MicrographCleaner: a python package for cryo-EM micrograph cleaning using deep learning

Ruben Sanchez-Garcia[1], Joan Segura[2], David Maluenda[1], C.O.S. Sorzano[1], J.M. Carazo[1]

National Center of Biotechnology (CSIC)/Instruct Image Processing Center, C/ Darwin nº 3, Campus of Cantoblanco, 28049 Madrid, Spain.

[2] Research Collaboratory for Structural Bioinformatics Protein Data Bank, San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA 92093, USA

Abstract

Cryo-EM workflows require from tens of thousands of high-quality particle projections to unveil the three-dimensional structure of macromolecules. Current methods for automatic particle picking tend to suffer from high false-positive rates, hurdling the reconstruction process. One common cause of this problem is the presence of carbon and different types of high-contrast contaminations that, in many cases, affect large areas of micrographs. In order to overcome this limitation, we have developed MicrographCleaner, a deep learning approach designed to discriminate which regions of micrographs are suitable for particle picking and which are not, that we will refer to as "contaminated". MicrographCleaner implements a U-net-like model trained on a manually curated dataset compiled from over five hundred micrographs. The benchmarking, carried out on about one hundred independent micrographs, shows that MicrographCleaner is a very efficient approach for micrograph preprocessing.

Availability and implementation
MicrographCleaner package is available at PyPI and Anaconda Cloud repositories as micrograph_cleaner_em. Source code is available at https://github.com/rsanchezgarc/micrograph_cleaner_em. Integration with the cryo-EM software Scipion/Xmipp is also provided through the deepMicrographScreen protocol.

1. Introduction.

Cryogenic-Electron Microscopy (cryo-EM) Single Particle Analysis (SPA) has recently become a powerful technique for the determination of macromolecular structures achieving, in many cases, atomic resolutions. SPA consists of a set of complex and variable operations that, departing from thousands of particle projections, leads to the synthesis of electronic density maps of macromolecules. The massive number of particles that are needed for SPA has made of automatic particle picking one of the most influential steps in virtually all reconstruction workflows. Nevertheless, some problems intrinsic to the cryo-EM pipelines, such as low signal-to-noise ratio and the presence of high contrast artifacts and contaminants in the micrographs, degrades the performance of particle picking algorithms (Zhu *et al.*, 2004; Vargas *et al.*, 2013) and leads to the addition of false positive particles in SPA workflows. This problem can be mitigated trough different algorithms that clean and remove incorrectly selected particles after automatic picking (Sanchez-Garcia et al., 2018; Vargas et al., 2013).

One of the most common shortcomings observed during automatic picking is the attraction of these methods to select grid carbon spots, especially at the holes edges. Due to its relevance, some algorithms have been designed to prevent particle selection in those regions. For example, the em_hole_finder program, included in the Appion package (Lander et al., 2009) is based on morphological image processing operations to compute masks around carbon holes. Similarly, EMHP (Berndsen et al., 2017)

was designed to perform a similar task through image filtering and thresholding operations followed by a circle fitting procedure. Although very useful when grid edges are clearly visible, both approaches struggle in those cases where high contrast contaminations are present in micrographs. Moreover, both of them require human supervision to determine the presence of carbon in the micrographs and to set some user-defined parameters. As a result, its applicability is limited to supervised scenarios. More recently, the Warp package (Tegunov and Cramer, 2018) included a deep learning particle picker algorithm that was explicitly trained to detect and avoid carbon and contaminated regions using a pixel-wise classification -segmentation- approach.

Following this line, and with the aim of overcoming these limitations, we have developed MicrographCleaner, a fully automatic, easy-to-install and easy-to-use deep learning solution that performs a pixel-wise classification of micrographs into two categories, desirable and undesirable regions for picking. Likewise Warp particle picker, MicrographCleaner relies on one of the most extended network architectures (Ronneberger et al., 2015), but the different choices in important parameters result, in turn, in quite different levels of performance. Thus, according to our benchmarking, MicrographCleaner is not only able to provide a more robust and accurate solution for carbon detection than earlier methods, but it is also able to improve the detection of other types of contaminations, such as ice crystals or ethane. Additionally, the usability of the two approaches is very different, as MicrographCleaner is an easy to handle Python package, while Warp is part of a larger framework restricted to Windows systems.

2. Material and methods

2.1 Algorithm

MicrographCleaner computes binary semantic segmentation of micrographs with the aim of delineating optimal regions for particle picking and isolating those areas containing high-contrast contaminants and other artifacts. To that end, MicrographCleaner implements a U-net-like model (Ronneberger et al., 2015) trained on a dataset of 539 manually segmented micrographs collected from 16 different EMPIAR (Iudin et al., 2016) entries. The evaluation was performed on an independent set of 97 micrographs compiled from two EMPIAR projects and another two in-home projects (see Supplementary Material S4). Both training and testing sets of micrographs include examples of clean, carbon-containing and contamination-containing as well as mixed cases. Neural network training was carried out using the Adam optimizer and a combination of perceptual loss (Johnson et al., 2016) and weighted binary cross-entropy (see Supplementary Material S1 and S2). A previous normalization step is required to adjust the different intensity scales of micrographs. Thus, all micrographs are normalized using a robust scaling strategy and donwsampled (see Supplementary Material S3). Finally, overlapping patches of 256x256 pixels with strides of 128 pixels are extracted from the micrograph and fed to the network.

2.1 Package

MicrographCleaner has been implemented as an easy-to-install and easy-to-employ Python 3.x package. Thus, the command line tool can be automatically installed from Anaconda Cloud and PyPI repositories whereas the GUI version can be installed through the Scipion (de la Rosa-Trevín et al., 2016) plugin manager. The neural network was implemented using the Keras (Chollet, 2015) package and the Tensorflow (Abadi *et al.*, 2016) backend. Micrograph preprocessing is carried out using the scikit-image (van der Walt et al., 2014) package.

3. Results

The evaluation of MicrographCleaner was performed comparing the predicted masks with the ground truth of testing micrographs. To that end, the mean Intersection over Union (mIoU) metric was calculated considering predicted and manually curated micrograph regions (see Supplementary Material S2). MicrographCleaner achieved a mIoU value of 0.544. This score implies a good agreement between ground truth and predicted masks, especially when taking into account that the testing set contains clean micrographs examples together with carbon-containing and contaminated micrographs. Figure 1 shows the predictions for four different micrographs, illustrating that MicrographCleaner is capable of successfully detecting both contaminations and carbon.

We also have compared MicrographCleaner with state-of-the-art carbon finder programs: em_hole_finder, EMHP and the Warp particle picker (WPP). Before entering into these comparisons, it is important to highlight that MicrographCleaner and the WPP, contrary to the others, are fast (in the order of seconds), parameter-free and they do not require manual intervention in order to determine whether or not carbon is present in a micrograph. Consequently, they can be employed in automatic pipelines and, thus, they are suitable for automatic Cryo-EM analysis at facilities. Yet, with the aim of strictly comparing carbon detection efficacy, we have taken the subset of the testing set in which all micrographs contain some carbon and executed the four algorithms. As it can be appreciated in Table 1 and in Supplementary Figure 1, deep learning-based methods are very well suited for this problem as both Warp and MicrographCleaner stand out from the others. Still, MicrographCleaner achieves the best performance of all them by a wide margin, improving results over the second best, WPP, by more than 20% in terms of agreement between masks predictions and ground truth. Additionally, we have also compared the performance of MicrographCleaner and WPP on the whole testing set, measuring a mean Intersection over Union (mIoU) of 0.544 for MicrographCleaner and 0.331 for WPP, showing how the more than 20% better performance of MicrographCleaner over WPP is also maintained in that data set (see Supplementary Material S5)

Table 1. MicrographCleaner performance for carbon detection compared to other methods.

Algorithm	**mIoU**	**stdIoU**	**Failure percentage**
MicrographCleaner	0.78833	0.22939	3.33%
EMHP	0.19805	0.21147	45.00%
em_hole_finder	0.05691	0.04691	63.00%
Warp Particle Picker	0.57297	0.23095	3.33%

Notes: mIoU: mean Intersection over Union (mean fraction of agreement between predictions and ground truth) ; stdIoT: standard deviation Intersection over Union; Failure percentage: percentage of the testing set for which the IoU was equal to 0, that its, those micrographs in which no a single pixel of carbon was detected independently of the quality of the prediction.

4. Conclusions

MicrographCleaner is an easy-to-install and easy-to-use python package that allows efficient and automatic micrograph segmentation with the aim of preventing particle pickers from selecting inappropriate regions. To that end, MicrographCleaner relays on a U-net-like model that has being trained on about 500 micrographs. When compared to other methodologies, MicrographCleaner has proven more robust, achieving results closer to the human criterion than other state-of-the-art methods. As a result, we consider that MicrographCleaner is a powerful approach to be applied at the very beginning of cryo-EM workflows, even within on-the-fly processing pipelines, leading to cleaner sets of input particle and, consequently, to a better processing performance.

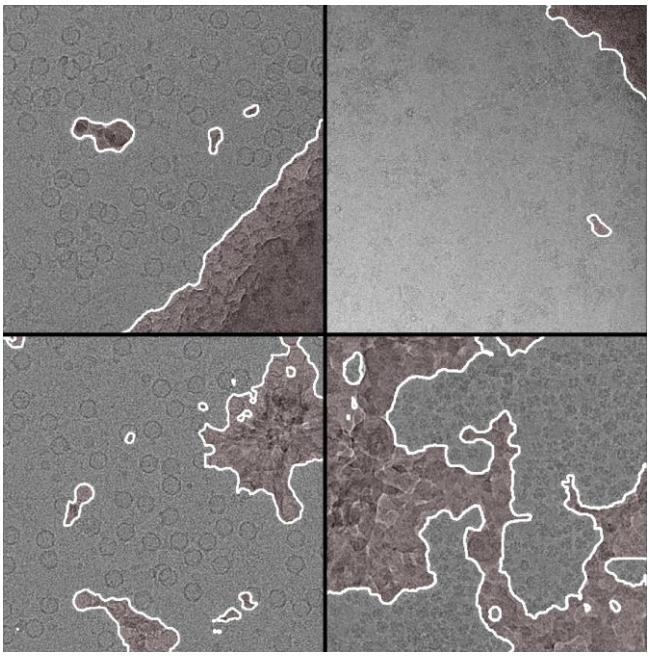

Fig 1. MicrographCleaner predictions. Red shadowed regions correspond to micrograph areas labeled as "non suitable" with 50% or more confidence. Top images show MicrographCleaner capability to detect carbon in the presence of contaminants. Bottom images show MicrographCleaner capability to detect a wide variety of different contaminants.

Funding

The Spanish Ministry of Economy and Competitiveness through Grants BIO2016-76400-R(AEI/FEDER, UE) and the "Comunidad Autónoma de Madrid" through Grant: S2017/BMD-3817. Ruben Sanchez-Garcia is recipient of an FPU fellowship. The authors acknowledge the support and the use of resources of Instruct-ERIC, a Landmark ESFRI project.

References

Abadi,M. *et al.* (2016) TensorFlow: A system for large-scale machine learning. In, *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*., pp. 265–283.

Berndsen,Z. *et al.* (2017) EMHP: an accurate automated hole masking algorithm for single-particle cryo-EM image processing. *Bioinformatics*, **33**, 3824–3826.

Chollet,F. (2015) Keras.

Iudin,A. *et al.* (2016) EMPIAR: a public archive for raw electron microscopy image data. *Nat. Methods*, **13**, 387–388.

Johnson,J. *et al.* (2016) Perceptual losses for real-time style transfer and super-resolution. In, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*., pp. 694–711.

de la Rosa-Trevín,J.M. *et al.* (2016) Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy. *J. Struct. Biol.*, **195**, 93–99.

Lander,G.C. *et al.* (2009) Appion: an integrated, database-driven pipeline to facilitate EM image processing. *J. Struct. Biol.*, **166**, 95–102.

Ronneberger,O. *et al.* (2015) U-net: Convolutional networks for biomedical image segmentation. In,

Medical Image Computing and Computer-Assisted Intervention-MICCAI., pp. 234–241.

Sanchez-Garcia,R. *et al.* (2018) Deep Consensus, a deep learning-based approach for particle pruning in cryo-electron microscopy. *IUCrJ*, **5**, 854–865.

Tegunov,D. and Cramer,P. (2018) Real-time cryo-EM data pre-processing with Warp. *bioRxiv*, 338558.

Vargas,J. *et al.* (2013) Particle quality assessment and sorting for automatic and semiautomatic particle-picking techniques. *J. Struct. Biol.*, **183**, 342–353.

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

Zhu,Y. *et al.* (2004) Automatic particle selection: Results of a comparative study. In, *Journal of Structural Biology*., pp. 3–14.