Yes	Yes	Yes
Programming language	Matlab	Python, Matlab	Python	Python	C#, C++, CUDA C	Python
File formats	EER, MRC, Tiff	MRC, Tiff	EER, MRC, Tiff, proprietary file formats	Raw, MRC,Tiff, jpeg, proprietary formats	MRC, Tiff, EM	EER, MRC, Tiff
GUI	No	No	Yes	Yes	Yes	Yes
Motion Correction	MotionCor2	MotionCor2	EMAN2			
MotionCor2, Xmipptomo, FlexAlign.	Warp	MotionCor2				
Alignment	Dynamo, IMOD	Modified Dynamo	EMAN2	IMOD	IMOD	AreTomo, IMOD
Reconstruction	IMOD	Relion	EMAN2	IMOD, Tomo3D, AreTomo, Nova-CTF	Warp	AreTomo, IMOD, Savu
CTF estimation	Gctf	CTFFind4	EMAN2	Cistem, CTFFIND4, Gctf, IMOD, Emantomo	Warp	CTFFind4, CTFSim
Dataset requirements	Unspecified	Fiducials required Dose-symmetric tilt scheme starting at 0	Unspecified	Unspecified	Unspecified	Unspecified
References	(21)	(22)	(23)	(24,25)	(26)	This work

Each plugin performs one task within the reconstruction process, for example, motion correction or alignment. The plugin has two main subroutines, one to capture user input and configure the parameters for the task to be performed, and the other which generates the commands to run the task on all tilt series. Each plugin follows a simple yet unified structure for its application programming interface (API): a Python class which encapsulates all the essential plugin-specific methods, followed by subroutines that enable the plugin to communicate with the Ot2Rec main API. These subroutines are called and executed directly as entry points by users. This well-defined, simple structure is helpful for developers and users alike as they become accustomed to the patterns of using Ot2Rec.

Plugins were chosen based on the software tools already used by users in our institute. The nine plugins currently available in Ot2Rec 0.2 are:

- Motion Correction
  - MotionCor2(28).
- CTF estimation and deconvolution
  - CTFFind4(29,30).
  - CTFSim (based on(30); described in Section 5.2).
  - RLDeconv (described in Section 5.2).
- Tilt series alignment
  - IMOD(31,32).
  - AreTomo(33).
- Reconstruction
  - IMOD(31,32).
  - AreTomo(33).
  - Savu(34).

2.1.2. Metadata handling

Metadata are small files used to record the locations and other useful information about actual data being processed. Metadata often plays a key role in a multi-step computational workflow as it defines how the data are linked to each other. Metadata generated from various software and processes are often incompatible (for instance, having different headers or different metadata file formats) creating obstructions in data flow. This is a currently known problem in the electron tomography community—there are many workflows and packages which use different and incompatible metadata formats, making interoperability between separate software packages extremely challenging.

In view of this, Ot2Rec has been designed with a central philosophy of unified metadata structure. To use Ot2Rec for image processing, the user first runs the command o2r.new with which Ot2Rec aggregates the project metadata that includes the path to the raw images (micrographs) and the tilt series indices of the individual micrographs.

In subsequent processing steps (i.e., plugins), Ot2Rec stores the metadata files from the individual programs and outputs an independent metadata file. This metadata file is a human-readable easily parsed YAML format that carries essential information for the next steps. Two YAML files are created, one contains the configuration information to set up the plugin, and the other contains information about the completed processes. For example, the configuration metadata file for an alignment would contain the process list (list of tilt series indices to process), path to the output folder, pixel size, rotation angle, bin factors, and patch sizes. Once the alignment is complete, the metadata file for the completed process would contain filepaths for the tilt series number, unaligned tilt series images, and aligned tilt series images. Subsequent steps can inherit properties taken from these metadata files, for example, the process list, pixel size, and paths to the aligned tilt series for a reconstruction plugin will be inherited from the previous alignment step.

The unified internal metadata system allows workflows to be built flexibly by interchanging the different plugins for each task. Whilst several default workflows are implemented in the package

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

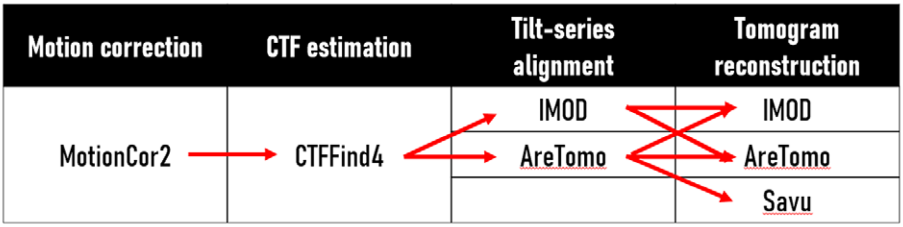

Figure 2. *Standard workflows implemented in Ot2Rec. Red arrows denote possible branches of the workflow, which currently covers all permutations of existing implemented plugins (though not necessarily all available options within a software, e.g., IMOD).*

(cf. Figure 2), for instance, MotionCor2 → IMOD (alignment + reconstruction), or MotionCor2 → IMOD (alignment) → Savu (reconstruction), the "network" of possible routes can continue to grow as more plugins are implemented into Ot2Rec. This flexibility and customization allow the user to maximize both result quality and computational performance.

As image processing programs each have different parameters, the quality of the output and computational performance can vary vastly depending on the chosen parameters. With a flexible workflow, users can choose the tools that best suit their purposes and test different parameters provided by the same tool. They can also compare the results using the reporting feature of Ot2Rec. Once the user has completed a comparative study, preferably using a small, representative subset of their data, they can continue to process the rest of their data using the optimal workflow and parameters they have identified.

2.2. User-friendliness

GUIs are a means for a program to communicate with the user. Its usage can range from the collection of essential parameters to the interactive display of results.

In Ot2Rec, a Linux-based program, since the building blocks are the individual plugins which have a wide range of parameters as inputs, a GUI is necessary to remove the use of command-line flags. For this reason, we have implemented MagicGUI(35) as Ot2Rec's GUI for gathering parameters from users. One of the biggest benefits of using a simple GUI over the command line is that parameters can be presented in a more human-readable and descriptive way rather than using internal variable names, which in many cases need to be brief. Another benefit of using a GUI as the first point of communication between the user and the program is that since the parameters are more descriptively presented, the chance of incorrect inputs can be reduced, as additional information can be displayed alongside the parameter inputs for explanation.

2.3. Report generation

Comparison of the outputs of different tomogram reconstruction workflows is facilitated in Ot2Rec by automatically generated reports, which contain a workflow diagram describing the plugins that have been applied, selected input parameters, and performance metrics for each plugin. This report is generated for the entire project, allowing the user to evaluate the reconstruction process for all tilt series in the project at a glance. These reports can be generated in a document, slideshow, or Jupyter notebook format(35), allowing users to customize how they would like to interact with the report. Reports for all the case studies in this work are included in the Supplementary Material as examples.

3. Case Studies

3.1. Case Study 1: Semi-automatic processing of large datasets

Tomographic image processing is often performed manually, with users processing one tilt-series at a time. Scripts can be written to automate some of the procedures especially for those being carried out

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

using command-line programs, such as MotionCor2, though this can be difficult and frustrating for many users without a programming background. Ot2Rec enables users to automatically set up and apply a reconstruction workflow to entire datasets of several tilt series at once, without any scripting or individual processing of datasets. A report can be automatically generated at the end for the user to evaluate the performance of the reconstruction process on all tilt series at a glance.

Here, we demonstrate the use of Ot2Rec to reconstruct tomograms from the EMPIAR-10364 dataset(36). This dataset consisted of 17 tilt series of *E. coli* minicells acquired on a Titan Krios at the electron Bio-Imaging Centre, Diamond Light Source. Each tilt series had 60 projections from $-60°$ to $+60°$ in 2° increments. The movie taken at each tilt angle contained five frames. Images were binned by a factor of 2 at the alignment step for faster processing, and no binning was applied in reconstruction for a final bin factor of $2\times$ between the raw data and tomogram. Motion correction was performed with MotionCor2, and alignment and weighted back-projection (WBP) reconstruction were performed with IMOD. See Section 5 for more details.

Ot2Rec substantially reduced the amount of user intervention required to process tomograms as configured workflows are automatically applied to all tilt series in the dataset. Typically, the user would have to take each individual tilt series through several software packages, which is feasible for experiments with a few tilt series, but quickly becomes a bottleneck when scaling to dozens or hundreds of tilt series. This case study shows that typical reconstruction workflows with popular software tools can be deployed with Ot2Rec for large datasets, with the potential for automated parameter tuning to determine the optimum reconstruction workflow in the future.

The report generated summarized the performance of each step in the reconstruction process. First, a diagram showing the processes run on the dataset is shown (Figure 3a). The shifts between frames reported by MotionCor2 were within 1–2 pixels and fairly uniform across all tilt series (Figure 3b). The average shifts between patches from the alignment patch-tracking step were between 10 and 38 Å, and all tilt series had relatively similar shifts (Figure 3c). The report also contains representative thumbnails of the central *x*–*y*, *x*–*z*, and *y*–*z* slices of every tomogram, allowing users to visually assess all tomograms at once. The full report for this case study is available in the Supplementary Material.

3.2. Case Study 2: Demonstration and testing of prototype features

Implementing custom plugins in Ot2Rec enables rapid development and deployment of bespoke algorithms and tools which can easily be integrated into existing tomogram reconstruction workflows. This case study demonstrates the results of a custom plugin called CTFSim, and how this bespoke tool was used in conjunction with other existing Ot2Rec plugins to enhance the quality of the reconstructed tomograms.

This case study investigated the use of deconvolution to improve contrast with the CTF estimated from CTFSim on tomograms of human choriocarcinoma cells (Figure 4). Images were acquired on a Titan Krios at the electron Bio-Imaging Centre, Diamond Light Source. Each tilt series had 41 projections taken at $-53°$ to $+27°$ in increments of 2°. Ot2Rec was used for motion correction, alignment, and reconstruction of the tomograms for all tilt series in the dataset. The tomograms were then deconvolved using the CTF estimated with the custom Ot2Rec plugin CTFSim (based on(30,37); see Section 5 for details).

The improvement in image quality was measured using the contrast-to-noise ratio (CNR)(38). Briefly, the signal was segmented from the background by thresholding the gray values with the automatically determined Otsu's threshold(39). The gray value distributions in the segmented signal and background regions were used to calculate the CNR.

The deconvolved tomograms showed improved contrast compared to the original, as shown visually and as measured by improved CNRs in the deconvolved images (Figure 4). CNR was higher in the deconvolved tilt series compared to nondeconvolved, with the largest increase in CNR seen in tilt series 12 with deconvolution increasing CNR by 65.6% over the original tomogram.

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

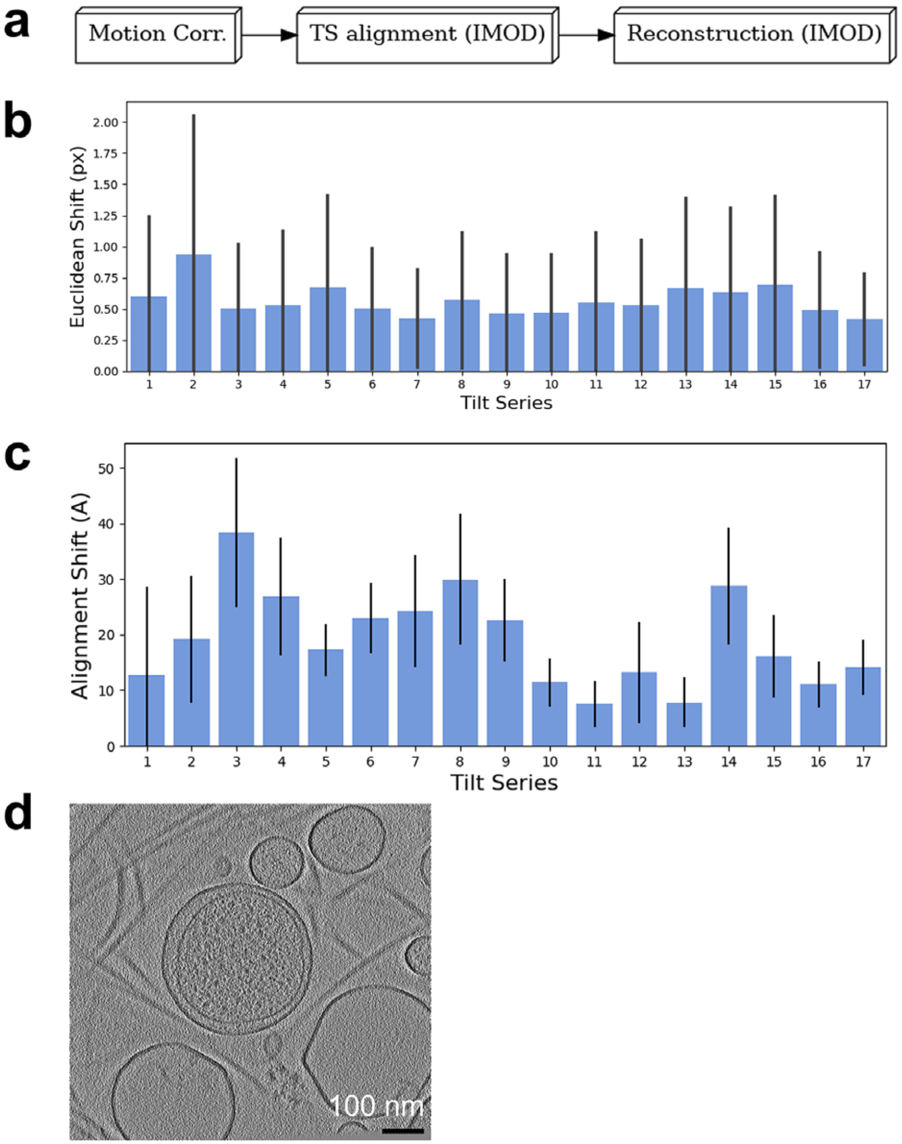

***Figure 3.** All tilt series in EMPIAR-10364 could be reconstructed at once in Ot2Rec and evaluated at a glance with the automatically generated report, some sections of which are highlighted below. (a) Workflow diagram for processes performed in Case Study 1. (b) Means and standard deviations of motion correction shifts between movie frames for all tilt series. (c) Means and standard deviations of alignment shifts between patches (L_2-norm) for all tilt series. (d) Central* x–y *slice of tomogram from tilt series 18 in EMPIAR-10364, with a Gaussian blur filter applied ($\sigma = 2.0\,px \approx 8.972\,\mathring{A}$). The full report is included in the Supplementary Material.*

3.3. Case Study 3: Optimisation of tomogram reconstruction workflow

Reconstruction of tomograms is a multi-stage process, where the final tomogram quality is affected by the results of earlier stages. Therefore, optimization of the entire workflow at each stage is necessary to obtain the best results. Ot2Rec provides the framework for this optimization by enabling users to mix-and-match different software tools on the same tilt series, and compare these results in a single report.

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

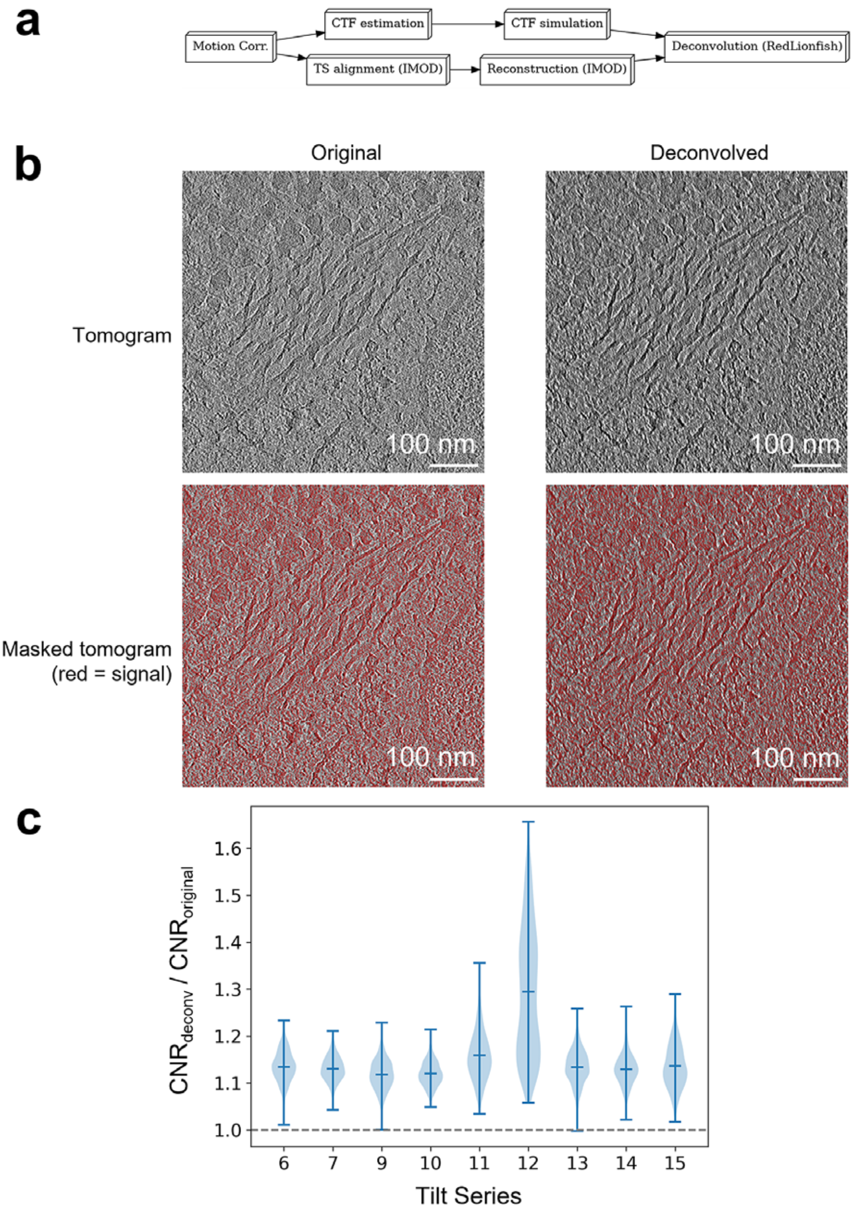

Figure 4. *PSF simulation and deconvolution implemented as custom plugins in Ot2Rec can be easily integrated into existing image processing workflows. Deconvolution improved image quality of all tomograms in this case study of human choriocarcinoma cells. (a) Workflow diagram for processes performed in Case Study 2. (b) Thumbnails showing an example original and deconvolved tomogram (tilt series 9), unmasked (top row), and masked (bottom row). The masked regions were used to calculate the CNR. (c) Violin plot showing a general boost (> 1) in contrast-to-noise ratios after deconvolution.*

In this case study, different combinations of alignment and reconstruction tools were applied to the same dataset, and Ot2Rec reports were generated to compare the performance of each (Figure 5). This case study consisted of 23 tilt series composed of 41 projection images each ranging from −60° to +60°. After motion-correction, fiducial-less alignment of the tilt series was performed with either IMOD or

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

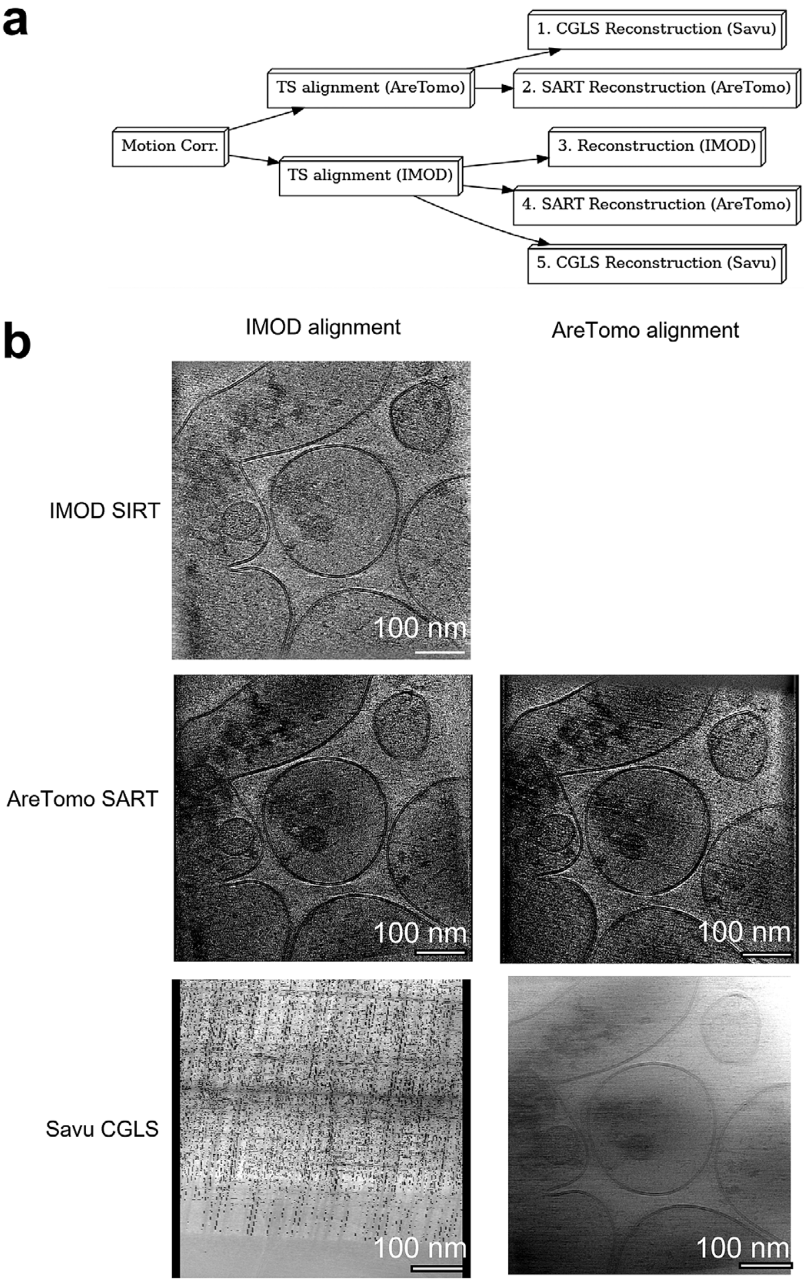

Figure 5. *Different combinations of IMOD, AreTomo, and Savu for alignment and reconstruction yielded varying results. Five combinations (1–5) were tested, as shown in the workflow diagram a. NB. Iterative reconstruction with CGLS in Savu (Route 5) failed due to poor alignment results from IMOD alignment. However, the same reconstruction method produced a good quality reconstruction when the AreTomo alignment method was used instead (Route 1): (a) Workflow diagram used in Case Study 3, (b) Central* x–y *slice of the tomogram reconstructed with IMOD, AreTomo, and Savu on IMOD and AreTomo aligned data. Note that IMOD reconstruction with AreTomo aligned data is not available on the current version of Ot2Rec, but will be supported in later versions. A movie showing this figure in 3D is available in the Supplementary Movie S1.*

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

AreTomo, followed by reconstruction with either IMOD simultaneous iterative reconstruction technique (SIRT), AreTomo simultaneous algebraic reconstruction technique (SART), or Savu conjugate gradient least squares (CGLS) reconstruction. Different combinations of alignment and reconstruction methods were expected to produce different results, which will be investigated in this section. The full reports for this case study are available in the Supplementary Material.

The shifts reported at the motion correction stage were found to be much larger for tilt series 13 onwards (Figure 6a), indicating a potential issue with tilt series acquisition from this and future datasets in the experiment. Shifts between patches in alignment were also larger overall for tilt series 13 onwards compared to the earlier datasets (Figure 6b). Inspection of the tilt series data showed obstructed views at high tilt angles for those affected by large motion correction shifts. In the future, Ot2Rec could include metrics to determine low-quality tilt angles to be excluded.

An example tomogram (tilt series 18) affected by the large shifts observed in motion correction was chosen to demonstrate differences in outcomes associated with each workflow (Figure 5b). Savu CGLS reconstruction failed with the IMOD-aligned data, but not with the AreTomo alignment, and the resultant tomogram had reasonable image quality. However, the CGLS reconstruction had lower contrast than the SIRT or SART reconstruction methods of IMOD or AreTomo. Reconstruction with AreTomo's SART was successful for both alignment methods (IMOD and AreTomo) with very few visual differences observed between the two outcomes. The IMOD SIRT reconstruction using IMOD alignments also produces a reasonable output which is visually similar to the AreTomo SART reconstructions created from either IMOD or AreTomo alignments. IMOD reconstruction using AreTomo alignments is not currently supported in Ot2Rec, but will be in the future.

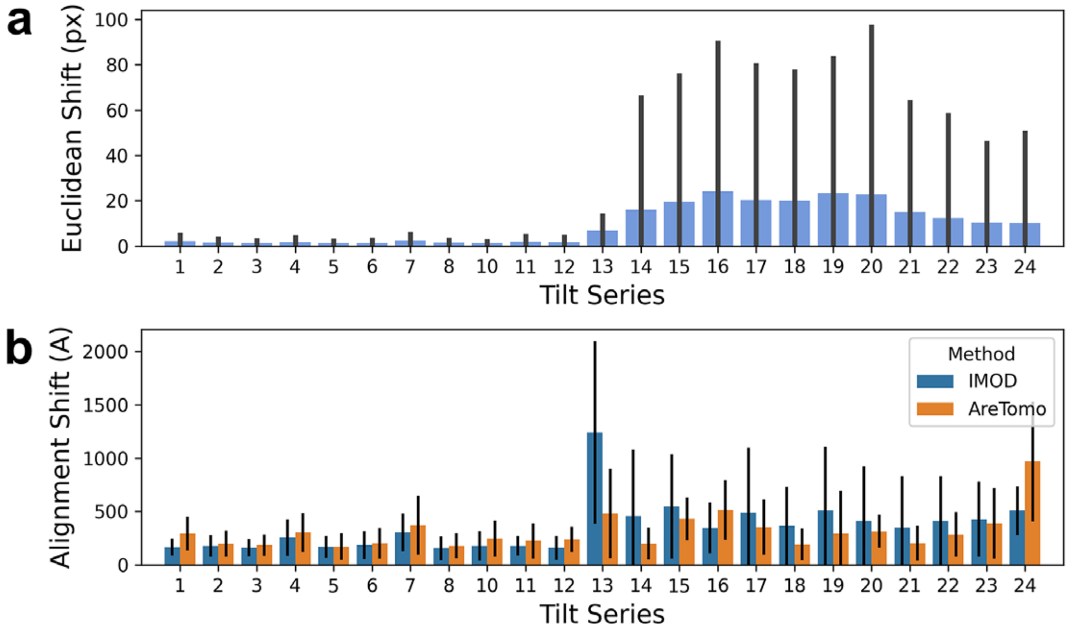

Figure 6. *Evaluation of all tilt series shifts from motion correction and alignment at once show substantially larger shifts from tilt series 13 onwards. (a) Euclidean shifts reported by motioncor2 for all tilt series in Case Study 3 (Section 3.3) show a large increase from tilt series 13 onwards. (b) Alignment shifts reported by IMOD and AreTomo alignment processes. The alignment shift here is the Euclidean distance between patches which are recorded as metadata from IMOD and AreTomo directly. In both (a) and (b), The bar plots represent the mean and the error bars are the standard deviations in shifts.*

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

4. Discussion

Ot2Rec enables reconstruction of tomograms from raw micrograph movies using a range of software tools, all within a single framework. This pipeline is distinct from available alternatives as users can choose different combinations of software tools to suit their specific experiment and automatically apply their workflow to several tilt series at once. Metadata is recorded at each stage and is easily retrievable from text files, and the performance of the overall reconstruction workflow is summarized in a human-readable, automatically generated report.

4.1. Future work

The case studies in this work have demonstrated that there are differences in the resultant tomograms from different reconstruction workflows. The reports generated by Ot2Rec also incorporate performance statistics from the individual plugins, for example, shifts between patches in fiducial-less IMOD tilt series alignment. However, independently measured quality metrics for each stage of the workflow would enable a more equitable comparison of different software tools, so the user (and eventually an algorithm) can quantitatively choose the best combination of software tools and the optimum input parameters for each. The next steps for improving the Ot2Rec reporting system will include development and implementation of such metrics. For example, Fourier Ring Correlation between reconstructions of even and odd projections is a commonly adopted, though computationally intensive, measure of resolution in final tomograms, which could be used to compare reconstruction tools if they are applied on the same aligned tilt series(40). In the future, tomograms could be reconstructed with a pre-defined range of software tools and selected input parameters, and the best overall workflow could be determined quantitatively and then applied to other tilt series in the experiment.

In the case study from Section 3.3, the difference in motion correction shifts between the first 12 and the remaining tilt series was easily observed from the plot of shifts for all tilt series in the project. Observing performance of the tomogram reconstruction process over time can be used as an indication of the quality of the upstream processes (sample preparation, data collection, microscope health)—enabling facility managers to capture valuable data to correct issues at these stages efficiently. Performance in specific metrics could help troubleshoot potential issues, for example, consistent large deviations between calculated and input tilt angles from the alignment process could be a sign of issues with the microscope stage. Issues with poor-quality images at specific tilt angles could also be detected and these images excluded from subsequent analysis.

Case Study 2 has demonstrated that custom image processing tools developed within the Ot2Rec framework can be integrated easily into existing image processing workflows, which is beneficial to both developers and users. For developers, maintenance and deployment efforts are significantly reduced as Ot2Rec already provides implementations for file handling of all intermediate steps in the tomography process. For users, customized image processing tools can be accessed quickly and tested alongside their existing workflows, all within a familiar framework.

Ot2Rec will also be extended to perform other cryo-ET processing tasks, for example, denoising tomograms, 3D particle picking, or STA. Some of these tasks will be supported in Ot2Rec by developing new plugins for existing software tools, or if no suitable tools are available, bespoke plugins can be developed which would then be available in conjunction with all the standard plugins in Ot2Rec. More work is needed to adapt the Ot2Rec architecture to handle iterative tasks which combine steps, for example, simultaneous CTF correction and reconstruction as implemented in NovaCTF(41) by Obr *et al.*(42), or iterative alignment and reconstruction(43). And further work is needed to enable a more automated approach that combines multiple plugins into a workflow that can be launched with a single command.

4.2. Conclusion

Ot2Rec currently includes the canonical initial steps of tilt series reconstruction, necessary for either STA or segmentation. Alongside of this, it also contains prototype plugins allowing for deconvolution of

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

tomograms to enhance contrast. Ot2Rec acts both as an open-source wrapper with a simple unified plugin structure and a developer and user-friendly prototyping tool for new plugins. The reporting functionality provides easy-to-access information, enabling comparisons and optimization of parameters and software packages. The further development of this pipeline opens the door to options such as automated data processing parameter tuning based on the purpose of data collection or automatic monitoring of upstream steps such as microscope health. Continued open-source cross-software development of the cryoET pipeline with a focus on ease-of-interaction for both the user and developer is the future of cryoET data processing.

5. Methods

5.1. Sample preparation and image acquisition

5.1.1. Case Study 1

Details of the sample preparation and image acquisition protocols are available in the original publication for the EMPIAR dataset(36).

For this study, motion correction of the micrographs was performed using MotionCor2, followed by a fiducial-less (patch tracking based) tilt series alignment using IMOD. The patch-tracking algorithm was configured such that there were 24 × 24 patches, each patch with the dimensions of 210 × 203 pixels, allowing a 15% overlap between patches. The aligned stacks were binned by a factor of 2 before tomogram reconstruction, which was also performed using the batchruntomo tool in the IMOD suite using the default WBP algorithm. The binned thickness of the volume was set to 1,920 pixels (≈861 nm), such that after the overall factor-2 binning, the dimensions of the output tomograms are 1,920 × 1,856 × 1,920 pixels, with the pixel spacing of 4.486 Å in all dimensions. Specific parameters used in Ot2Rec for this case study are in Supplementary Table S1.

In all case studies presented in this paper, processing was performed on a virtual machine with 12 Intel Xeon Gold 5218 2.30 GHz CPUs and 1 Tesla V100 GP.

5.1.2. Case Study 2

JEG-3 human choriocarcinoma cell line was purchased from Merck Life Science (UK) and cultured in Dulbecco’s modified Eagle’s media (DMEM/F-12, HEPES, Gibco, 11330057) supplemented with 10% fetal bovine serum (Gibco, 10500) and 1% Penicillin–streptomycin (10,000 U/ml, Gibco, 11548876). Cells were cultured in T75 flasks and incubated at 37°C with 5% CO_2.

A total of 200 mesh gold holey carbon grids (either R2/2 Quantifoil or R3.5/1 Quantifoil, Agar scientific) were sterilized by dipping into 100% ethanol for 2–5 min and rinsing with phosphate-buffered saline (PBS) (Gibco, 10010). The grids were coated with 30 μl of 1:20 fibronectin (Merck, F0895) in PBS, and kept in the incubator for 2 hr. The grids were then rinsed with PBS two times and put in a petri dish containing JEG-3 media, and incubated at 37°C and 5% CO_2 for overnight.

Prior to cell seeding, the grids were transferred into a 6-well plate (one grid per well) containing fresh JEG-3 media. A total number of trypsinized JEG-3 (3×10^4 cells) were seeded directly onto each grid. The cells were then allowed to settle in the incubator for 2–3 days until they were well spread on the grids.

For vitrification, the GP2 (Leica) was used. The humidity-controlled chamber was set to reach >80% humidity. After the grid containing the cells was placed into the GP2, a 2.5 μl JEG-3 medium droplet was applied onto the EM grid, and the grid was blotted from the reverse side for 6–10 s. The grid was plunged into liquid ethane, and the frozen grids were stored until focused ion beam (FIB) milling(44) was performed.

Prior to the FIB milling, the frozen EM grids were clipped into clip rings (ThermoFisher). Scios scanning electron microscope (ThermoFisher) at the electron Bio-Imaging Centre at Diamond Light Source was used to perform the FIB milling. Gallium ion beams were used to mill the cellular samples to a lamella thickness of 200 nm. The ion beam parameters were set to 30 kV beam energy, and 50 pA–0.3 nA beam current (for coarse milling) and 30pA (for fine milling). The grids with FIB-milled lamellae were

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

stored again until cryoET was undertaken using a Titan Krios (ThermoFisher Scientific) at 300 kV at eBIC. Each tilt series had 41 projections taken at −53° to +27° in increments of 2° using a dose-symmetric imaging scheme[18]. Data were collected using a Falcon 4i SelectrisX at 64,000× magnification (1.97 Å/pixel).

Motion correction of the micrographs was first performed with MotionCor2, followed by the estimation of the true defocus values using CTFFind4 on a *per-micrograph* basis. The CTFFind4 outputs were then processed using the CTFSim plugin, an Ot2Rec prototype feature, which also simulated and internally reconstructed the 3D PSF of the tilt series. The PSF volumes were truncated to the central $30 \times 30 \times 30$ pixels as the PSF is fast-declining by nature.

Fiducial-less tilt series alignment was performed on the motion-corrected micrographs using IMOD. The patch-tracking algorithm in IMOD was configured such that there were 24×24 patches, each patch with the dimensions of 224 pixels squared, allowing a 15% overlap between patches. The aligned stacks were binned by a factor of 8 before tomogram reconstruction. Reconstruction was also performed using the batchruntomo tool in the IMOD suite, and the default WBP was used with the unbinned tomogram thickness set to 4,096 pixels (matching with the unbinned micrograph dimensions of $4,096 \times 4,096$). The final tomograms have a thickness of 765.9 nm, which translates to overall dimensions of $512 \times 512 \times 512$ pixels, given an overall binning factor of 8.

Lastly, the reconstructed tomograms were deconvolved using the RLDeconv tool in Ot2Rec, with the corresponding simulated PSF volumes as deconvolution kernels. Specific parameters used in Ot2Rec for this case study are in Supplementary Table S2.

The CNR, as defined in equation (1)[38], was used to compare the tomograms before and after deconvolution with CTFSim and RLDeconv.

$$\mathrm{CNR} = \frac{|\mu_1 - \mu_2|}{\sqrt{{\sigma_1}^2 + {\sigma_2}^2}}, \tag{1}$$

where $\mu_1, \mu_2, \sigma_1, \sigma_2$ are the average gray values of the signal, that of the background, the standard deviation of signal gray values, and that of the background gray values, respectively.

The CNR was calculated from the central 90% of the original and deconvolved volumes in the z-direction. Then for each z-slice in a volume, the Ōtsu threshold[39] was calculated and used for separating[1] the signal from the background (cf. masks shown in red in the second row of Fig 4b). The CNRs of the i th z-slices of the two volumes were calculated using equation (1). The ratio between the CNR of the deconvolved volume and that of the original tomogram was calculated to gauge the extent of enhancement in image contrast. A value over 1 signifies a positive boost in image contrast after deconvolution.

5.1.3. Case Study 3

Primary cortical neurons of embryonic C57BL/6 J mice were dissociated and seeded on Quantifoil R 2/2 SiO_2 Au200 grids that were glow discharged using a GloCube Plus (Quorum) at 20 mA for 30s and treated with 0.1 mg/mL Poly-D-Lysine (Gibco) at 37°C overnight and rinsed with PBS and left to dry. Cells were seeded onto grids and maintained in Neurobasal Medium (Gibco) supplemented with B-27TM (Gibco), 1% penicillin/streptomycin (Gibco), 2 mM Glutamate (Gibco). 20% of the volume of old media was replaced with fresh media every 3–4 days. Cells were plunge frozen on DIV 14.

Data were collected with a Titan Krios G4 (ThermoFisher Scientific) transmission cryo-electron microscope at 300 kV equipped with a cold field emission source gun, a Selectris (ThermoFisher Scientific) electron imaging filter, and a post-imaging filter mounted Falcon 4 (ThermoFisher Scientific) direct electron detector. Tilt series were acquired at 81,000× magnification (pixel size 1.47 Å) in a dose symmetric tilt scheme[18] with a tilt range of −60° to 60° imaging at 3° increments, with $2e^-/Å$ dose per

[1] NB. This is only a rudimentary measures to pick out potential signals, and is by no means an attempt to automatically segment useful features from the slice.

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

tilt image. Tilt series were acquired using Tomography software (ThermoFisher Scientific) and collected as EER files(45).

Motion correction was first performed with MotionCor2(28). Next, fiducial-less tilt series alignment was performed with IMOD(31,32) and AreTomo 1.1.0(33). In both cases, a binning factor of 8 was used. The IMOD aligned data were reconstructed with three methods: IMOD SIRT, AreTomo SART, and Savu CGLS reconstruction via the ASTRA reconstruction toolbox(46–48). The AreTomo aligned tilt series was reconstructed with AreTomo SART and Savu CGLS, though IMOD reconstruction will be supported in later versions of Ot2Rec. Specific parameters used in Ot2Rec for this case study are in Supplementary Table S3.

5.2. Prototype features

5.2.1. CTFSim

As mentioned previously, the program CTFFind4 is a tool used for estimating the *true* defocus value for each micrograph collected. However, whilst it produces parameters for determining the shape of the CTF associated with the microscope settings, it does not directly have the CTF as a standard output. In view of this, we developed an in-house plugin, named CTFSim, to simulate the CTF of the micrograph by reverse-engineering the equations described in the Rohou *et al.* paper(30,49).

$$\mathrm{CTF}(\lambda,\mathbf{g},\Delta f,C_s,\Delta\varphi,w_2) = -\sin\left(\chi(\lambda,\mathbf{g},\Delta f,C_s,\Delta\varphi,w_2)\right) \tag{2}$$

with

$$\chi(\lambda,\mathbf{g},\Delta f,C_s,\Delta\varphi,w_2) = \pi\lambda|\mathbf{g}|^2\left(\Delta f-\frac{1}{2}\lambda^2|\mathbf{g}|^2C_s\right)+\Delta\varphi+\arctan\left(\frac{w_2}{\sqrt{1-w_2{}^2}}\right), \tag{3}$$

where λ is the associated electron wavelength, $\mathbf{g}$ is the spatial frequency vector, Δf is the defocus value evaluated from CTFFind4, C_s is the spherical aberration, $\Delta\varphi$ is the phase shift, and the value w_2 is associated with the relative phase contrast w_1 via the relation $w_1 = \sqrt{1-w_2{}^2}$.

By definition, the CTF is a function defined in the complex reciprocal space, and is hence rather difficult to visualize and use "as is." Therefore, an additional feature of the CTFSim is to convert the simulated two-dimensional CTF to the point spread function (PSF) in the real space via a direct inverse Fourier transform. Lastly, since the CTF (and PSF) are calculated using the size of the motion-corrected micrographs, these images would have the same size of those micrographs. However, due to the quick-decaying feature of the PSF, the user can choose to truncate the resultant PSF image to a certain size (e.g., 30 pixels squared, by default) in order to facilitate further usage of the obtained results.

Once the per-micrograph 2D PSF profiles for a tilt-series are simulated, CTFSim reconstructs the profiles internally using the WBP algorithm into a tomogram of the PSF.

As suggested in Croxford *et al.*(37), the *cropped* 3D PSF is post-processed in a three-step process. The PSF tomogram is firstly converted into the 3D CTF with a Fourier transform. Then the pixels are "normalized"[2] through a division by the value at zero-frequency (i.e., $|\mathbf{g}| = 0$). Lastly, the normalized CTF stack is converted back to the PSF in real-space via an inverse Fourier transform.

Finally, the processed PSF is then normalized by dividing the array elements by the global maximum of the array. Although it can be derived trivially from equation (5) that the analytical output of the iterations should be independent of a linear scaling of the PSF, numerically speaking it is still a multi-step process and an unnormalized PSF could potentially cause numerical instabilities especially in the first fraction in equation (5). Therefore we assert that if the PSF is to be used to deconvolve the raw tomogram via the Richardson–Lucy scheme, this extra operation is necessary.

[2] We acknowledge that this use of "normalization," directly quoted from the source, may be an abuse of terminology, as the normalization of a complex matrix is usually defined as the division by the norm of the matrix of associated moduli, rather than by the value at $|\mathbf{g}| = 0$.

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

5.2.2. RLFDeconv

The blurring of images due to the optical or electronic properties of the PSF has long been a concern for microscopists. In transmission electron microscopy, the obtained image can be modeled, in the simplest form, as[50]:

$$I_0 = I_{\mathrm{perf}} \otimes \mathrm{PSF} + \mathcal{N}, \tag{4}$$

where I_0 is the final output image, I_{perf} is the perfect image, $\mathcal{N}$ is a stochastic noise term obeying Poisson distribution, and $\otimes$ denotes the convolution operation. In order to reverse-engineer the perfect image I_{perf}, an inverse operation must be carried out. We note that equation (4) cannot be solved exactly with any analytical or numerical method, since the noise term is unknown and cannot be isolated from the output image. Hence to obtain a reasonable solution, an approximation is necessary. Here a common approximation is that the image signal-to-noise ratio is high enough such that the noise contribution can be ignored.

However, even with such approximation, the inverse of a convolution is still unsolvable by any analytical means, as the solution is nonunique. Therefore one needs to resort to an iterative approach numerically. In 1972 and 1974, Richardson and Lucy separately discovered an iterative scheme[51,52] that gives an approximation approaching an ideal solution as the number of iteration increases. Mathematically, the scheme reads

$$I_{i+1} = \left(\frac{I_0}{I_i \otimes \mathrm{PSF}} \otimes \overline{\mathrm{PSF}} \right) I_i, \tag{5}$$

$$\lim_{i \to \infty} I_i = I_{\mathrm{perf}}, \tag{6}$$

where I_i is the image in the i th step, and the over-bar on PSF denotes a flip operation of the PSF, hence $\overline{\mathrm{PSF}(\mathbf{x})} \equiv \mathrm{PSF}(-\mathbf{x})$.

In Ot2Rec, rather than developing an in-house solver, we have implemented a wrapper plugin for the Python library RedLionfish[53] for performing the 3-dimensional deconvolution between a reconstructed tomogram and a simulated PSF stack (similarly reconstructed as the experimental tomogram). To facilitate the deconvolution of larger stacks, which might cause GPU/CPU memory issues, we also implemented the 3D version of the so-called “block-iterative” algorithm, inspired by Lee[54], which breaks the volume (tomogram) into chunks, on each of which a normal Richardson–Lucy deconvolution is performed independently. The separately deconvolved chunks are then stitched back together. A merit of this method is that due to the massive reduction in the size of deconvolution operands, a much lower memory cost for the GPU/CPU can be achieved. However, the authors would like to emphasize that this block-iterative method must be used with care, as the padding setting in the convolution operations could introduce artifacts around the borders of the chunks as they are stitched back.

5.3. Basic usage guide

This short section is dedicated to explain the basics of user interaction with Ot2Rec.

5.3.1. Installation

Ot2Rec is an open-source Linux-based program. Its code repository can be accessed at https://github.com/rosalindfranklininstitute/Ot2Rec.

The easiest way to install Ot2Rec is to download the shell script of the latest release from the Release tab on GitHub and execute the script on the command line.

An alternative method of installation, should the user wish to build the software from source, is to clone the repository to the local environment. As Ot2Rec uses miniconda, it is recommended that the user first create a new conda virtual environment (“venv”) with Python 3.8 (or above) pre-installed. Once the venv

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

Table 2. *Ot2Rec commands and their descriptions.*

Ot2Rec Command	Description
o2r.new	Creates a new Ot2Rec project
o2r.<plugin>.<function>.new	Creates a new instance of each plugin. For example, o2r.imod.align.new starts a new alignment in IMOD and captures user inputs in the GUI for this plugin. The user parameters are stored in a yaml file
o2r.<plugin>.<function>.run <proj_name>	Runs the commands that perform the plugin's functions on the project where <proj_name> is the project name. For example, o2r.imod.align.run TS runs IMOD alignment on all tilt series in project TS, which has user parameters stored in TS_align.yaml created from o2r.imod.align.new
o2r.report.run <proj_name> –to_html –to_slides	Generate the report for the project <proj_name> as a html document and as slides. Note that this requires the o2r_report environment to be activated (detailed instructions available on GitHub)

is loaded, the user can then use the command pip install. in the cloned root folder to automatically install Ot2Rec and its dependencies locally.

5.4. Basic usage

Once the virtual environment has been activated, all Ot2Rec command-line functions are available in the terminal, which generates the corresponding GUI for each plugin. All Ot2Rec commands follow a specific format which has been summarized in Table 2.

To start an image processing pipeline with Ot2Rec, the first command to be used is o2r.new. With this, a GUI will be displayed, allowing the user to enter some essential project-dependent parameters, with which Ot2Rec will aggregate the master metadata for downstream operations.

For subsequent steps, all the plugins contain a new and a run function. The new commands, like the previous o2r.new, prompt the user to input essential parameters for the specified plugin, then collect and pre-propulate the configuration files with metadata from previous steps. The run commands look for the relevant configuration files and previous metadata records, and execute the specified plugin. For all run operations, the project name (as defined by the user at the beginning with the o2r.new command) is required as a command-line argument, as it is used for seeking the correct configuration files.

Acknowledgments. We acknowledge Diamond Light Source for access and support of the cryo-EM facilities at the UK's national Electron Bio-Imaging Centre (eBIC), funded by the Wellcome Trust, MRC, and BBRSC. We also thank eBIC for ongoing collaboration and feedback on Ot2Rec and providing the initial scripts on which Ot2Rec was based. We also thank the Core Team at the Artificial Intelligence and Informatics Theme at the Rosalind Franklin Institute, especially Drs. Joss Whittle and Laura Shemilt, for their help in configuring the computing infrastructure for running and testing of Ot2Rec.

Competing interest. M.C.D. is an employee of SPT Labtech. The remaining authors declare that they have no competing interests.

Authorship contribution. Data acquisition: W.T., J.L.R.S., M.G.; Data processing: N.B.-Y.Y., E.M.L.H.; Software development: N.B.-Y.Y., E.M.L.H.; Writing—original draft: N.B.-Y.Y., E.M.L.H., W.T., M.D., M.C.D., M.B. All authors approved the final submitted draft.

Funding statement. The Rosalind Franklin Institute is funded by UK Research and Innovation through the Engineering and Physical Sciences Research Council. Funding was also provided by the Wellcome Trust through the Electrifying Life Science grants 220526/Z/20/Z.

Data availability statement. The GitHub repository of Ot2Rec is open-source and can be accessed at https://github.com/rosalindfranklininstitute/Ot2Rec. Ot2Rec Report is also open-source and is available at https://github.com/rosalindfranklininstitute/ot2rec_report. All data used in this manuscript can be found on EMPIAR (case study 1: EMPIAR-10364; case study 2: EMPIAR-XXXXX; case study 3: EMPIAR-XXXXX).

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

Supplementary material. The supplementary material for this article can be found at http://doi.org/10.1017/S2633903X23000107.

References

1. Schmid M (2011) Chapter 2: single-particle electron cryotomography. *Adv Protein Chem Struct Biol* **82**, 37–65.
2. Briggs JAG (2013) Structural biology *in situ* – the potential of subtomogram averaging. *Curr Opin Struct Biol* **23**(2), 261–267.
3. Förster F & Hegerl R (2007) Structure determination *in situ* by averaging of tomograms. *Methods Cell Biol* **79**, 741–767.
4. Schur FKM, *et al.* (2013) Determination of protein structure at 8.5 Å resolution using cryo-electron tomography and sub-tomogram averaging. *J Struct Bio* **184**(3), 394–400.
5. Dai W, *et al.* (2013) Visualizing virus assembly intermediates inside marine cyanobacteria. *Nature* **502**, 707–710.
6. Sutton G, *et al.* (2020) Assembly intermediates of orthoreovirus captured in the cell. *Nat Commun* **11**, 4445.
7. Wu GH, *et al.* (2020) Multi-scale 3D cryo-correlative microscopy for vitrified cells. *Structure* **28**(11), 1231–1237.
8. Schertel A, *et al.* (2013) Cryo FIB-SEM: volume imaging of cellular ultrastructure in native frozen specimens. *J Struct Bio* **184** (2), 355–360.
9. Zhu Y, *et al.* (2021) Serial cryoFIB/SEM reveals cytoarchitectural disruptions in leigh syndrome patient cells. *Structure* **29**(1), 82–87.
10. Dubochet J, *et al.* (1988) Cryo-electron microscopy of vitrified specimens. *Quart Rev Biophys* **21**(2), 192–228.
11. Dahl R & Staehelin A (1989) High-pressure freezing for the preservation of biological structure: theory and practice. *J Electron Microsc Tech* **13**(3), 165–174.
12. Al-Amoudi A, *et al.* (2004) Cryo-electron microscopy of vitreous sections of native biological cells and tissues. *J Struct Bio* **148**(1), 131–135.
13. Krueger R (1999) Dual-column (FIB–SEM) wafer applications. *Micron* **30**(3), 221–226.
14. Hayles MF, *et al.* (2010) The making of frozen-hydrated, vitreous lamellas from cells for cryo-electron microscopy. *J Struct Bio* **172**(2), 180–190.
15. Wolff G, *et al.* (2019) Mind the gap: micro-expansion joints drastically decrease the bending of FIB-milled cryo-lamellae. *J Struct Bio* **208**(3), 107389.
16. Peddie C & Collinson LM (2014) Exploring the third dimension: volume electron microscopy comes of age. *Micron* **61**, 9–19.
17. Peddie C, *et al.* (2022) Volume electron microscopy. *Nat Rev Methods Primers* **2**, 51.
18. Hagen W, *et al.* (2017) Implementation of a cryo-electron tomography tilt-scheme optimized for high resolution subtomogram averaging. *J Struct Bio* **197**(2), 191–198.
19. Koning RI, *et al.* (2018) Advances in cryo-electron tomography for biology and medicine. *Ann Anat* **217**, 82–96.
20. Pyle E & Zanetti G (2021) Current data processing strategies for cryo-electron tomography and subtomogram averaging. *Biochem J* **478**(10), 1827–1845.
21. Balyschew N, Kudryashev M, *et al.* (2022) TomoBEAR. https://github.com/KudryashevLab/TomoBEAR (accessed 15 November 2022).
22. Pyle E & Zanetti G (2022) Automated workflow to pre-process raw tomography data for RELION 4.0. https://github.com/EuanPyle/relion4_tomo_robot.
23. Tang G, Peng L, *et al.* (2007) EMAN2: an extensible image processing suite for electron microscopy. *J Struct Bio* **157**, 38–46.
24. de la Rosa-Trevín JM, Quintana A, *et al.* (2016) Scipion: a software framework toward integration, reproducibility and validation in 3D electron microscopy. *J Struct Bio* **195**(1), 93–99.
25. de la Morena JJ, *et al.* (2022) ScipionTomo: towards cryo-electron tomography software integration, reporducibility, and validation. *J Struct Bio* **214**(3), 107872.
26. Tegunov D & Cramer P (2019) Real-time cryo-electron microscopy data preprocessing with Warp. *Nat Methods* **16**, 1146–1152.
27. Kluyver T, *et al.* (2016) Jupyter Notebooks – a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87–90. Amsterdam: IOS Press.
28. Zheng S, *et al.* (2017) MotionCor2: anisotropic correction of beam-induced motion for improved cryo-electron microscopy. *Nat Methods* **14**, 331–332.
29. Mindell JA & Grigorieff N (2003) Accurate determination of local defocus and specimen tilt in electron microscopy. *J Struct Bio* **142**(3), 334–347.
30. Rohou A & Grigorieff N (2015) CTFFIND4: Fast and accurate defocus estimation from electron micrographs. *J Struct Bio* **192** (2), 216–221.
31. Kremer J, *et al.* (1996) Computer visualization of three-dimensional image data using IMOD. *J Struct Bio* **116**(1), 71–76.
32. Mastronarde DN & Held SR (2017) Automated tilt series alignment and tomographic reconstruction in IMOD. *J Struct Bio* **197**(2), 102–113.
33. Zheng S, *et al.* (2022) AreTomo: an integrated software package for automated marker-free, motion-corrected cryo-electron tomographic alignment and reconstruction. *J Struct Bio X* **6**, 100068.
34. Wadeson N & Basham M (2016) Savu: A python-based, MPI framework for simultaneous processing of multiple, N-dimensional, large tomography datasets. arxiv: 1610.08015v1.

https://doi.org/10.1017/S2633903X23000107 Published online by Cambridge University Press

35. Lambert T, *et al.* (2022) napari/magicgui. Zenodo. https://doi.org/10.5281/zenodo.7254817
36. Burt A, *et al.* (2020) Complete structure of the chemosensory array core signalling unit in an *E. coli* minicell strain. *Nat Comms* **11**, 743.
37. Croxford M, *et al.* (2021) Entropy-regularized deconvolution of cellular cryotransmission electron tomograms. *PNAS* **118** (50), e2108738118.
38. Timischi F (2015) The contrast-to-noise ratio for image quality evaluation in scanning electron microscopy. *Scanning* **37**(1), 54–62.
39. Ōtsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Sys Man Cyber* **9**(1), 62–66.
40. Cardone G, Grünewald K, *et al.* (2005) A resolution criterion for electron tomography based on cross-validation. *J Struct Bio* **151**(2), 117–129.
41. Turoňová B, *et al.* (2017) Efficient 3D-CTF correction for cryo-electron tomography using NovaCTF improves subtomogram averaging resolution to 3.4 A. *J Struct Bio* **199**(3), 187–195.
42. Obr M, *et al.* (2022) Exploring high-resolution cryo-ET and subtomogram averaging capabilities of contemporary DEDs. *J Struct Bio* **214**(2), 107852.
43. Gürsoy D, *et al.* (2017) Rapid alignment of nanotomography data using joint iterative reconstruction and reprojection. *Sci Rep* **7**, 11818.
44. Marko M, *et al.* (2007) Focused-ion-beam thinning of frozen-hydrated biological specimens for cryo-electron microscopy. *Nat Methods* **4**, 215–217.
45. Guo H, *et al.* (2020) Electron-event representation data enable efficient cryoEM file storage with full preservation of spatial and temporal resolution. *IUCrJ* **7**(5), 860–869.
46. van Aarle W, *et al.* (2016) Fast and flexible X-ray tomography using the ASTRA toolbox. *Optics Express* **24**(22), 25129–25147.
47. van Aarle W, *et al.* (2015) The ASTRA Toolbox: a platform for advanced algorithm development in electron tomography. *Ultramicroscopy* **157**, 35–47.
48. Palenstijn WJ, *et al.* (2011) Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs). *J Struct Bio* **176**(2), 250–253.
49. Fernando KV & Fuller SD (2007) Determination of astigmatism in TEM images. *J Struct Bio* **157**, 189–200.
50. Reimer L & Kohl H. (2007) *Transmission Electron Microscopy: Physics of Image Formation*. New York: Springer.
51. Richarson WH (1972) Bayesian-based iterative method of image restoration. *JOSA* **62**(1), 55–59.
52. Lucy LB (1974) An iterative technique for the rectification of observed distributions. *Astron J* **79**(6), 745–754.
53. Perdigão LMA, *et al.* (2022) rosalindfranklininstitute/RedLionfish. Github. https://github.com/rosalindfranklininstitute/RedLionfish (accessed 15 November 2022).
54. Lee NY (2015) Block-iterative Richardson-Lucy methods for image deblurring. *EURASIP J Image Video Process* **2015**, 14.

Cite this article: Yee N. B. -Y, Ho E. M. L, Tun W, Smith J. L. R, Dumoux M, Grange M, Darrow M. C and Basham M (2023). Ot2Rec: A semi-automatic, extensible, multi-software tomographic reconstruction workflow. *Biological Imaging*, 3: e10. doi: https://doi.org/10.1017/S2633903X23000107