EPySeg: a coding-free solution for automated segmentation of epithelia using deep learning

Benoit Aigouy[1],*, Claudio Cortes[1], Shanda Liu[2] and Benjamin Prud'Homme[1],*

ABSTRACT

Epithelia are dynamic tissues that self-remodel during their development. During morphogenesis, the tissue-scale organization of epithelia is obtained through a sum of individual contributions of the cells constituting the tissue. Therefore, understanding any morphogenetic event first requires a thorough segmentation of its constituent cells. This task, however, usually involves extensive manual correction, even with semi-automated tools. Here, we present EPySeg, an open-source, coding-free software that uses deep learning to segment membrane-stained epithelial tissues automatically and very efficiently. EPySeg, which comes with a straightforward graphical user interface, can be used as a Python package on a local computer, or on the cloud via Google Colab for users not equipped with deep-learning compatible hardware. By substantially reducing human input in image segmentation, EPySeg accelerates and improves the characterization of epithelial tissues for all developmental biologists.

KEY WORDS: Computer vision, Deep learning, Epithelia, Quantitative biology, Segmentation, Software

INTRODUCTION

Epithelia are dynamic tissues undergoing dramatic shape changes throughout their development. A prerequisite for understanding these morphogenetic events is the thorough segmentation of cells constituting the tissue. To this aim, numerous semi-automated methods have been developed (Aigouy et al., 2016; Farrell et al., 2017; Cilla et al., 2015; Heller et al., 2016), but they require time-consuming manual correction to achieve optimal segmentation.

Over the past few years, deep learning, and more particularly convolutional neural networks (CNNs), has reshaped the computer vision field. In particular, deep-learning approaches should be beneficial for image segmentation because they could, in theory, reduce or even eliminate the need for end-user correction of the segmentation output. The advent of simple programming frameworks, such as Keras (https://github.com/fchollet/keras) and TensorFlow (Abadi et al., 2016 preprint), has made deep learning accessible to most developers but still excludes people lacking coding skills, preventing deep learning from being broadly adopted by the scientific community. A few attempts to bring CNNs to well-known image processing frameworks such as ImageJ or FIJI exist (Schmidt et al.; Weigert et al., 2018; Gómez-de-Mariscal et al., 2019 preprint; Schindelin et al., 2012; Schneider et al., 2012), but they require an up-to-date and adequately configured computer. More importantly, most often those powerful, yet very poorly generalizable, CNNs need to be trained *de novo* on user-provided data to work efficiently. Unfortunately, in most cases, such training cannot be done directly in FIJI or ImageJ and requires coding expertise. So far, little effort has been made to facilitate CNN training and use by regular users (von Chamier et al., 2020 preprint; Buchholz et al., 2020 preprint).

To address all these limitations, we present EPySeg, a coding-free solution to efficiently segment raw images of epithelial tissues, using a pre-trained neural network. Furthermore, EPySeg comes with a complete and straightforward graphical user interface (GUI), allowing users that are curious about deep learning, as well as more advanced users, to build and train custom networks to achieve any segmentation paradigm of interest. EPySeg is available at https://github.com/baigouy/EPySeg, and a minimal version can also be used on Google Colab (https://github.com/baigouy/notebooks) for users equipped with low-end graphics cards.

RESULTS AND DISCUSSION

In this study, we set out to develop a software that uses deep learning to automate the time-consuming segmentation of 2D epithelial tissue images. We selected LinkNet architectures, because they are known to perform well at image segmentation tasks (Chaurasia and Culurciello, 2017; also see Materials and Methods). Our network was trained on a large number of images of very divergent fly epithelia acquired using several microscopy setups (see Materials and Methods) to allow our segmentation paradigm to be robust and able to segment a broad range of epithelial tissues. Cell segmentation was generated using the watershed algorithm (Vincent and Soille, 1991), followed by careful manual curation to remove errors (see Materials and Methods). In EPySeg, this watershed segmentation was converted into a set of five watershed-like segmentations and two watershed seeds (see Materials and Methods) that the EPySeg neural network is trained to generate when given an input epithelial image (Fig. 1). The seven outputs generated by the neural network are combined into a single watershed mask upon averaging and thresholding (Fig. 1). This mask then corresponds to an optimized watershed-like segmentation of the tissue.

EPySeg, although trained exclusively on fly epithelia, can efficiently segment evolutionarily distant 2D epithelial tissues imaged with different optics (Fig. 2; Table S1). We compared our software to Cellpose, the only software available to date that can segment cells without the need for prior model training (Stringer et al., 2020 preprint). On average, EPySeg outperformed Cellpose on epithelia in two ways: its approximation of the cell outline was more precise than that of Cellpose (Fig. S1, Table S1), and it missed fewer cells (Fig. 2; Fig. S2; Table S1). We note, however, that unlike Cellpose, EPySeg was not able to segment cells in culture

[1]Aix Marseille University, CNRS, IBDM, 13288 Marseille, France. [2]Max Planck Institute for Plant Breeding Research, 50829 Köln, Germany.

*Authors for correspondence (benoit.aigouy@univ-amu.fr; benjamin.prudhomme@univ-amu.fr)

B.A., 0000-0002-2053-4494; B.P., 0000-0003-2889-9081

Handling Editor: James Briscoe
Received 30 June 2020; Accepted 17 November 2020

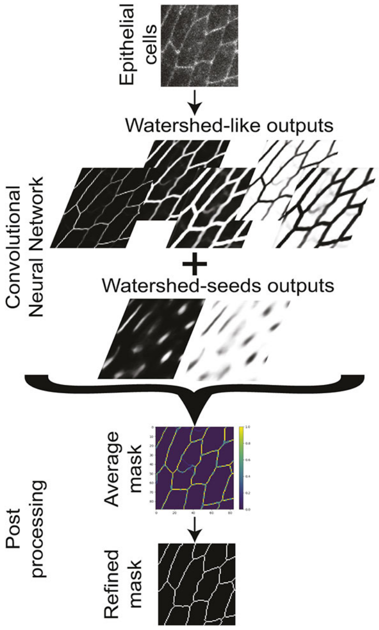

Fig. 1. EPySeg segmentation pipeline. An unseen image of cells labelled with a membrane marker is provided to the EPySeg pre-trained neural network. EPySeg produces seven outputs from it: five of them are watershed-like outputs, while the remaining two are watershed seeds. Those seven outputs are used to generate seven watershed masks. Upon thresholding the average of these seven masks, we obtain a refined mask.

(Table S1) and is likely to be less efficient at segmenting non-cellular objects than Cellpose, because it was not trained to accomplish such tasks.

Finally, to make our epithelial segmentation tool easily accessible to a broad audience, we created a GUI and detailed documentation for its use (https://github.com/baigouy/EPySeg). This interface allows for building, training and running CNNs. It is built in such a way that non-expert users can rely on the default settings to easily train a network and gain knowledge on using deep learning for image analysis, whereas advanced users can visually fine-tune parameters to achieve optimal results. Because the majority of computers available in research labs are not deep learning-ready, we also provide a minimal user interface to run EPySeg online, in Google Colab, granting a broader audience access to deep-learning approaches (https://github.com/baigouy/notebooks).

MATERIALS AND METHODS

Recommended equipment

The EPySeg CNN was trained on a Dell Precision 7820 with 64 GB RAM, equipped with a Nvidia GeForce RTX 2070 graphic card with 8 GB RAM. Most training lasted less than 12 h. We could also successfully train and run our CNN on Google Colab, hereby providing a good alternative for users with deep learning-incompatible systems.

Data

The EPySeg CNN was trained on several *Drosophila* epithelia stained with E-cadherin:GFP that largely diverged from one another. One training set consisted of tissue from embryonic stages, where E-cadherin staining in epithelia appeared dotted (Tepass and Hartenstein, 1994; Truong Quang et al., 2013; Cavey et al., 2008) and the boundary-to-cytoplasm signal ratio was low. Another training set used pupal wing tissue, where E-cadherin staining appeared continuous and presented a higher boundary-to-cytoplasm ratio, except for stretched cells. Finally, our third training set contained images of the fly abdomen, including giant, polyploid, larval cells and tiny histoblast nest cells (Madhavan and Madhavan, 1980), in order to have a network that can segment cells without a size bias. Input images were either max- or stack-focuser projections (using Stack Focuser ImageJ plugin; https://imagej.nih.gov/ij/plugins/stack-focuser.html) of all or part of confocal *z*-stacks of epithelial tissues. Segmented cell outlines, serving as ground truth for training the network and for evaluating the segmentation quality, were generated using the watershed algorithm of Tissue Analyzer (Aigouy et al., 2016; Vincent and Soille, 1991). Importantly, we paid a lot of attention to the quality of the segmentation masks fed to the CNN, and we cropped out regions where segmentation quality was poor as well as regions that were not segmented (e.g. cells adjacent to the tissue of interest) in order not to perturb the learning process. For training the model, every watershed segmentation mask was used to generate seven images, the first image was the curated watershed mask itself, the second and third were the same watershed mask after one or two binary dilations, respectively. The fourth and fifth images were the negatives of the second and third images, respectively (akin to a non-cellular background). The sixth image was generated to contain a single seed (group of pixels) per cell, scaled by the cell size. The seventh image was the negative of the sixth image. The model was asked to reproduce these seven outputs for any given input. Two of the three training datasets were acquired on regular Leica or Zeiss confocal microscopes (Leica SP2 and LSM 510, respectively), whereas the third dataset was acquired on a spinning-disc microscope (Roper) to expand the breadth of optics used. The plant sample used for testing our segmentation is the fourth leaf of a 7 days after germination transgenic *Arabidopsis thaliana* labeled with UBQ10::acyl:tdTomato (modified from the construct by Willis et al., 2016). The vertebrate test sample is a ventral view of the dorsal pericardial wall epithelium stained with Phalloidin. Both test samples were acquired using a Leica SP8 upright confocal microscope and a Zeiss LSM 780, respectively. The fly wing and abdominal test samples were acquired using an Olympus FV-1000 confocal microscope. The fly head sample was acquired using an LSM 510 microscope.

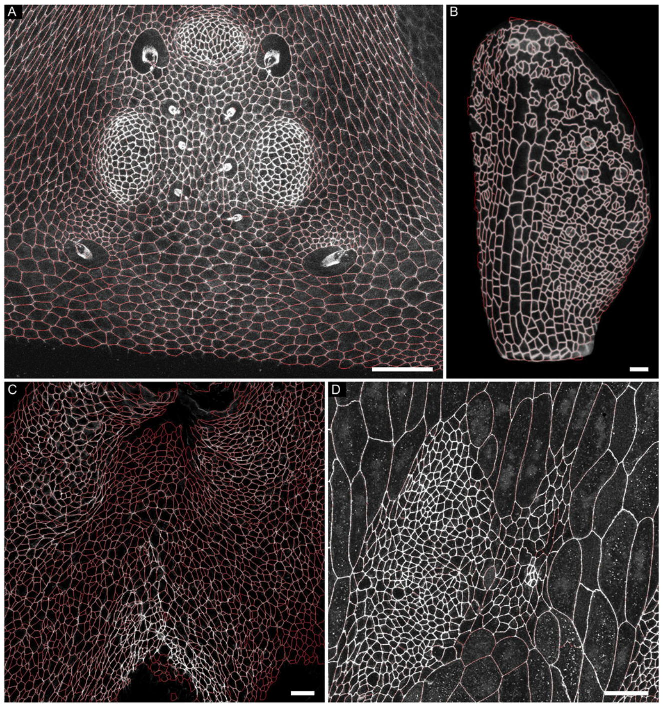

Fig. 2. EPySeg segmentation of unseen epithelium images. (A-D) EPySeg segmentation (red) overlaid on unseen images. (A) Segmentation of the *Drosophila* head epithelium, including ocelli, labelled with E-cadherin:GFP (greyscale). (B) Segmentation of the fourth leaf of a plasma membrane-labelled *Arabidopsis thaliana* plant at 7 days post germination (greyscale, UBQ10::acyl:tdTomato). (C) Segmentation of Phalloidin-labelled (greyscale) vertebrate dorsal pericardial wall epithelium (Cortes et al., 2018; Francou et al., 2017). (D) Segmentation of the *Drosophila* abdominal region surrounding a histoblast nest, labelled with E-cadherin:GFP (greyscale). Scale bars: 25 μm.

Data augmentation

To further increase the size of our training set for deep learning (images and cells) and to prevent the neural network from overfitting, we used data augmentation: we randomly applied the same transformation (rotation, translation, magnification, flip, …) within a given range to both the input and output images. Our data augmentation algorithm currently supports 2D and 3D images (only 2D images were used in this study).

CNN building and training

Our CNN was generated using the segmentation_models library from (https://github.com/qubvel/segmentation_models) and relies on TensorFlow and Keras. We used a LinkNet (Chaurasia and Culurciello, 2017) architecture with a VGG16 encoder (Simonyan and Zisserman, 2015, preprint). We found that this encoder, known to perform well at classification tasks, was also very efficient at segmenting epithelia. Of note, the detailed model architecture is available in the log window of the software upon loading the model. The network was trained for 300 epochs on the complete training set at every epoch. We used Adam (Kingma and Ba, 2017 preprint) as the optimizer with an initial 10^{-3} learning rate for the first 150 epochs and a 10^{-4} learning rate for the next 150 epochs. The network was trained with a batch size of 24 images and a tile size of 256 pixels in width and height. We chose the intersection over union (IoU), also called the Jaccard index, for the loss function, because it is particularly well suited to evaluate differences between binary images (Rahman and Wang, 2016).

Segmentation quantification

To measure the accuracy of cell segmentation (i.e. quality of the cell mask) we used the SEG score (Ulman et al., 2017). Briefly, this measure evaluates the average amount of overlap between the reference segmentation and the corresponding neural network-generated segmentation. As a measure for segmentation quality (i.e. an evaluation of over- and under-segmentation), we used the average precision score (AP) defined as $AP=TP/(TP+FP+FN)$, where FP corresponds to over-segmented cells and FN correspond to under-segmented cells. TP, the properly segmented cells, are defined as segmented cells having an IoU score ≥0.7 when compared with the corresponding ground truth cell.

Software

The software was entirely coded in Python 3. The graphical user interface was made with PyQT5 (Riverbank). The source code of our tool along with installation instructions can be found at https://github.com/baigouy/EPySeg.

Ethical approval

Animal experiments were carried out in agreement with national and European laws and approved by the Ethics Committee for Animal Experimentation of Marseille and the French Ministry for National Education, Higher Education and Research.

Acknowledgements

We would like to thank Robert Kelly and Miltos Tsiantis for sharing unpublished images with us.

Competing interests

The authors declare no competing or financial interests.

Author contributions

Conceptualization: B.A., B.P.; Software: B.A.; Resources: C.C., S.L., B.P.; Writing - original draft: B.A., B.P.; Writing - review & editing: B.A., C.C., S.L., B.P.; Supervision: B.P.; Funding acquisition: B.P.

Funding

S.L. was supported by a Max Planck Core grant to Miltos Tsiantis. C.C. was supported by a grant from the Fondation Leducq to Robert Kelly (Transatlantic Network of Excellence 15CVD01). The project was supported by the Centre National de la Recherche Scientifique, the France-BioImaging/PICsL infrastructure (ANR-10-INSB-04-01) and the European Research Council under the European Union's Seventh Framework Programme [(FP/2007-2013)/ERC Grant Agreement 615789 to B.P.]. Deposited in PMC for immediate release.

Supplementary information

Supplementary information available online at https://dev.biologists.org/lookup/doi/10.1242/dev.194589.supplemental

Peer review history

The peer review history is available online at https://dev.biologists.org/lookup/doi/10.1242/dev.194589.reviewer-comments.pdf

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M. et al. (2016). TensorFlow: large-scale machine learning on heterogeneous distributed systems. *arXiv* preprint, *arXiv*:1603.04467 [cs.DC].

Aigouy, B., Umetsu, D. and Eaton, S. (2016). Segmentation and Quantitative Analysis of Epithelial Tissues. In *Drosophila: Methods and Protocols* (ed. C. Dahmann), pp. 227-239. New York: Springer.

Buchholz, T.-O., Prakash, M., Krull, A. and Jug, F. (2020). DenoiSeg: joint denoising and segmentation. *arXiv* preprint, *arXiv*:2005.02987 [cs.CV].

Cavey, M., Rauzi, M., Lenne, P.-F. and Lecuit, T. (2008). A two-tiered mechanism for stabilization and immobilization of E-cadherin. *Nature* **453**, 751-756. doi:10.1038/nature06953

Chaurasia, A. and Culurciello, E. (2017). LinkNet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1-4. IEEE. doi:10.1109/VCIP.2017.8305148

Cilla, R., Mechery, V., Hernandez De Madrid, B., Del Signore, S., Dotu, I. and Hatini, V. (2015). Segmentation and tracking of adherens junctions in 3D for the analysis of epithelial tissue morphogenesis. *PLoS Comput. Biol.* **11**, e1004124. doi:10.1371/journal.pcbi.1004124

Cortes, C., Francou, A., De Bono, C. and Kelly, R. G. (2018). Epithelial properties of the second heart field. *Circ. Res.* **122**, 142-154. doi:10.1161/circresaha.117.310838

Farrell, D. L., Weitz, O., Magnasco, M. O. and Zallen, J. A. (2017). SEGGA: a toolset for rapid automated analysis of epithelial cell polarity and dynamics. *Development* **144**, 1725-1734. doi:10.1242/dev.146837

Francou, A., De Bono, C. and Kelly, R. G. (2017). Epithelial tension in the second heart field promotes mouse heart tube elongation. *Nat. Commun.* **8**, 14770. doi:10.1038/ncomms14770

Gómez-De-Mariscal, E., García-López-de-Haro, C., Donati, L., Unser, M., Muñoz-Barrutia, A. and Sage, D. (2019). DeepImageJ: A user-friendly plugin to run deep learning models in ImageJ. *bioRxiv*. doi:10.1101/799270

Heller, D., Hoppe, A., Restrepo, S., Gatti, L., Tournier, A. L., Tapon, N., Basler, K. and Mao, Y. (2016). EpiTools: an open-source image analysis toolkit for quantifying epithelial growth dynamics. *Dev. Cell* **36**, 103-116. doi:10.1016/j.devcel.2015.12.012

Kingma, D. P. and Ba, J. (2017). Adam: a method for stochastic optimization. *arXiv* preprint, *arXiv*:1412.6980 [cs.LG].

Madhavan, M. M. and Madhavan, K. (1980). Morphogenesis of the epidermis of adult abdomen of Drosophila. *J. Embryol. Exp. Morphol.* 60, 1-31.

Rahman, M. A. and Wang, Y. (2016). Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. In *Advances in Visual Computing*, Vol. 10072 (ed. G. Bebis et al.), pp. 234-244. Springer International Publishing.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B. et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676-682. doi:10.1038/nmeth.2019

Schmidt, U., Weigert, M., Broaddus, C. and Myers, G. (2018). Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018* (ed. A. Frangi, J. Schnabel, C. Davatzikos, C. Alberola-López and G. Fichtinger), pp. 265-273. Cham: Springer International Publishing.

Schneider, C. A., Rasband, W. S. and Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **9**, 671-675. doi:10.1038/nmeth.2089

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *arXiv* preprint, arXiv:1409.1556v6 [cs.CV]

Stringer, C., Wang, T., Michaelos, M. and Pachitariu, M. (2020). Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods*. https://doi.org/10.1038/s41592-020-01018-x

Tepass, U. and Hartenstein, V. (1994). The development of cellular junctions in the drosophila embryo. *Dev. Biol.* **161**, 563-596. doi:10.1006/dbio.1994.1054

Truong Quang, B.-A., Mani, M., Markova, O., Lecuit, T. and Lenne, P.-F. (2013). Principles of E-cadherin supramolecular organization in vivo. *Curr. Biol.* **23**, 2197-2207. doi:10.1016/j.cub.2013.09.015

Ulman, V., Maška, M., Magnusson, K. E. G., Ronneberger, O., Haubold, C., Harder, N., Matula, P., Matula, P., Svoboda, D., Radojevic, M. et al. (2017). An objective comparison of cell-tracking algorithms. *Nat. Methods* **14**, 1141-1152. doi:10.1038/nmeth.4473

Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 583-598. doi:10.1109/34.87344

von Chamier, L., Laine, R. F., Jukkala, J., Spahn, C., Krentzel, D., Nehme, E., Lerche, M., Hernández-Pérez, S., Mattila, P. K., Karinou, E., et al. (2020). ZeroCostDL4Mic: an open platform to use Deep-Learning in microscopy. *bioRxiv* doi:10.1101/2020.03.20.000133

Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S. et al. (2018). Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nat. Methods* **15**, 1090-1097. doi:10.1038/s41592-018-0216-7

Willis, L., Refahi, Y., Wightman, R., Landrein, B., Teles, J., Huang, K. C., Meyerowitz, E. M. and Jönsson, H. (2016). Cell size and growth regulation in the Arabidopsis thaliana apical stem cell niche. *Proc. Natl Acad. Sci. USA* **113**, E8238-E8246. doi:10.1073/pnas.1616768113

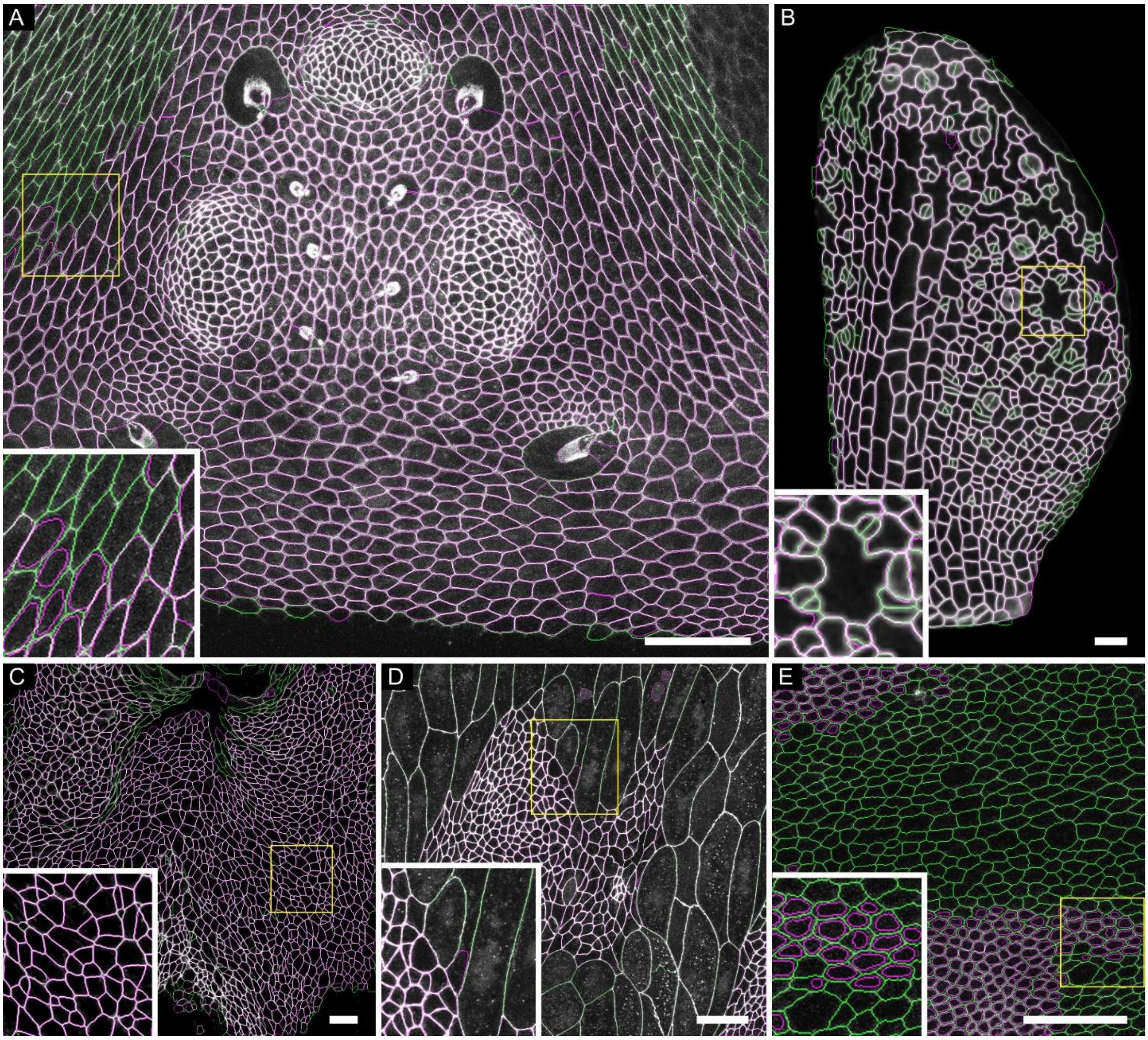

Figure S1: EPySeg segmentation of cell outlines on various epithelia

Comparison of cell outlines generated by EPySeg (green) and Cellpose (magenta) overlaid over original images. **(A)** E-cadherin:GFP staining of the fly head epithelium including fly ocelli. **(B)** Image of the fourth leaf of a seven days after germination plant. **(C)** Vertebrate epithelial cells. The image is a ventral view of the apical surface of splanchnic mesodermal cardiac progenitor cells in the dorsal pericardial wall of a mouse embryo at embryonic day 9.5, after removal of the heart tube. **(D)** Fly abdomen showing a histoblast nest surrounded by giant larval cells. **(E)** Image of the central region of drosophila wing epithelium. Insets show close-up views of the segmentations. Of note, EPySeg approximation of the cell outline is more precise than that of Cellpose, see inset in **(E)**. There, Cellpose detects the cell cytoplasm or a part of it rather than the cell outline, unlike EPySeg, resulting in a significant shift of the mask (magenta) with respect to the membrane maximum intensity. Altogether EPySeg detects more cells than Cellpose. Scale bars represent 25 µm.

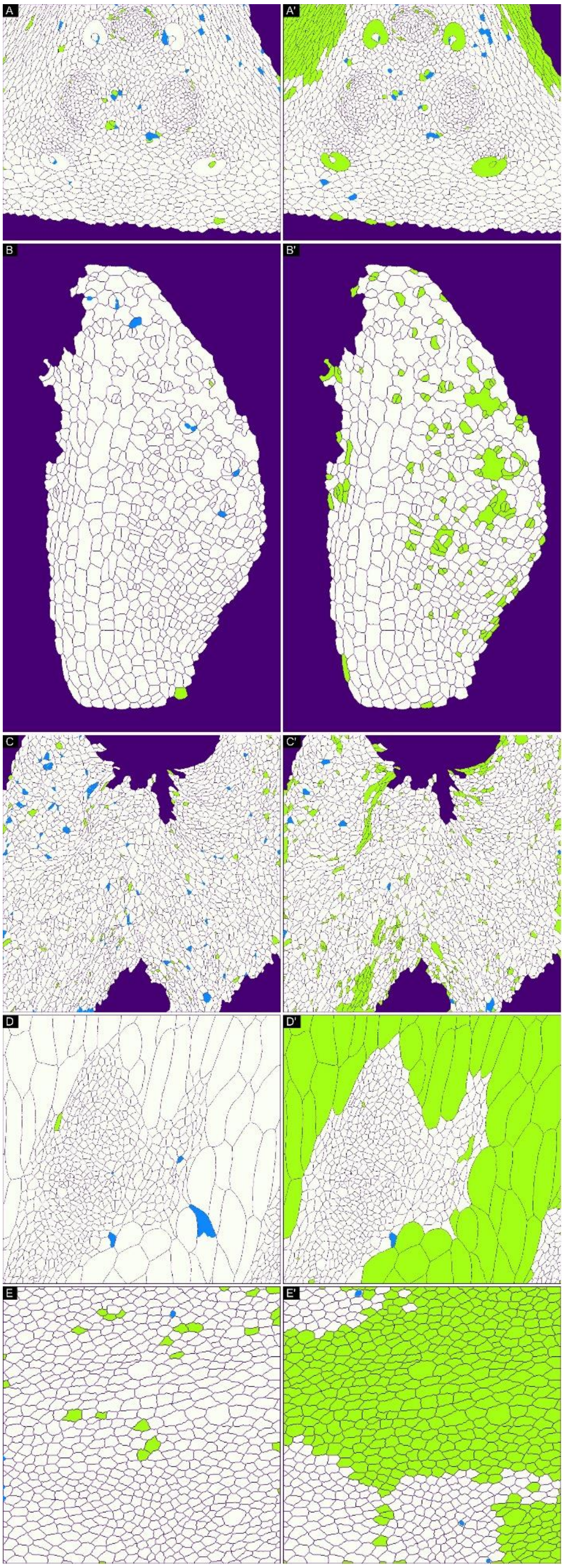

Figure S2: A comparison of EPySeg and Cellpose segmentation

Side by side comparison of EPySeg (left panels) and Cellpose (right panels) over- (blue) and under-segmentation (green) overlaid over the ground truth segmentation (see also **Table S1** for average precision quantification). Images in **(A, A')**, **(B, B')**, **(C, C')**, **(D, D')**, **(E, E')** correspond to images **(A, B, C, D, E)** in **Fig. S1**, respectively. **(A, A')** Cellpose did not segment stretched cells of the fly head. **(B, B')** Cellpose fails to segment non-convex (complex) and small cells in the plant epithelium. **(C, C')** Cellpose does not segment most small cells in the vertebrate epithelium. **(D, D')** Cellpose segmentation eludes all the giant larval cells from the fly abdomen. **(E, E')** Cellpose misses dim cells in the central region of the drosophila wing. Altogether, both tools show little over-segmentation (blue).

Table S1. Quantitative comparison of the segmentation efficiency of EPySeg and Cellpose on cells and tissues. The AP score is a measure of the efficiency of detection of cells by the algorithms. The AP score, that ranges between 0 and 1, is computed with an IoU of 0,7. the SEG score, that also ranges between 0 and 1, is a measure of the quality of the segmentation mask as compared to the ground truth. EPySeg scores were computed using default software settings. Cellpose scores were computed after the optimal cell size was determined by Cellpose. A representative image from the test sets labelled with asterisks is shown in Fig. 2 and Fig. S1.

Click here to Download Table S1