Contents lists available at ScienceDirect

Journal of Structural Biology

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

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

Ruben Sanchez-Garcia[a,*], Joan Segura[b], David Maluenda[a], C.O.S. Sorzano[a], J.M. Carazo[a]

[a] National Center of Biotechnology (CSIC)/Instruct Image Processing Center, C/ Darwin n° 3, Campus of Cantoblanco, 28049 Madrid, Spain
[b] Research Collaboratory for Structural Bioinformatics Protein Data Bank, San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA 92093, USA

ARTICLE INFO

Keywords:
Cryo-EM
Deep learning
Micrographs
Cleaning
Carbon
Contaminants

ABSTRACT

Cryo-EM Single Particle Analysis workflows require tens of thousands of high-quality particle projections to unveil the three-dimensional structure of macromolecules. Conventional methods for automatic particle picking tend to suffer from high false-positive rates, hampering the reconstruction process. One common cause of this problem is the presence of carbon and different types of high-contrast contaminations. In order to overcome this limitation, we have developed MicrographCleaner, a deep learning package designed to discriminate, in an automated fashion, between regions of micrographs which are suitable for particle picking, and those which are not. MicrographCleaner implements a U-net-like deep learning model trained on a manually curated dataset compiled from over five hundred micrographs. The benchmarking, carried out on approximately one hundred independent micrographs, shows that MicrographCleaner is a very efficient approach for micrograph pre-processing. MicrographCleaner (micrograph_cleaner_em) package is available at PyPI and Anaconda Cloud and also as a Scipion/Xmipp protocol. Source code is available at https://github.com/rsanchezgarc/micrograph_cleaner_em.

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 starting from thousands of particle projections, leads to the reconstruction of density maps of macromolecules. The massive number of particles that are needed for SPA has made automatic particle picking one of the most important steps in virtually all reconstruction workflows. However, 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 (Vargas et al., 2013; Zhu et al., 2004) and leads to the addition of false positive particles in SPA workflows. This problem can be mitigated through the use of different algorithms which 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 hole 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. Furthermore, both require human supervision in order 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, deep learning particle pickers have been developed with the aim of improving picking accuracy. (Bepler et al., 2019; Wagner et al., 2019; Wang et al., 2016; Zhang et al., 2019; Zhu et al., 2017). These new particle pickers are more robust to false positives and most of them have been explicitly or implicitly designed to avoid carbon areas and large contaminants. One such explicitly designed particle picker is included in the Warp package (Tegunov and Cramer, 2019). Thus, the Warp picking algorithm approaches the problem of particle picking by performing a pixel-wise classification (segmentation) of the micrographs in which one of the possible categories is the undesirable region.

[*] Corresponding author.
E-mail address: rsanchez@cnb.csic.es (R. Sanchez-Garcia).

https://doi.org/10.1016/j.jsb.2020.107498
Received 3 February 2020; Received in revised form 3 March 2020; Accepted 26 March 2020
Available online 07 April 2020

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

Algorithm	mIoU	stdIoU
MicrographCleaner	0.78833	0.22939
EMHP	0.19805	0.21147
em_hole_finder	0.05691	0.04691
Warp Particle Picker	0.57297	0.23095

Notes: mIoU: mean Intersection over Union (mean fraction of agreement between predictions and ground truth); stdIoU: standard deviation Intersection over Union.

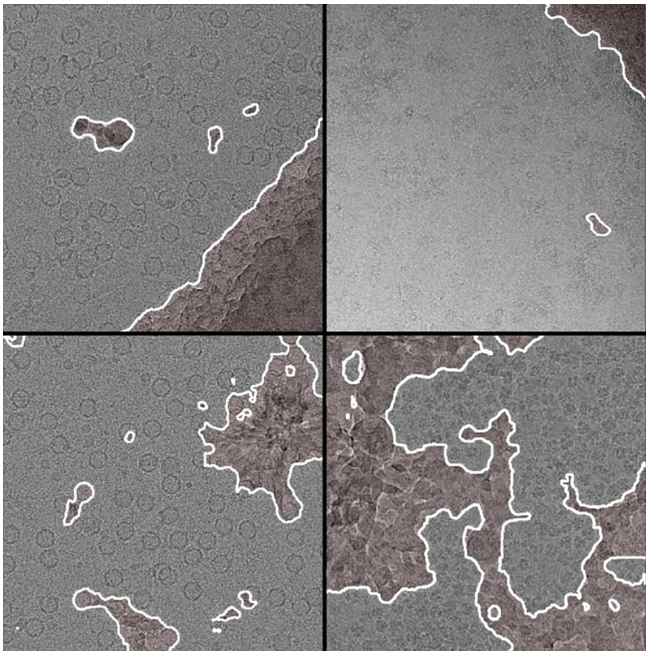


Fig. 1. MicrographCleaner identifies non-suitable regions. Red shadowed regions correspond to micrograph areas labeled as “non-suitable” with 50% or greater confidence. Top images show MicrographCleaner capability to detect carbon in the presence of contaminants. Bottom images show MicrographCleaner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

In spite of these developments, conventional particle pickers are still the preferred choice in recent publications (Gilman et al., 2019; Hiraizumi et al., 2019; Jain et al., 2019; Molina et al., 2019; Stone et al., 2019; Yan et al., 2019). Although it is likely that deep learning particle pickers will become increasingly popular, they are not perfect, with different situations requiring different approaches, thus conventional particle pickers, especially those based on templates, will likely remain popular.

In response to these challenges, and with the aim of improving classical particle pickers and complementing deep-learning-based ones, 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, separating them into two categories, desirable and undesirable regions for picking. Like Warp particle picker, MicrographCleaner relies upon one of the most extended network architectures (Ronneberger et al., 2015), the U-net. However, the different choices in important parameters result, in turn, in quite different levels of performance. Thus, according to our benchmark, MicrographCleaner is not only able to provide a more robust and accurate solution for carbon detection than previous methods, but it is also able to improve the detection of other types of contaminants, such as ice crystals or ethane bubbles.

2. Material and methods

2.1. Algorithm

MicrographCleaner computes binary 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 architecture (Ronneberger et al., 2015). Our model, carefully selected after a cross-validation process, consists of 5 downsampling blocks followed by 5 upsampling blocks with 32, 64, 128, 256, and 512 kernels per block respectively. Further details are described in Supplementary Material S1 and S4.

Neural network training was carried out during 200 epochs using the Adam optimizer and a combination of perceptual loss (Johnson et al., 2016) and weighted binary cross-entropy (Falk et al., 2019). Data augmentation was performed during training. See Supplementary Material S2 and S4 for more details.

2.2. Dataset and preprocessing

MicrographCleaner was 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-house projects (see Supplementary Material S5). Both training and testing set micrographs include examples of clean, carbon-containing, contamination-containing and aggregation-containing areas as well as mixed ones that were labeled by an expert.

Before micrographs are fed to the network, a previous normalization step is required to adjust the different intensity scales and sizes of micrographs. Thus, all micrographs are normalized in both intensity and size using a robust scaling strategy and a constant particle size donwsampling (see Supplementary Material S3). Finally, due to GPU memory limitations, the full downsampled micrograph is processed in chunks using a sliding window approach of overlapping patches of size 256 × 256.

2.3. Evaluation metrics

As evaluation criterium, we computed the Intersection over Union (IoU) metric between the network predictions and the manually curated masks and averaged it for all micrographs in the testing set (mIoU). Consequently, mIoU is defined as:

$$mIoU = \sum_{i=1}^{N} \frac{|P_i \cap L_i|}{|P_i \cup L_i|}$$

where i is the testing micrograph index, N the number of testing micrographs and P_i and L_i are, respectively, the predicted mask and the manually curated mask for testing micrograph i.

2.4. Package

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

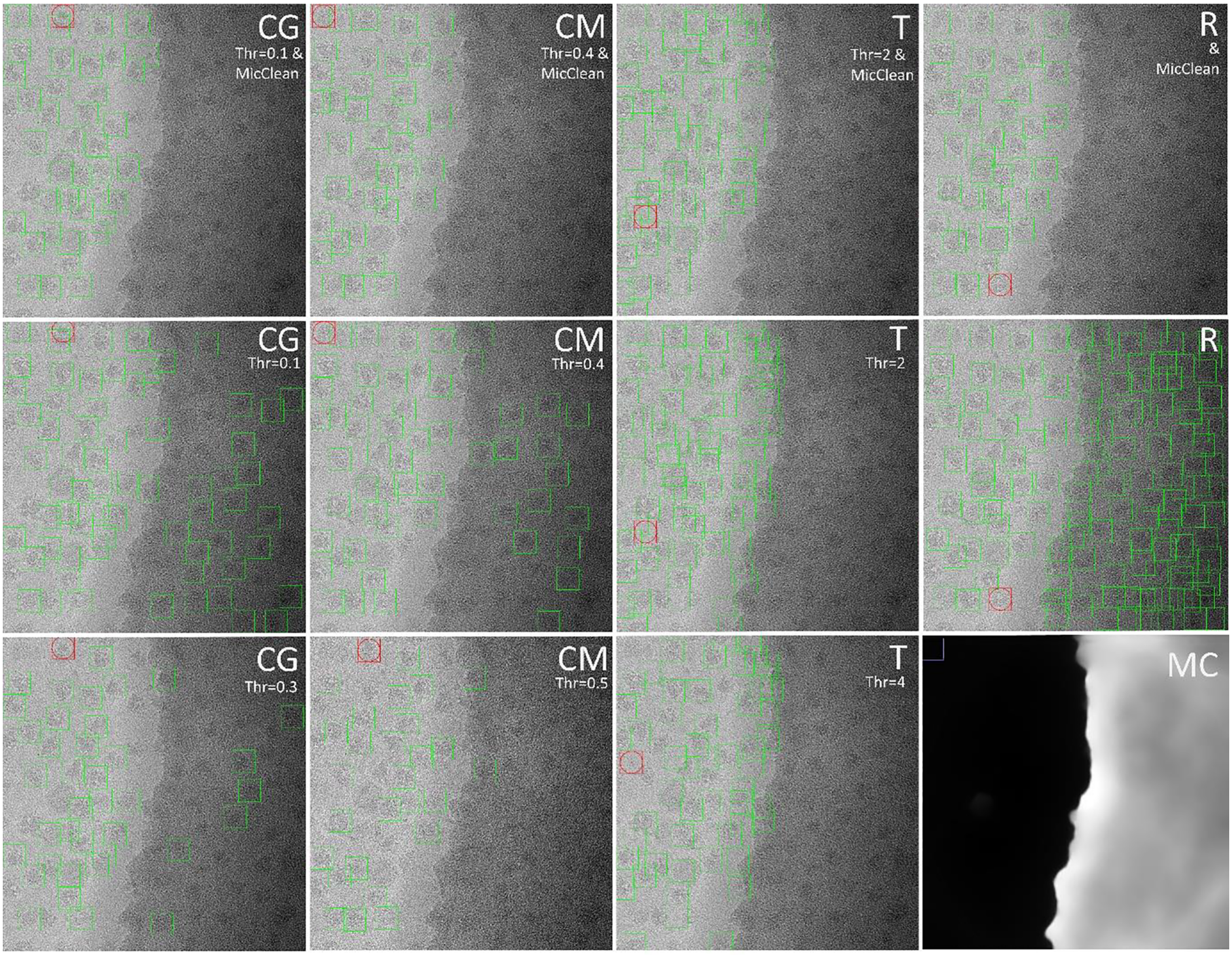


Fig. 2. MicrographCleaner improves particle picking on EMPIAR-10156 dataset. Coordinates selected with Cryolo pretrained general model (CG), Cryolo manually trained model (CM), Topaz (T) and Relion autopicker (R) are respectively displayed in columns one to four. Top row images correspond to the remaining particles after applying MicrographCleaner mask (MC) to the low threshold Topaz, Cryolo general and Cryolo manual solutions and the Relion autopicker outcome. As it can be seen, MicrographCleaner removes the particles selected in the carbon area and its edge while preserving many more true positive particles than using stricter thresholds. Red box represents the lowest confidence particle according the picking algorithm.

3. Results

3.1. Carbon detection

In order to estimate carbon detection capability, we took the subset of the testing set micrographs in which all contain some carbon and executed MicrographCleaner on them, achieving a mIoU of 0.78833, which indicated a good agreement between the carbon areas manually curated and the predicted ones. Additionally, we compared MicrographCleaner with several carbon finder programs: em_hole_finder, EMHP and the Warp particle picker (WPP) (see Supplementary Material S6 for more details). Before entering into these comparisons, it is important to highlight that both MicrographCleaner and the WPP, in comparison to the other programs, 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. As can be seen in Table 1 and in Supplementary Material SM1, deep learning-based methods are very well suited to this problem, both Warp and MicrographCleaner stand out from the others. Moreover, MicrographCleaner achieves the best performance of all them by a wide margin, improving upon the results of the second best, WPP, by more than 20% in terms of agreement between masks predictions and ground truth.

3.2. Undesirable regions and contaminants detection

MicrographCleaner evaluation for undesirable regions and small contaminants detection was performed comparing the predicted masks with the ground truth for all the testing micrographs. Under this test, MicrographCleaner achieved a mIoU value of 0.544. This score, although lower than the score for carbon detection, 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. Fig. 1 shows the predictions for four different micrographs, illustrating that MicrographCleaner is capable of successfully detecting both contaminants and carbon.

Additionally, we have also evaluated the global performance of WPP on the whole testing set, showing a mIoU of 0.331 and performing worse than MicrographCleaner for 77% of the micrographs. This supposes that the 20% better performance of MicrographCleaner over WPP for carbon detection is also maintained when contaminants detection is also considered. The predictions for some micrographs using both

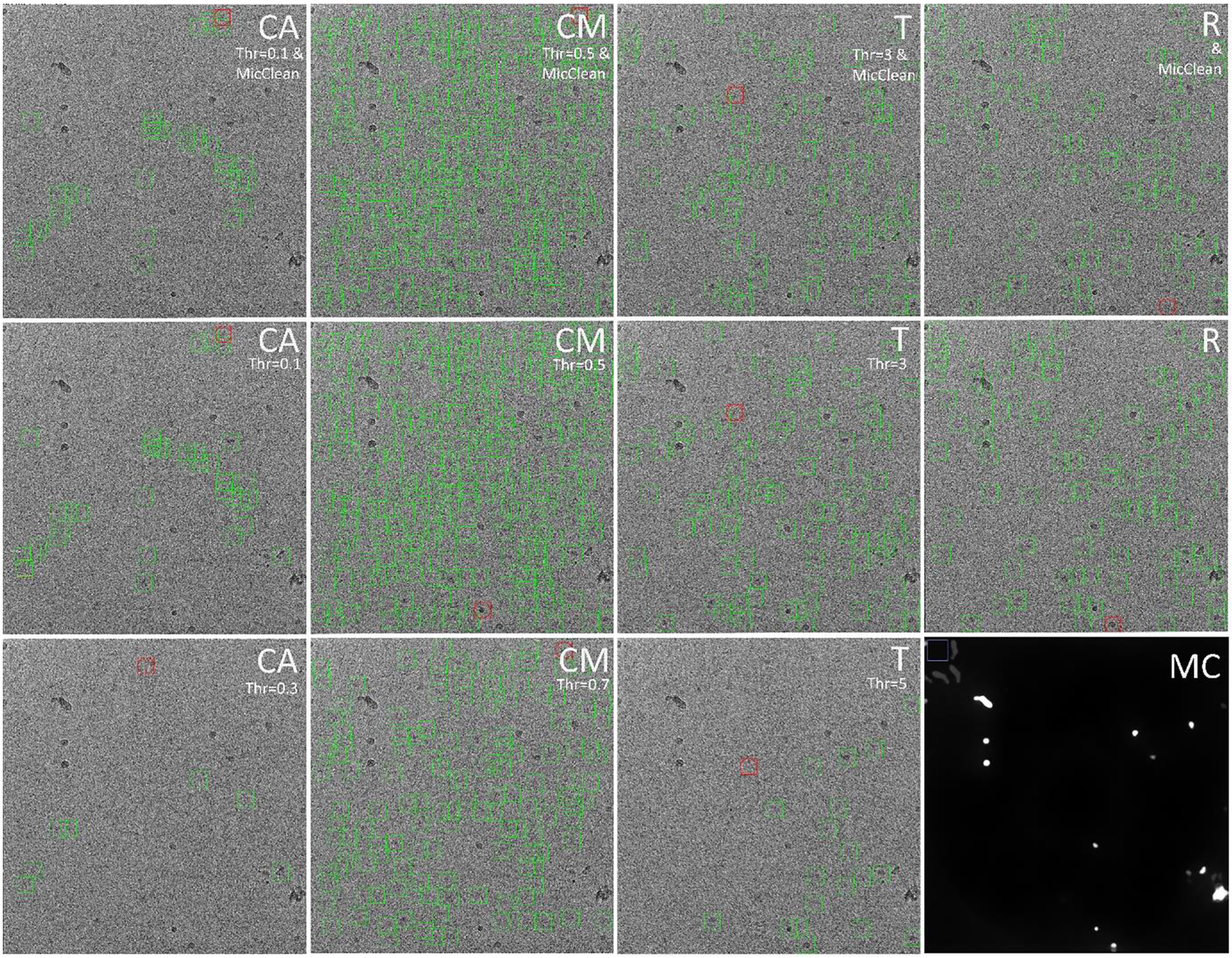


Fig. 3. MicrographCleaner improves particle picking on EMPIAR-10265 dataset. Coordinates selected with Cryolo pretrained general model (CG), Cryolo manually trained model (CM), Topaz (T) and Relion autopicker (R) are respectively displayed in columns one to four. Top row images correspond to the remaining particles after applying MicrographCleaner mask (MC) to the low threshold Topaz, Cryolo general and Cryolo manual solutions and the Relion autopicker outcome. As it can be seen, MicrographCleaner removes many of the contaminants incorrectly selected as particles while preserving much more true positive particles than using stricter thresholds. Red box represents the lowest confidence particle according the picking algorithm.

MicrographCleaner and WPP are shown in Supplementary Material Figure SM2.

3.3. Use cases

In this section we present two examples, not included in the training and testing sets, in which both traditional particle pickers and deep-learning-based pickers struggle discerning problematic regions and contaminants from clean regions, and thus they both could benefit from MicrographCleaner. As deep learning representatives, we chose Topaz (Bepler et al., 2019) and the Cryolo (Wagner et al., 2019) particle pickers. Both Cryolo and Topaz algorithms were trained using ten manually curated micrographs. Additionally, the Cryolo general model, which does not require any training, was also employed. Relion autopicker (Scheres, 2015) was chosen as the representative of traditional particle pickers. Further details can be found in Supplementary Material S8.

3.3.1. Empiar-10156

The main difficulties for particle pickers that EMPIAR-10156 dataset (von Loeffelholz et al., 2018) presents is that it contains large areas of carbon (greater than 50% of the micrograph) and that the intensity of these areas is not uniform, either within an individual micrograph or across the whole dataset. Thus, as it is illustrated in Fig. 2, both the Relion and the Cryolo particle pickers (using a general model and a trained one) tend to pick particles located at the carbon region, whereas Topaz particle picker is able to avoid most of the carbon region, although it still selects many false positives at the edge.

It is interesting to note that, although the number of particles picked at the carbon area/edge can easily be decreased using stricter thresholds, it comes at the cost of ruling out true positive particles. Thus, as it is shown in Fig. 2, large enough thresholds for discarding most of the false positive particles would cause the rejection of some true positive particles. Ultimately, this translates to the precision/recall tradeoff in which most people favor the latter option aiming to remove false positives in successive steps. On the other hand, MicrographCleaner is able to mask out those false positive particles while not affecting the true positive ones, hence it can be used as a complement for any particle picker independently of threshold decisions. This behavior is illustrated in Fig. 2, in which MicrographCleaner proposed solutions were better than the solutions obtained directly by using the other methods at different thresholds. For more details see Supplementary Material S8.

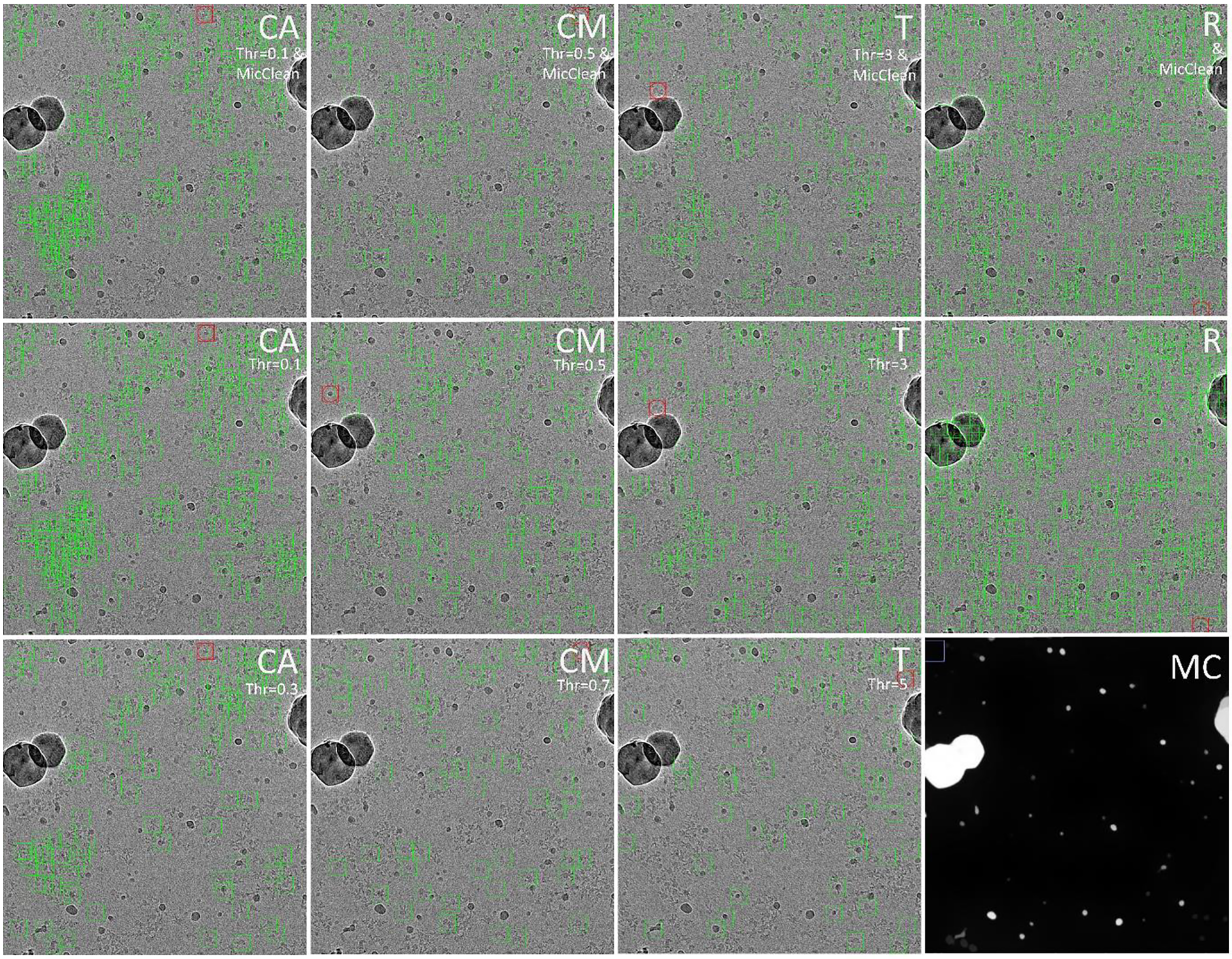


Fig. 4. MicrographCleaner improves particle picking on EMPIAR-10265 dataset. Coordinates selected with Cryolo pretrained general model (CG), Cryolo manually trained model (CM), Topaz (T) and Relion autopicker (R) are respectively displayed in columns one to four. Top row images correspond to the remaining particles after applying MicrographCleaner mask (MC) to the low threshold Topaz, Cryolo general and Cryolo manual solutions as well as the Relion autopicker outcome. As it can be seen, MicrographCleaner removes many of the contaminants incorrectly selected as particles while preserving much more true positive particles than using stricter thresholds. Red box represents the lowest confidence particle according to the respective picking algorithm.

3.3.2. *Empiar-10265*

The EMPIAR-10265 dataset (Lee et al., 2019) is extremely challenging. In this dataset, the particles of most micrographs are difficult to visualize, whereas in some others they are easily recognizable (see Figs. 3 and 4 respectively). Due to this profound disparity, the performance of the employed deep-learning-based methods is worse than in other datasets, and although they are able to avoid large contaminated regions, they still incorrectly select many small contaminants as particles, as it is illustrated in Figs. 3 and 4. Again, as per the previous example, the number of selected contaminants can be reduced by increasing the threshold but, as a result, the total number of particles would be severely reduced.

Thus, the process of threshold selection for this dataset is not trivial, as micrographs differ enormously and thresholds that detect most of the particles in some micrographs discard many particles in others. As a consequence, manual inspection for each micrograph should be performed to obtain the best balance between the number of removed contaminants and total number of recovered particles. Alternatively, although it remains costly, more micrographs could be picked manually in order to further train some of the methods.

On the other hand, when MicrographCleaner was applied to the particles that had been selected using a conservative threshold, more true positive particles were recovered while ruling out most of the small contaminants that were incorrectly selected (see Figs. 3 and 4). This ultimately improved the quality of the set of picked particles and also simplified threshold selection, which can be set to more conservative values with the confidence that contaminants will be equally removed. See Supplementary Material S8 for additional information.

3.4. *MicrographCleaner complements 2D-classification*

Although the previous section demonstrates that MicrographCleaner is able to reduce false positive levels for many particle pickers, it could also be argued that this reduction is not of enormous impact as such a reduction could equally be achieved via subsequent steps of the image processing workflow, particularly at the 2D-classification step. With the aim of testing this hypothesis, we conducted one 2D classification analysis for each of the particle sets picked by the four particle pickers considered in Section 3.3.1 and we compared the outcome of all of them with the particle sets processed with MicrographCleaner. Fig. 5 illustrates the experiment for one of the picked sets of particles (see Supplementary Material S9 for additional information and other examples).

Roughly speaking, our results point out that 2D-clustering is a much

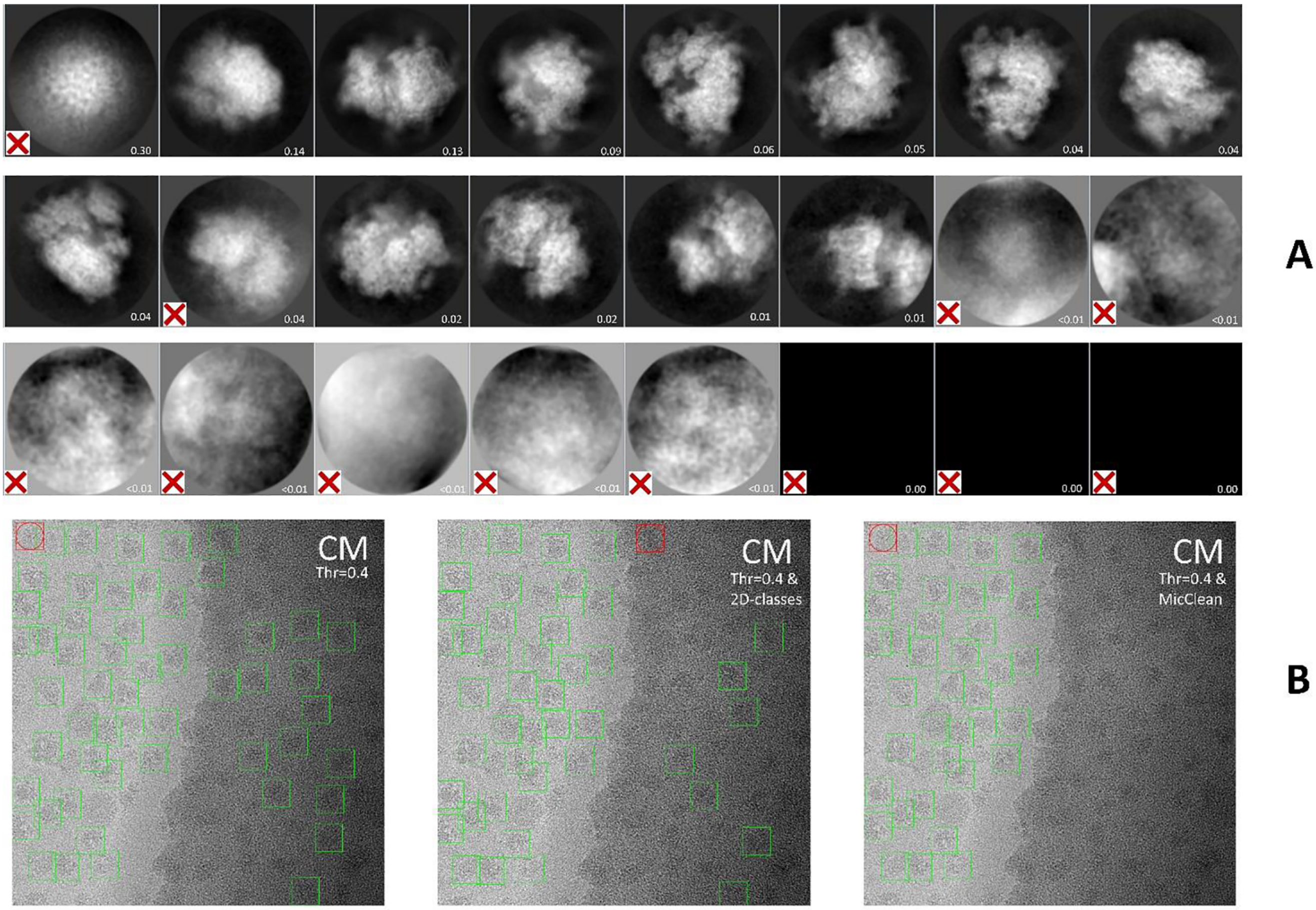


Fig. 5. MicrographCleaner complements 2D-classification. A: Gallery of 2D averages obtained from the set of particles collected by Cryolo manually trained on EMPIAR-10265 dataset. B, from left to right: (left) Particles originally picked by Cryolo and used as input for 2D-classification; (middle) the previous set of particles after cleaning by a round of 2D classification (note that discarded particles correspond to those ones belonging to rejected 2D classes, which are marked with a red cross in A); (right) Cryolo original set of particles after application of MicrographCleaner. It can be seen that MicrographCleaner removed all particles picked on carbon area but 2D-classification did not.

more aggressive strategy which removes many more particles than MicrographCleaner (between 20% and 40% compared to 9% to 25%). Obviously, these results should not be surprising as MicrographCleaner was not designed to remove some types of false positive cases (e.g. background) which 2D-classification can.

However, the most interesting conclusions can be drawn when counting the number of particles removed by MicrographCleaner but not ruled out after 2D-classification (we acknowledged that particle pruning through 2D classification has a certain subjectivity, difficult to reproduce precisely). Thus, we have measured that between 19% and 29% of the particles discarded by MicrographCleaner survived to the 2D-classification process. Even more interestingly, when a second step 2D-classification is performed, the number of particles not removed, although fewer, is still considerable (between 10% and 20%, see Supplementary Material S9). These numbers suggest that MicrographCleaner and 2D-classification should better be regarded as complementary options rather than competitors.

4. Discussion

Deep learning particle pickers are increasingly gaining popularity. Their ability to avoid contaminated regions and their reported superior accuracies compared with traditional approaches explains this trend. Yet, traditional particle pickers are still the preferred option, as seen in recent publications. Irrespective of the particular method that a researcher considers appropriate for a specific case, we introduce here an approach that is specifically tailored to detect those particles that are located in problematic areas of the micrograph. In other words, rather than concentrating on reporting specimen-like images, we focus on detecting those areas of the micrograph that are likely to contribute with lower quality particles. Thus, we can select from any picking method only those particles found in the best areas of the micrograph. Our evaluations have also shown how this contextual approach can provide an excellent complement to other traditional particles selection procedures. For example, we see that in pruning by 2D classification, quite a substantial percentage of incorrect particles tended to be accepted by 2D classification cleaning, which in contrast, were detected and discarded using our methodology. Thus, following the general trend in the machine learning field in which top performing solutions are based on ensembles of methods, it is very likely (indeed, it is our vision) that top performing image processing or preprocessing workflows will likely be constructed by combining different approaches, MicrographCleaner included, especially when facing difficult samples.

5. 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 on the micrograph. To that end, MicrographCleaner relies on a U-net-like model that has being trained on approximately 500 micrographs. When compared to other methodologies, MicrographCleaner has proven more robust, achieving results closer to the human criterion than other methods for both carbon and contaminant detection. In conclusion, we consider MicrographCleaner to be a powerful approach which can be applied at the very beginning of cryo-EM workflows, even within on-the-fly/streaming processing pipelines, leading to cleaner sets of input particle and, consequently, to a better processing performance.

CRediT authorship contribution statement

Ruben Sanchez-Garcia: Conceptualization, Methodology, Software, Data curation, Writing - original draft. **Joan Segura:** Methodology, Validation, Writing - review & editing. **David Maluenda:** Software, Data curation. **C.O.S. Sorzano:** Supervision, Writing - review & editing. **J.M. Carazo:** Supervision, Funding acquisition, Writing - review & editing.

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.

Acknowledgements

The authors would like to acknowledge economical support from: The Spanish Ministry of Economy and Competitiveness through Grants BIO2016-76400-R(AEI/FEDER, UE) and the “Comunidad Autónoma de Madrid and Ministry of Education of Spain” 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.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jsb.2020.107498.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283.

Bepler, T., Morin, A., Rapp, M., Brasch, J., Shapiro, L., Noble, A.J., Berger, B., 2019. Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs. Nat. Methods 16, 1153–1160. https://doi.org/10.1038/s41592-019-0575-8.

Berndsen, Z., Bowman, C., Jang, H., Ward, A.B., 2017. EMHP: an accurate automated hole masking algorithm for single-particle cryo-EM image processing. Bioinformatics 33, 3824–3826. https://doi.org/10.1093/bioinformatics/btx500.

Chollet, F., 2015. Keras.

de la Rosa-Trevín, J.M., Quintana, A., del Cano, L., Zaldívar, A., Foche, I., Gutiérrez, J., Gómez-Blanco, J., Burguet-Castell, J., Cuenca-Alba, J., Abrishami, V., Vargas, J., Otón, J., Sharov, G., Vilas, J.L., Navas, J., Conesa, P., Kazemi, M., Marabini, R., Sorzano, C.O.S., Carazo, J.M., 2016. Scipion: a software framework toward integration, reproducibility and validation in 3D electron microscopy. J. Struct. Biol. 195, 93–99. https://doi.org/10.1016/j.jsb.2016.04.010.

Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., Dovzhenko, A., Tietz, O., Dal Bosco, C., Walsh, S., Saltukoglu, D., Tay, T.L., Prinz, M., Palme, K., Simons, M., Diester, I., Brox, T., Ronneberger, O., 2019. U-Net: deep learning for cell counting, detection, and morphometry. Nat. Methods 16, 67–70. https://doi.org/10.1038/s41592-018-0261-2.

Gilman, M.S.A., Liu, C., Fung, A., Behera, I., Jordan, P., Rigaux, P., Ysebaert, N., Tcherniuk, S., Sourimant, J., Eléouët, J.-F., Sutto-Ortiz, P., Decroly, E., Roymans, D., Jin, Z., McLellan, J.S., 2019. Structure of the respiratory syncytial virus polymerase complex. Cell 179, 193–204.e14. https://doi.org/10.1016/j.cell.2019.08.014.

Hiraizumi, M., Yamashita, K., Nishizawa, T., Nureki, O., 2019. Cryo-EM structures capture the transport cycle of the P4-ATPase flippase. Science (80-.) 365, 1149–1155. https://doi.org/10.1126/science.aay3353.

Iudin, A., Korir, P.K., Salavert-Torres, J., Kleywegt, G.J., Patwardhan, A., 2016. EMPIAR: a public archive for raw electron microscopy image data. Nat. Methods 13, 387–388. https://doi.org/10.1038/nmeth.3806.

Jain, R., Rice, W.J., Malik, R., Johnson, R.E., Prakash, L., Prakash, S., Ubarretxena-Belandia, I., Aggarwal, A.K., 2019. Cryo-EM structure and dynamics of eukaryotic DNA polymerase δ holoenzyme. Nat. Struct. Mol. Biol. 26, 955–962. https://doi.org/10.1038/s41594-019-0305-z.

Johnson, J., Alahi, A., Fei-Fei, L., 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. https://doi.org/10.1007/978-3-319-46475-6_43.

Lander, G.C., Stagg, S.M., Voss, N.R., Cheng, A., Fellmann, D., Pulokas, J., Yoshioka, C., Irving, C., Mulder, A., Lau, P.-W., Lyumkis, D., Potter, C.S., Carragher, B., 2009. Appion: an integrated, database-driven pipeline to facilitate EM image processing. J. Struct. Biol. 166, 95–102.

Lee, Y., Wiriyasermkul, P., Jin, C., Quan, L., Ohgaki, R., Okuda, S., Kusakizako, T., Nishizawa, T., Oda, K., Ishitani, R., Yokoyama, T., Nakane, T., Shirouzu, M., Endou, H., Nagamori, S., Kanai, Y., Nureki, O., 2019. Cryo-EM structure of the human L-type amino acid transporter 1 in complex with glycoprotein CD98hc. Nat. Struct. Mol. Biol. 26, 510–517. https://doi.org/10.1038/s41594-019-0237-7.

Molina, R., Stella, S., Feng, M., Sofos, N., Jauniskis, V., Pozdnyakova, I., López-Méndez, B., She, Q., Montoya, G., 2019. Structure of Csx1-cOA4 complex reveals the basis of RNA decay in Type III-B CRISPR-Cas. Nat. Commun. 10, 4302. https://doi.org/10.1038/s41467-019-12244-z.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention-MICCAI, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.

Sanchez-Garcia, R., Segura, J., Maluenda, D., Carazo, J.M., Sorzano, C.O.S., 2018. Deep Consensus, a deep learning-based approach for particle pruning in cryo-electron microscopy. IUCrJ 5, 854–865. https://doi.org/10.1107/S2052252518014392.

Scheres, S.H.W., 2015. Semi-automated selection of cryo-EM particles in RELION-1.3. J. Struct. Biol. 189, 114–122. https://doi.org/10.1016/J.JSB.2014.11.010.

Stone, N.P., Demo, G., Agnello, E., Kelch, B.A., 2019. Principles for enhancing virus capsid capacity and stability from a thermophilic virus capsid structure. Nat. Commun. 10, 4471. https://doi.org/10.1038/s41467-019-12341-z.

Tegunov, D., Cramer, P., 2019. Real-time cryo-electron microscopy data preprocessing with Warp. Nat. Methods 16, 1146–1152. https://doi.org/10.1038/s41592-019-0580-y.

van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., 2014. scikit-image: image processing in Python. PeerJ 2, e453. https://doi.org/10.7717/peerj.453.

Vargas, J., Abrishami, V., Marabini, R., de la Rosa-Trevín, J.M., Zaldivar, A., Carazo, J.M., Sorzano, C.O.S., 2013. Particle quality assessment and sorting for automatic and semiautomatic particle-picking techniques. J. Struct. Biol. 183, 342–353. https://doi.org/10.1016/J.JSB.2013.07.015.

von Loeffelholz, O., Papai, G., Danev, R., Myasnikov, A.G., Natchiar, S.K., Hazemann, I., Ménétret, J.F., Klaholz, B.P., 2018. Volta phase plate data collection facilitates image processing and cryo-EM structure determination. J. Struct. Biol. 202, 191–199. https://doi.org/10.1016/j.jsb.2018.01.003.

Wagner, T., Merino, F., Stabrin, M., Moriya, T., Antoni, C., Apelbaum, A., Hagel, P., Sitsel, O., Raisch, T., Prumbaum, D., Quentin, D., Roderer, D., Tacke, S., Siebolds, B., Schubert, E., Shaikh, T.R., Lill, P., Gatsogiannis, C., Raunser, S., 2019. SPHIRE-crYOLO is a fast and accurate fully automated particle picker for cryo-EM. Commun. Biol. 2. https://doi.org/10.1038/s42003-019-0437-z.

Wang, F., Gong, H., Liu, G., Li, M., Yan, C., Xia, T., Li, X., Zeng, J., 2016. DeepPicker: A deep learning approach for fully automated particle picking in cryo-EM. J. Struct. Biol. 195, 325–336. https://doi.org/10.1016/j.jsb.2016.07.006.

Yan, K., Yang, J., Zhang, Z., McLaughlin, S.H., Chang, L., Fasci, D., Ehrenhofer-Murray, A. E., Heck, A.J.R.R., Barford, D., 2019. Structure of the inner kinetochore CCAN complex assembled onto a centromeric nucleosome, Nature. Nature Publishing Group. https://doi.org/10.1038/s41586-019-1609-1.

Zhang, J., Wang, Z., Chen, Y., Han, R., Liu, Z., Sun, F., Zhang, F., 2019. PIXER: an automated particle-selection method based on segmentation using a deep neural network. BMC Bioinf. 20, 41. https://doi.org/10.1186/s12859-019-2614-y.

Zhu, Y., Carragher, B., Glaeser, R.M., Fellmann, D., Bajaj, C., Bern, M., Mouche, F., De Haas, F., Hall, R.J., Kriegman, D.J., Ludtke, S.J., Mallick, S.P., Penczek, P.A., Roseman, A.M., Sigworth, F.J., Volkmann, N., Potter, C.S., 2004. Automatic particle selection: results of a comparative study. J. Struct. Biol. 3–14. https://doi.org/10.1016/j.jsb.2003.09.033.

Zhu, Y., Ouyang, Q., Mao, Y., 2017. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy. BMC Bioinf. 18, 348. https://doi.org/10.1186/s12859-017-1757-y.

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

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

SUPPLEMENTARY MATERIAL

S1. Neural Network Architecture.

A U-net-like (Ronneberger et al., 2015) architecture was employed in this work. Several architectural variations were considered and compared through cross-validation (see S4). The main difference between the classical configuration and ours is that our convolution blocks are indeed residual blocks (Wu et al., 2017) and that we do not perform the crop operation. Other changes are the depth, measured in terms of downsampling operations, which is five instead of four and the size of the input images, which is 256x256 instead 572x572. The number of filters in each convolution operation for each block is 32, 64, 128, 256 and 512 respectively and the filter size is 5x5, except for the first layer, which is 7x7. As activation function, we employed leakyRelu with α=0.05. Batch normalization is also employed. Finally, dropout is added after each residual block of the downsampling part of the network. An scheme of the network architecture can be found in http://campins.cnb.csic.es/micrograph_cleaner/architecture.png and it is summarized in table TS1.1 and TS1.2. The total number of parameters contained in our network is 35,479,712.

Table TS1.1. Neural network architecture of downsampling block number i:

Layer	Type	Parents	# kernels	Kernel size
1	Conv2d+BN+LeakyRelu	Previous_block	2^{i+4}	5x5 (7x7 if i=1)
2	Conv2d+BN	1	2^{i+4}	5x5 (7x7 if i=1)
3	Conv2d+BN	Previous_block	2^{i+4}	1x1
4	Add+LeakyRelu+Dropout	2, 3	None	None
5	MaxPooling	4	None	2x2

Table TS1.2. Neural network architecture of upsampling block number i:

Layer	Type	Parents	# kernels	Kernel size
1	Upsampling2D	Previous_block	None	None
2	Concatenation	1, dowmsampling_block_i_l4	None	None
3	Conv2d+BN+LeakyRelu	2	2^{N-i+5}	5x5
4	Conv2d+BN	3	2^{N-i+5}	5x5
5	Conv2d+BN	2	2^{N-i+5}	1x1
6	Add+LeakyRelu	4, 5	None	None

S2. Neural Network Training.

We have employed as loss function the sum of perceptual loss (Johnson et al., 2016) and weighted binary cross-entropy (Falk et al., 2019) at 1:1 proportion. Other alternatives were ruled out after cross-validation (see Section S4). The rationale behind the addition of the perceptual loss term is that for the particular task of contaminants detection, we are not really concerned about the perfect matching of all the pixels, which in large contaminated regions tend to be homogeneous, but about the detection of informative, and thus, more abstract features such as edges that can help the network to better identify the boundaries of the regions.

In order to obtain the perceptual loss estimator, we have trained a classical VGG-Net-16 (Simonyan and Zisserman, 2014) on a grayscale version of the ImageNet dataset (Deng et al., 2009). The VGG-Net-16 trained network is available at http://campins.cnb.csic.es/imagenet_grayscale). Although it is true that the images contained in the ImageNet dataset are of quite different nature compared to micrographs, it is also true that the purpose of the network is to identify carbon and contaminated areas, which are more similar to natural images in terms of signal to noise ratio. Moreover, it is common in the field of deep learning to use networks trained on ImageNet for fine tuning or feature extraction in other domains such as medical imaging (Bar et al., 2015).

Regarding our U-net-like model, Adam optimizer was employed on batches of size 12. L1 and L2 regularization was added to all kernels weights with strength 1e-4. The network was trained until no improvement in validation loss was detected once two learning rate decays on plateau were performed. Initial learning rate was 1e-2.

Severe data augmentation has been performed at a ratio 1:4 using as transformations translations, rotations, contrast and brightness alterations, local blurring and small zooming in/out (with the aim of dealing with slightly different particle size estimations).

S3. Micrograph preprocessing.

Each micrograph is normalized in intensity by subtracting the median value of its pixels and dividing by the percentile 5-95 range.

$$I_{norm} = \frac{I - median(I)}{percentile_{95}(I) - percentile_{5}(I)}$$

Then, micrographs are downsampled with the aim of normalizing the particle size to 16 pixels. The idea behind this size normalization is that if we are interested in detecting contaminants while preserving particles, having always particles of the same size will simplify the problem. Thus, we apply the following downsampling factor:

$$downFactor = \frac{16}{estimatedParticleWidthInPixels}$$

Thus, the final size of the micrograph is:

$$micrographSize = originalMicrographSize \cdot downFactor$$

When users do not provide an estimation of the particle size, the Scipion “particle boxsize” protocol can be employed to obtain such estimation.

S4. Architectural decision based on cross-validation

In order to perform cross-validation, the training set was divided into training and validations splits at a proportion 10:1. The following tables summarize the most relevant trials regarding the architecture and other hyperparameters. Metrics included in this table are computed per patch (256x256), contrary to metrics reported in the main text that are computed using the whole micrographs.

Table S4.1. Cross-validation results with respect different numbers of blocks

Model number	Training mIoU	Validation mIoU	Depth
1	0.6272	0.5312	4
2	0.6341	0.5498	5
3	0.6661	0.5305	6
4	0.6832	0.4721	7

Table S4.2. Cross-validation results with respect different losses

Model number	Training mIoU	Validation mIoU	Loss
5	0.5991	0.4165	BCE
6	0.6523	0.5212	WBCE
7	0.6221	0.4782	PL
8	0.6341	0.5498	PL+ WBCE

Notes: BCE: binary cross-entropy; WBCE: weighted binary cross-entropy; PL: perceptual loss.

Table S4.1. Cross-validation results with respect downsampling factor

Model number	Training mIoU	Validation mIoU	Particle size
9	0.6001	0.4799	8
10	0.6341	0.5498	16
11	0.6279	0.4812	32

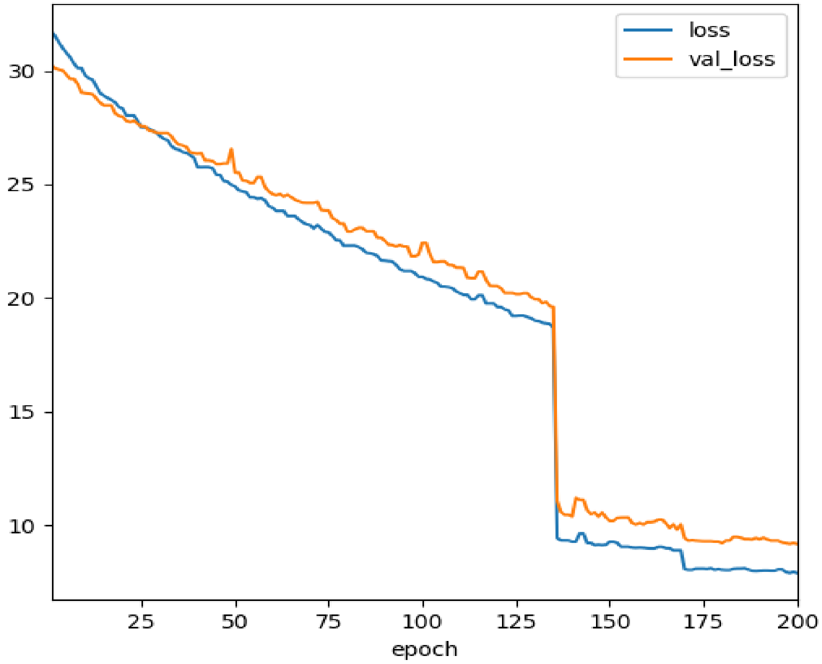


Fig SM1. MicrographCleaner learning curve for the final model.

S5. Datasets

Evaluation datasets

EMPIAR-10205. Cowpea mosaic virus
EMPIAR-10217. Bovine liver glutamate dehydrogenase
In-home dataset1. Phage T7 tails
In-home dataset2. Phage T7 ejection machinery

Training datasets

EMPIAR-10005. TRPV1
EMPIAR-10028. Ribosome
EMPIAR-10033. Picornavirus
EMPIAR-10049. RAG1-RAG2 Complex
EMPIAR-10061. β-galactosidase
EMPIAR-10075. Phage MS2
EMPIAR-10077. Elongation factor SelB
EMPIAR-10081. HCN1 ion channel
EMPIAR-10090. AAA-ATPase in 26S proteasome
EMPIAR-10093. Ion channel in nano disc
EMPIAR-10097. Influenza Hemagglutinin Trimer
EMPIAR-10099. Hrd1 and Hrd3 complex
EMPIAR-10168. RNA Polymerase III pre-initializ
EMPIAR-10175. Hemagglutinin
EMPIAR-10190. RNA Polymerase III transcribing
EMPIAR-10203. Nodavirus

S6. Carbon detection capability comparison

In order to compare MicrographCleaner carbon detection capability with em_hole_finder, EMHP and the Warp particle picker (WPP) BoxNet2Mask_20180918 model, we have executed the four software packages on the same testing dataset consisting of a subset of 60 micrographs, all them containing carbon. Contrary to MicrographCleaner and WPP, EMHP and em_hole_finder require from the user to set certain parameters. For this comparison, we have set those parameters to default values. While custom tuning of them should improve results, we have adopted this approach because of two reasons. First, we have tried different parameters settings and we have not observed too important performance differences on average, since some settings caused some inputs to improve at the cost of worsening others. Secondly, we believe that solutions that require considerable human intervention are in danger of extinction as the cryo-EM field is moving towards streaming and automatic processing and thus, default parameters should perform decently for most cases. Other methods used in computer vision (k-mean clustering and SLIC), where initially considered for comparison purposes, but were discarded for the reasons presents above, particularly, because of the difficulty of setting parameters and thresholds that could be valid for all the elements of the dataset.

The quality of the predicted masks has been assessed by comparing them to ground truth masks manually compiled. In the case of MicrographCleaner and WPP, as they also find contaminated regions, we have only considered the predicted patches that overlapped with the ground truth carbon masks (if they exist), ignoring predictions for contaminated regions. As shown in Main Text Table 1, MicrographCleaner is able to produce predictions much more similar to the ground truth masks than the other methods, with a mean Intersection over Union (mIoU) value close to 0.8. The second-best performing algorithm is WPP, which does a much better job than EMHP and em_hole_finder, although still far from MicrographCleaner performance. The small mean mIoU obtained by EMHP and, especially, by em_hole_finder, can be explained by the lower quality masks they produced when compared with MicrographCleaner and, mainly, by the number of total failure cases that both methods suffers, that is, cases in which the overlapping between the ground truth and the predicted mask is 0 (see Figure SM1 E). Figure SM1 shows and example of success in which all the four algorithms are able to detect, to an acceptable extend, the carbon present in the micrographs (superior row images) and an example of failure, in which only MicrographCleaner and WPP have been able to detect the carbon.

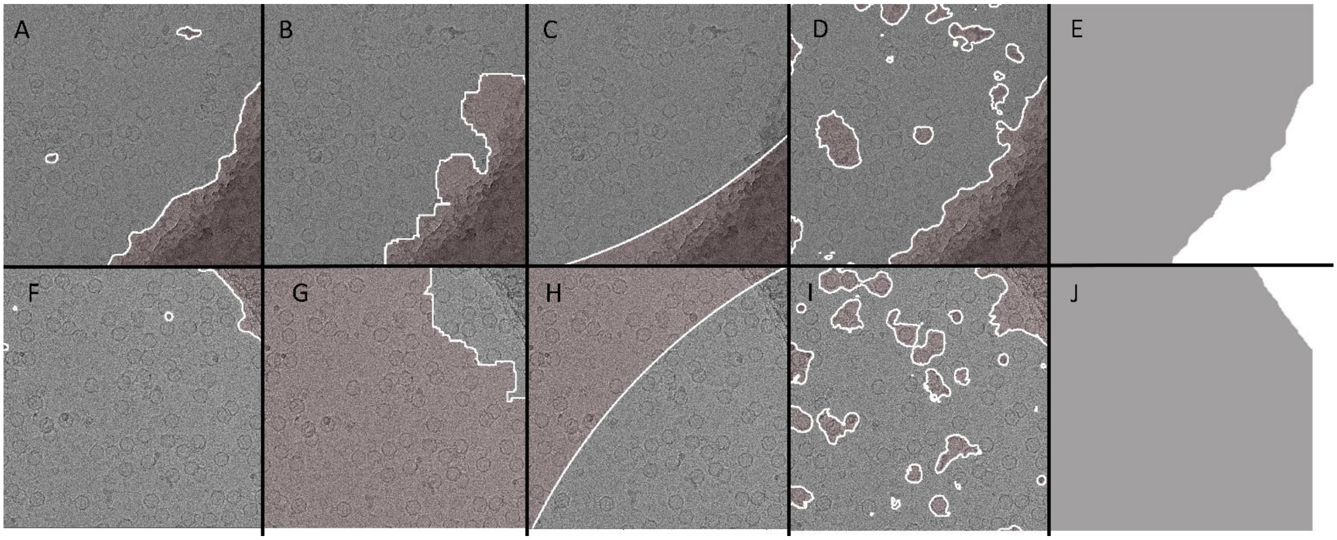


Figure SM2. Images A-D display the predicted masks (red shadowed areas) obtained with MicrographCleaner, em_hole_finder, EMHP and WPP, respectively, on the same micrograph. Image E shows the manually curated mask used to evaluate carbon detection. In this case, all algorithms have been able to obtain acceptable solutions. Images F-I display the predicted masks (red shadowed areas) obtained with MicrographCleaner, em_hole_finder, EMHP and WPP respectively, on another micrograph. Similarly, image J displays the manually curated mask for evaluation. Figure G represents an example of total failure in which not even a single carbon pixel was detected.

S7. Undesirable regions detection

Additionally, we have compared MicrographCleaner and WPP segmentation capability, measuring the mIoU of the predictions of both algorithms for all the micrographs contained in the testing set. Thus, MicrographCleaner has being able to obtain better predictions than WPP for 77.66% of the micrographs used for evaluation and, over all, it has achieved a mIoU of 0.544 as compared to the 0.331 value measured for WPP. Figure SM2 shows the predictions obtained for four different micrographs using MicrographCleaner (bottom) and WPP (top). As it can be appreciated in these results, MicrographCleaner predictions are, generally speaking, better fitted to the actual contaminants present in the micrographs. Nevertheless, WPP is able to perform better than MicrographCleaner for 22.33% of the evaluated cases (as illustrated in SM2 D) and thus, both approaches could be jointly considered to obtain better results.

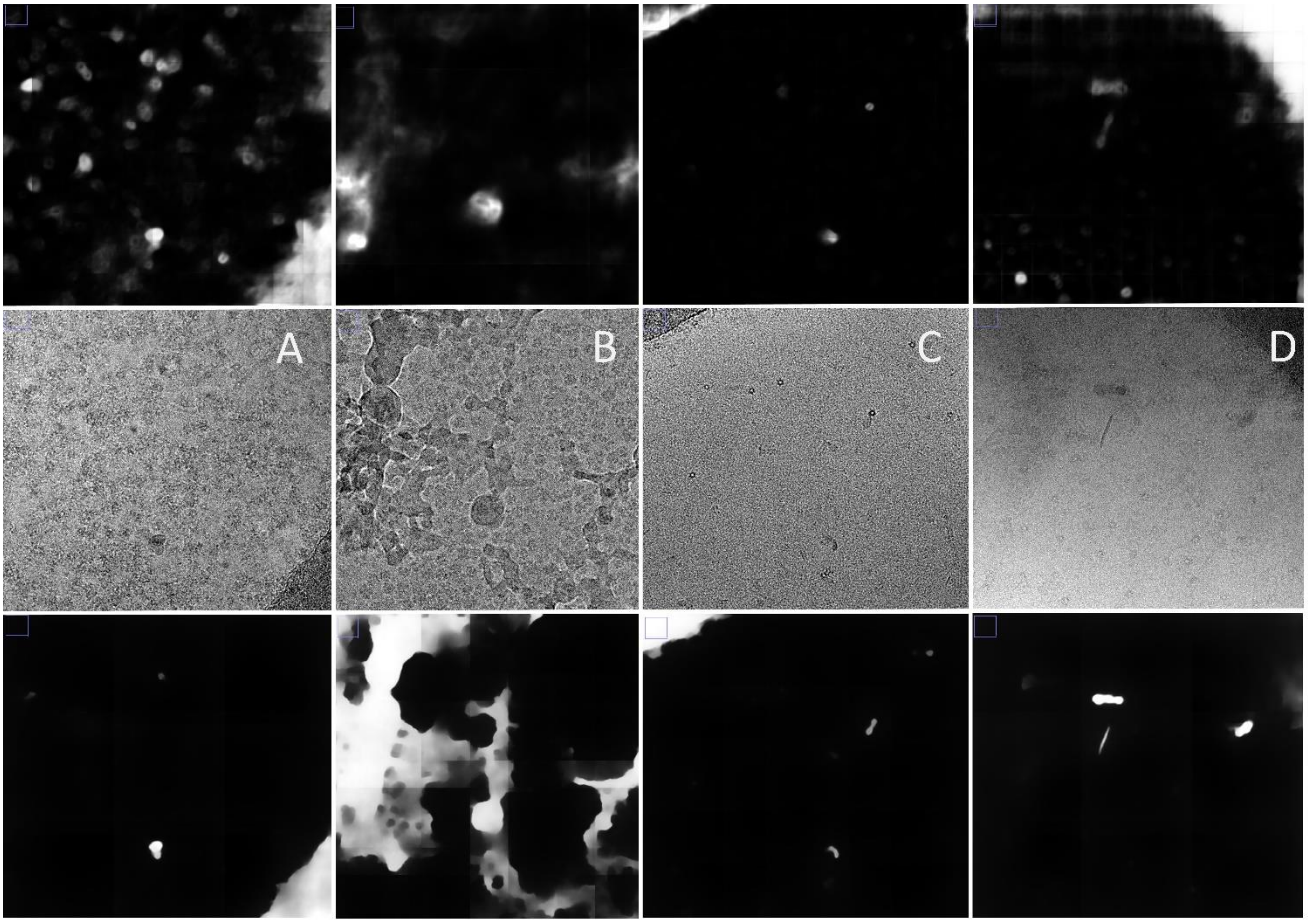


Figure SM3. Four different examples (A-D) computed using MicrographCleaner (bottom row) and the Warp particle picker (top row). Predictions for the examples A and B are clearly favorable to MicrographCleaner, whereas example C predictions are comparable for both algorithms. Example D compares favorably to Warp particle picking algorithm.

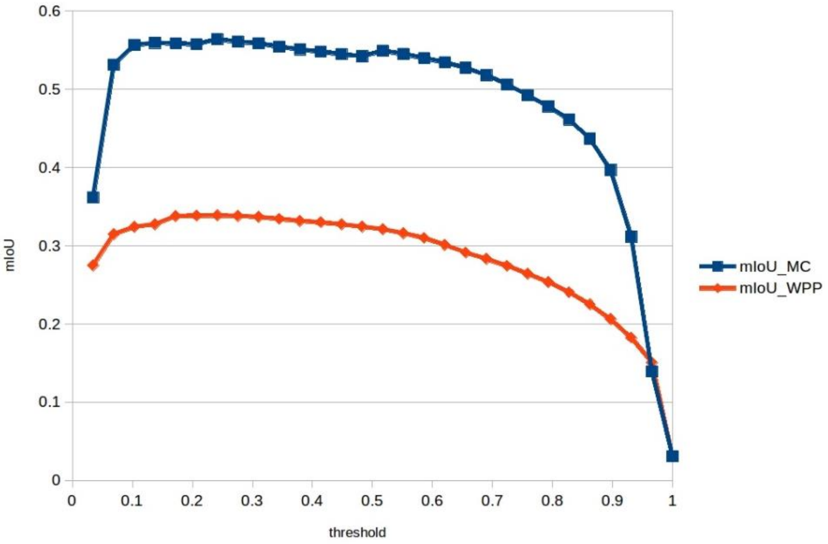


Figure SM4. Performance of MicrographCleaner and the Warp particle picker over the micrographs of the testing set for different threshold options. As it can be appreciated, both approaches are quite robust to threshold selection.

S8. Use cases

The use cases included in this publication aim to illustrate how MicrographCleaner can boost particle picking no matter the type of used algorithm. The fundament behind this claim is the “No Free Launch” theorem (Wolpert, 1996; Wolpert and Macready, 1997) which implies that there is no better algorithm for all [optimization] problems. Although, generally speaking, deep-learning particle pickers tend to outperform classical ones, these examples show some cases in which their performance is not extraordinary and it may be comparable to classical particle pickers. We also show that using an orthogonal method to the pickers (a segmentation method instead a detection/classification one) we can improve their results. This strategy of employing a blend of different algorithms to improve performance is widely used in the field of machine learning, in which real life solutions tend to be based on boosting and stacking of models and it may be a wise strategy for the cryo-EM specimens that are hard to pick.

In both examples, we have employed the Cryolo particle picker using the general model provided by the authors (version gmodel_phosnet_201910) and also, we have trained a custom model using 10 micrographs manually picked. The Topaz particle picker was also trained using the same micrographs. On the contrary, the Relion autopicker was executed using default parameters. Topaz and Cryolo solutions were manually examined in order to select an adequate global threshold. An alternative stricter threshold was also manually selected with the aim of removing most of the detected particles in the carbon area/edges as well as the contaminants picked as particles.

S.8.1. EMPIAR-10156

None of the particle pickers studied was able to perfectly avoid the carbon regions/edges at reasonable thresholds. Main Text Figure 2 is one of the many examples in which a non-negligible number of false positive particles are selected independently of the algorithm and the threshold. Generally speaking, all the particle pickers are able to select most of the true positive particles but the carbon areas are not being avoided by the Relion particle picker and Cryolo is also missing some of them. On the contrary, the Topaz algorithm is able to better avoid the carbon regions, although it is strongly attracted by the carbon edges, which are equally if not more dangerous.

When comparing the total number of picked particles (Table S.8.1), it can be appreciated that the Cryolo particle picker solution is the one which is less modified by MicrographCleaner and thus it seems to be the cleaner set of particles. On the contrary, the Relion autopicker is the one that is more affected by MicrographCleaner, which supports the idea that traditional particle pickers are more affected by false positives than deep-learning-based solutions. However, after MicrographCleaning usage, all the different approaches contain a similar number of particles, which suggests that the quality of the datasets might be comparable. This should not be so surprising as the picking of the particles in this example is not complicate leaving aside the problem of the carbon areas/edges.

Table S.8.1. Number of picked particles using different algorithms and thresholds.

Algorithm	Threshold	#particles
Cryolo general model	0.1	13916
Cryolo general model + MicrographCleaner	0.1	12620
Cryolo general model	0.3	9801
Cryolo trained	0.4	14262
Cryolo trained+ MicrographCleaner	0.4	13211
Cryolo trained	0.5	10013
Topaz	2	17981

Topaz + MicrographCleaner	2	15321
Topaz	4	13592
Relion auto	Default	19616
Relion auto + MicrographCleaner	Default	14852

S.8.2. EMPIAR-10265

In this example, and contrary to the previous one, the Cryolo solutions, specially the one obtained from the manually trained model is substantially better that the others. Yet, it is still not perfect as illustrated by the small contaminants displayed in Main Text Figure 3 and 4 and by the fact that MicrographCleaner is still able to remove more than 2000 false positive particles from them. Hence, and likewise the previous case, if most of the contaminants that can visually be identified are discarded using a higher threshold, there is still an important number of true positive particles that are also removed, as it can be derived from the numbers exposed in Table S.8.2. However, these effects are less pronounced that for the other methods.

Regarding the Topaz and the Relion solutions, it is important to notice that in this case, the number of discarded particles, 6% and 15% respectively, is of importance. It might seem surprising that a relatively small set of discarded images make a difference in the quality of downstream analysis of cryEM particles, but we should note that there is a very profound statistical difference between adding or removing "randomly" a set of particles, and adding or removing a totally biased set of particles (those in bad micrograph areas). This is an area in which we anticipate substantial developments in the cryoEM area be coming: Being aware of bias in the calculation of cryoEM maps, and introducing the appropriate state of correction, as we are doing here wih MicrographCleaner.

Table S.8.2. Number of picked particles using different algorithms and thresholds.

Algorithm	Threshold	#particles
Cryolo general model	0.1	168820
Cryolo general model + MicrographCleaner	0.1	165070
Cryolo general model	0.3	97085
Cryolo trained	0.5	203131
Cryolo trained+ MicrographCleaner	0.5	198383
Cryolo trained	0.7	104121
Topaz	3	140204
Topaz + MicrographCleaner	3	131572
Topaz	5	63843
Relion auto	Default	177029
Relion auto + MicrographCleaner	Default	149720

S9. MicrographCleaner complementes 2D-classification

In order to study the effectiveness of MicrographCleaner when 2D-classification is considered, we have run Relion 2D-classification (Kimanius et al., 2016; Scheres, 2012) algorithm on the sets of particles that were presented in Main Text section 3.3.1 and Supplementary Material S8.1. Then, we have ruled out all the particles that belonged to "bad classes" and we have counted the number of particles that remained. Additionally, we have computed the intersection between the sets of particles that remained

after 2D-classification processing and the particles removed by MicrographCleaner. As the number of particles discarded by MicrographCleaner that are not removed by 2D-classification was large (19% to 29%), we performed a second round of 2D-classification in order to clean further the sets of particles. Yet, the number of particles ruled out by MicrographCleaner that survived after two rounds of classification, as illustrated in table S.9. is important, between 10% to 19%.
The next pages, we present the same example than Main Text Figure 5 when the remaining particle pickers are considered instead. In all them, the situation is similar to Main Text Figure 5, as MicrographCleaner is able to remove the particles that lay on the carbon/edges while the 2D-classification algorithm is not able to fully remove them.

Table S.9. Number of remaining particles after 2D classification and MicrocraphCleaner in *EMPIAR-10156*

Algorithm	Threshold	initial	2D-classes$_1$	micClean	2D-classes$_1$ & micClean removed	2D-classes$_2$	2D-classes$_1$ & micClean removed
Cryolo general	0.1	13916	11045	12620	377	6776	205
Cryolo trained	0.4	14626	8873	13211	329	7021	273
Topaz	2	17981	12046	15321	498	6876	253
Relion auto	NA	19616	12554	14852	1055	7941	608

NOTE: initial: Number of particles that were selected by the particle picker; 2D-classes$_1$: Number of particles that survived first step 2D-classifaction; micClean: Number of particles that survived to MicrographCleaner execution; 2D-classes$_1$ & micClean removed: number of particles that survived to the first step of 2D-classification but are false positives according MicrographC

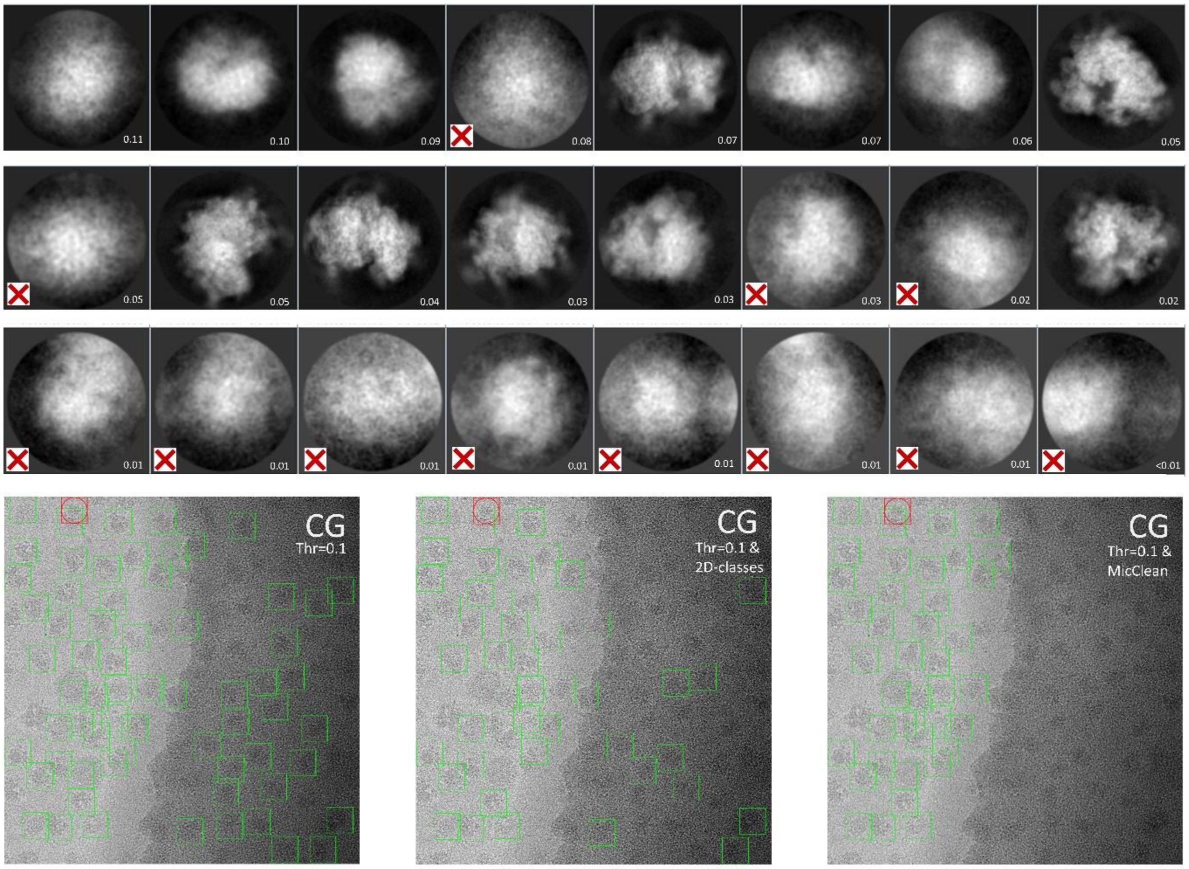


Figure SM5. MicrographCleaner complements 2D-classification. Top: 2D averages obtained from the set of particles collected with Cryolo general model on EMPIAR-10265 dataset. Bottom, from left to right: Particles originally picked by Cryolo and used as input for 2D-classification; particles that correspond to the 2D-classes that were not discarded (not marked with red cross); particles not discarded by MicrographCleaner. It can be appreciated that MicrographCleaner removed all particles picked in carbon but 2D-classification did not.

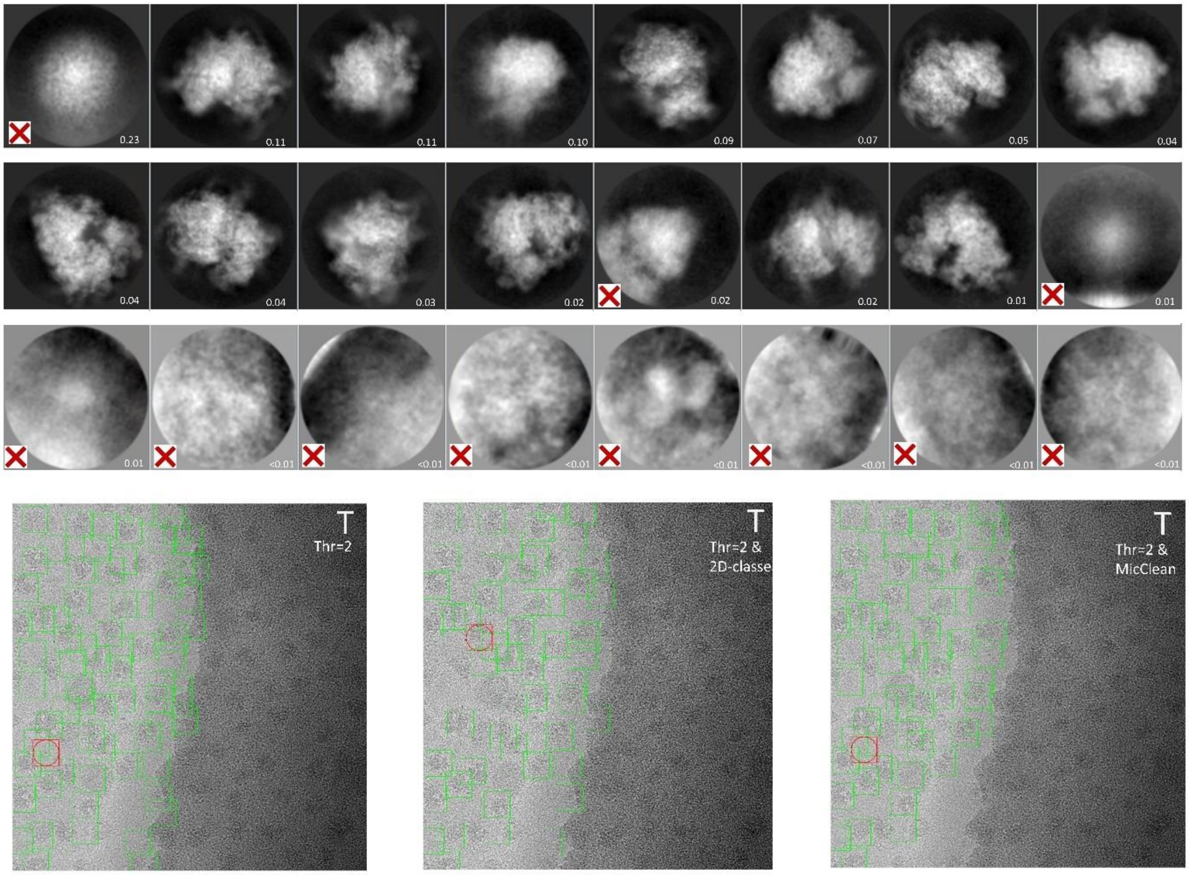


Figure SM6. MicrographCleaner complements 2D-classification. Top: 2D averages obtained from the set of particles collected with Topaz on EMPIAR-10265 dataset. Bottom, from left to right: Particles originally picked by Topaz and used as input for 2D-classification; particles that correspond to the 2D-classes that were not discarded (not marked with red cross); particles not discarded by MicrographCleaner. It can be appreciated that MicrographCleaner removed all particles picked in carbon but 2D-classification did not

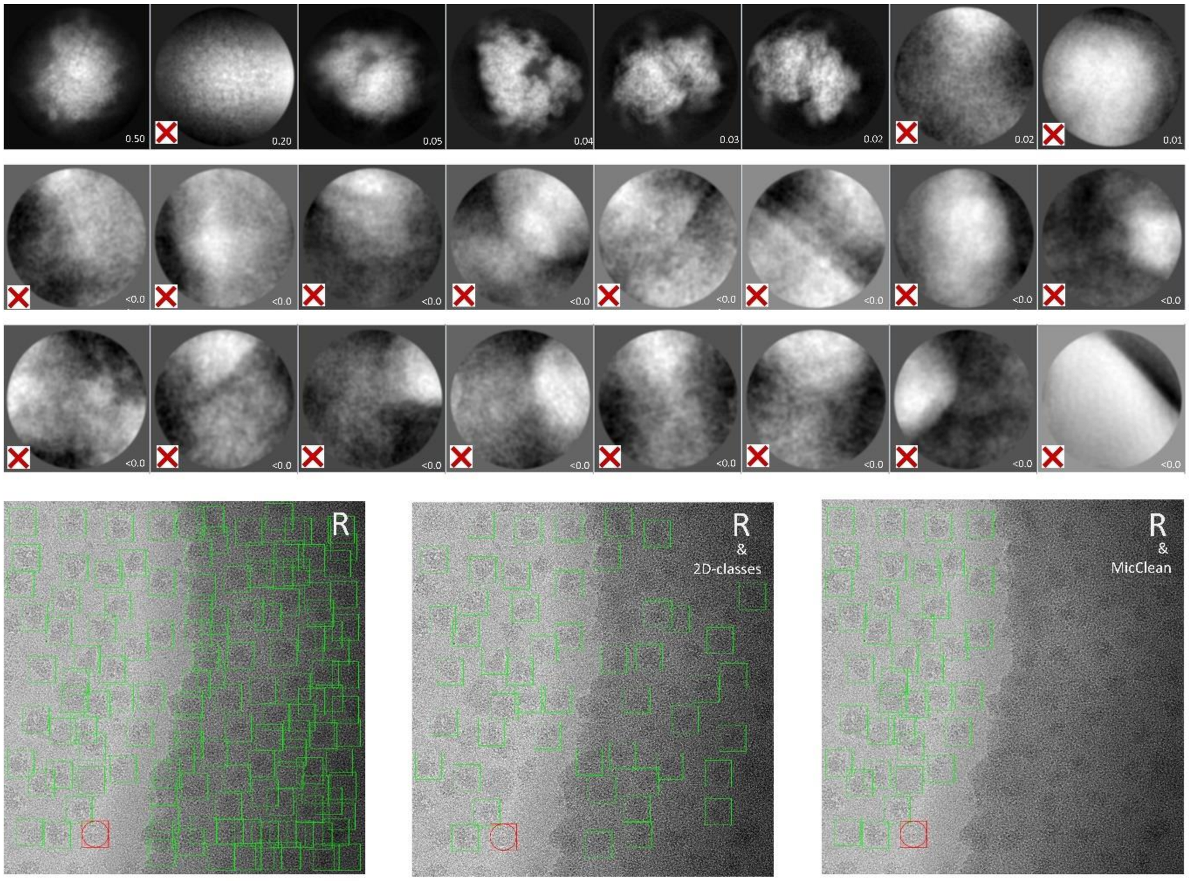


Figure SM7. MicrographCleaner complements 2D-classification. Top: 2D averages obtained from the set of particles collected with Relion autopicker on EMPIAR-10265 dataset. Bottom, from left to right: Particles originally picked by Relion and used as input for 2D-classification; particles that correspond to the 2D-classes that were not discarded (not marked with red cross); particles not discarded by MicrographCleaner. It can be appreciated that MicrographCleaner removed all particles picked in carbon but 2D-classification did not

S10. Usage guide

A complete installation and command line execution guide can be found in https://github.com/rsanchezgarc/micrograph_cleaner_em/tree/master.

In order to compute masks from micrographs, just two commands are needed to be employed once the package is installed.

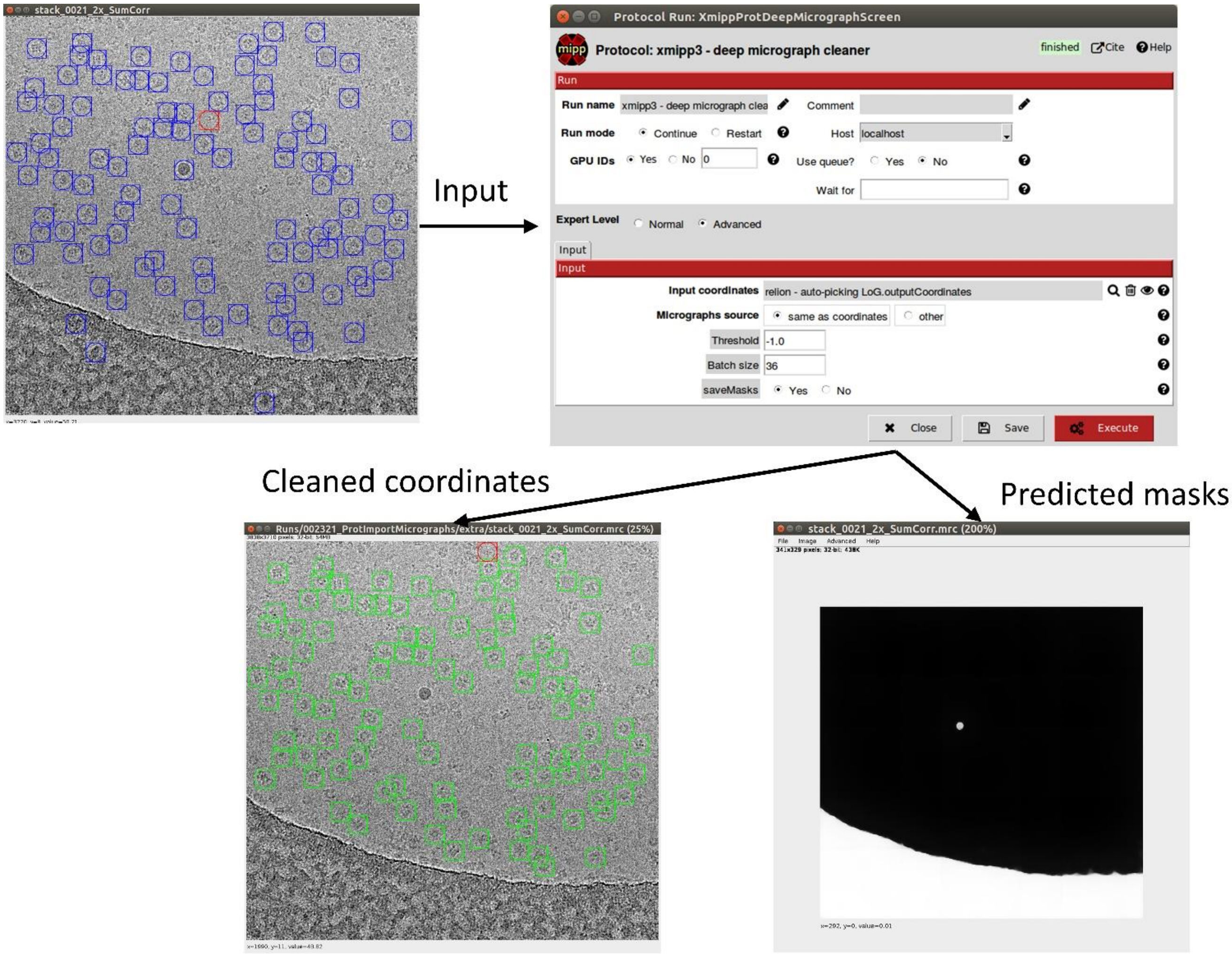
First, with the aim of downloading an updated version of the deep learning model, the following command should be executed:

```
cleanMics --download
```

Then, in order to compute masks for a given set of micrographs, the following command should be executed:

```
cleanMics -b $BOX_SIXE -i /path/to/micrographs/ --predictedMaskDir path/to/store/masks
```

Additionally, MicrographCleaner can also be executed within the cryo-EM framework Scipion (de la Rosa-Trevín et al., 2016) through the protocol deepMicrographScreen. An illustration of the form is depicted in Figure SM2.



Protocol Run: XmippProtDeepMicrographScreen

Protocol: xmipp3 - deep micrograph cleaner

Run

Run name: xmipp3 - deep micrograph clea
Comment:
Run mode: ● Continue ○ Restart
Host: localhost
GPU IDs: ● Yes ○ No 0
Use queue? ○ Yes ● No
Wait for:

Expert Level: ○ Normal ● Advanced

Input

Input coordinates: relion - auto-picking LoG.outputCoordinates
Micrographs source: ● same as coordinates ○ other
Threshold: -1.0
Batch size: 36
saveMasks: ● Yes ○ No

Figure SM8. MicrographCleaner GUI illustration within Scipion.

As it can be appreciated in SM2, the only two parameters required for the execution of MicrographCleaner scipion protocol are the batch size, that only has impact over the GPU computation efficiency and the threshold, that can be manually set or leaved as default.

Additionally, MicrographCleaner can be programmatically executed, and thus, integrated with other tools, using a simple API documented in https://github.com/rsanchezgarc/micrograph_cleaner_em/tree/master.

References

Bar, Y., Diamant, I., Wolf, L., Greenspan, H., 2015. Deep learning with non-medical training used for chest pathology identification, in: Hadjiiski, L.M., Tourassi, G.D. (Eds.), Medical Imaging 2015: Computer-Aided Diagnosis. p. 94140V. https://doi.org/10.1117/12.2083124

de la Rosa-Trevín, J.M., Quintana, A., del Cano, L., Zaldívar, A., Foche, I., Gutiérrez, J., Gómez-Blanco, J., Burguet-Castell, J., Cuenca-Alba, J., Abrishami, V., Vargas, J., Otón, J., Sharov, G., Vilas, J.L., Navas, J., Conesa, P., Kazemi, M., Marabini, R., Sorzano, C.O.S., Carazo, J.M., 2016. Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy. J. Struct. Biol. 195, 93–99. https://doi.org/10.1016/j.jsb.2016.04.010

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., Dovzhenko, A., Tietz, O., Dal Bosco, C., Walsh, S., Saltukoglu, D., Tay, T.L., Prinz, M., Palme, K., Simons, M., Diester, I., Brox, T., Ronneberger, O., 2019. U-Net: deep learning for cell counting, detection, and morphometry. Nat. Methods 16, 67–70. https://doi.org/10.1038/s41592-018-0261-2

Johnson, J., Alahi, A., Fei-Fei, L., 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. https://doi.org/10.1007/978-3-319-46475-6_43

Kimanius, D., Forsberg, B.O., Scheres, S.H., Lindahl, E., 2016. Accelerated cryo-EM structure determination with parallelisation using GPUs in RELION-2. Elife 5, e18722. https://doi.org/10.7554/eLife.18722

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention-MICCAI. pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Scheres, S.H.W., 2012. RELION: Implementation of a Bayesian approach to cryo-EM structure determination. J. Struct. Biol. 180, 519–530. https://doi.org/10.1016/j.jsb.2012.09.006

Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition.

Wolpert, D.H., 1996. The Lack of A Priori Distinctions Between Learning Algorithms. Neural Comput. 8, 1341–1390. https://doi.org/10.1162/neco.1996.8.7.1341

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1, 67–82. https://doi.org/10.1109/4235.585893

Wu, S., Zhong, S., Liu, Y., 2017. Deep residual learning for image steganalysis. Multimed. Tools Appl. 1–17. https://doi.org/10.1007/s11042-017-4440-4