

Supplementary Information

BCM3D 2.0: Accurate segmentation of single bacterial cells in dense biofilms using computationally generated intermediate image representations

Ji Zhang^{1,§}, Yibo Wang^{1,§}, Eric D. Donarski¹, Tanjin T. Toma², Madeline T. Miles¹, Scott T. Acton², Andreas Gahlmann^{1,3,*}

¹Department of Chemistry, University of Virginia, Charlottesville, Virginia, USA

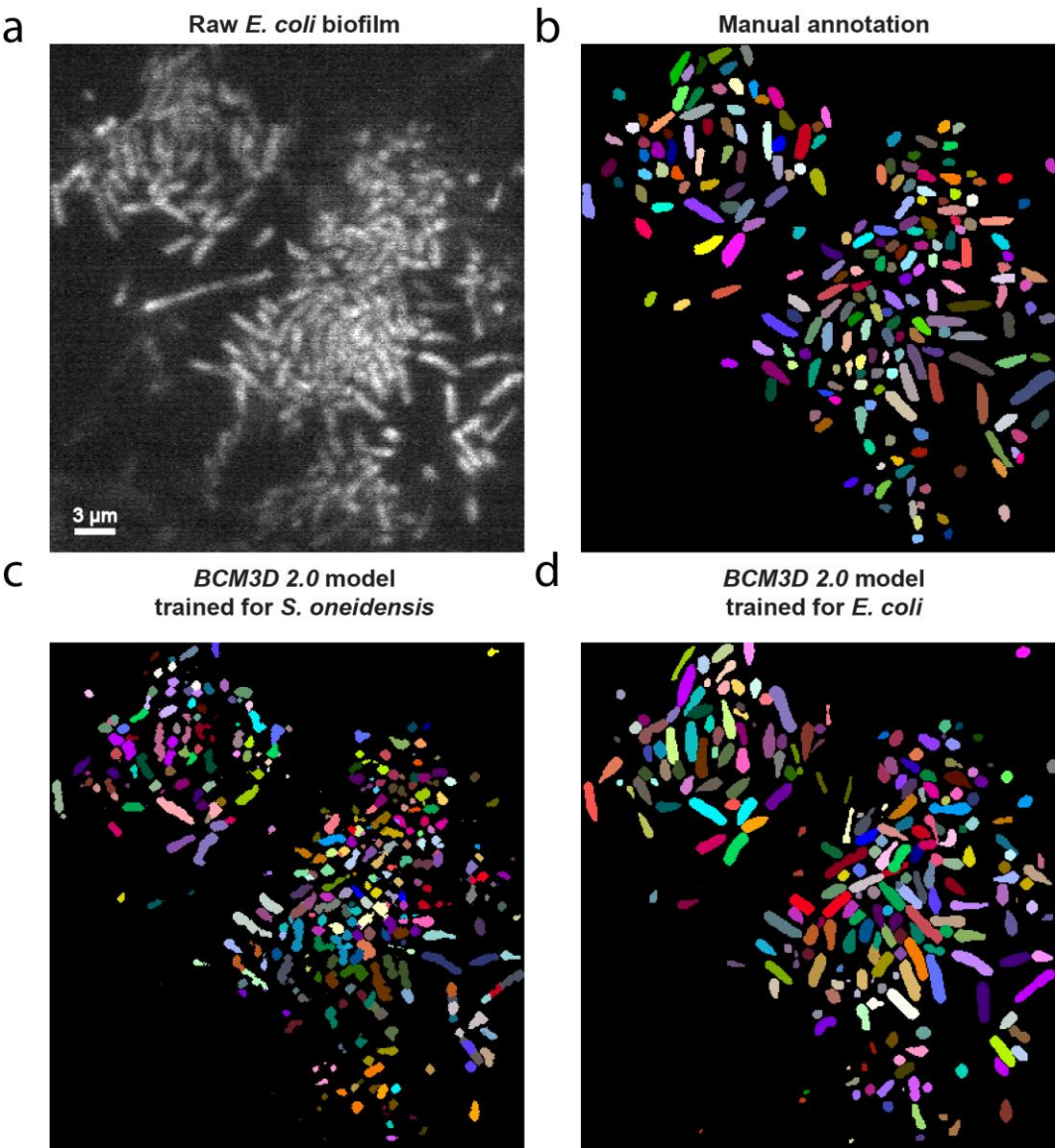
²Department of Electrical & Computer Engineering, University of Virginia, Charlottesville, VA, USA

³Department of Molecular Physiology & Biological Physics, University of Virginia School of Medicine, Charlottesville, Virginia, USA

§These authors contribute equally to this paper

*Corresponding Author: agahlmann@virginia.edu

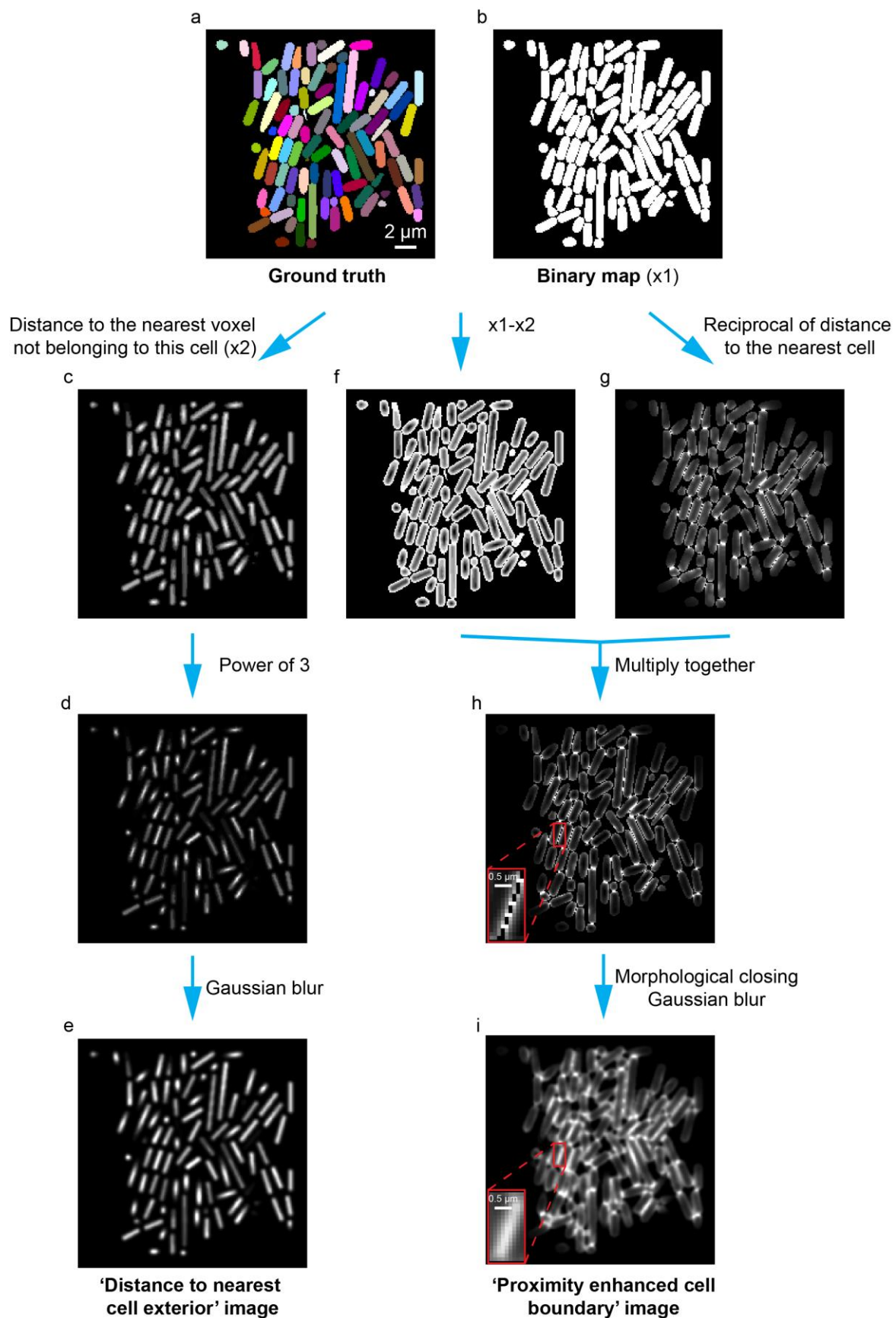
27



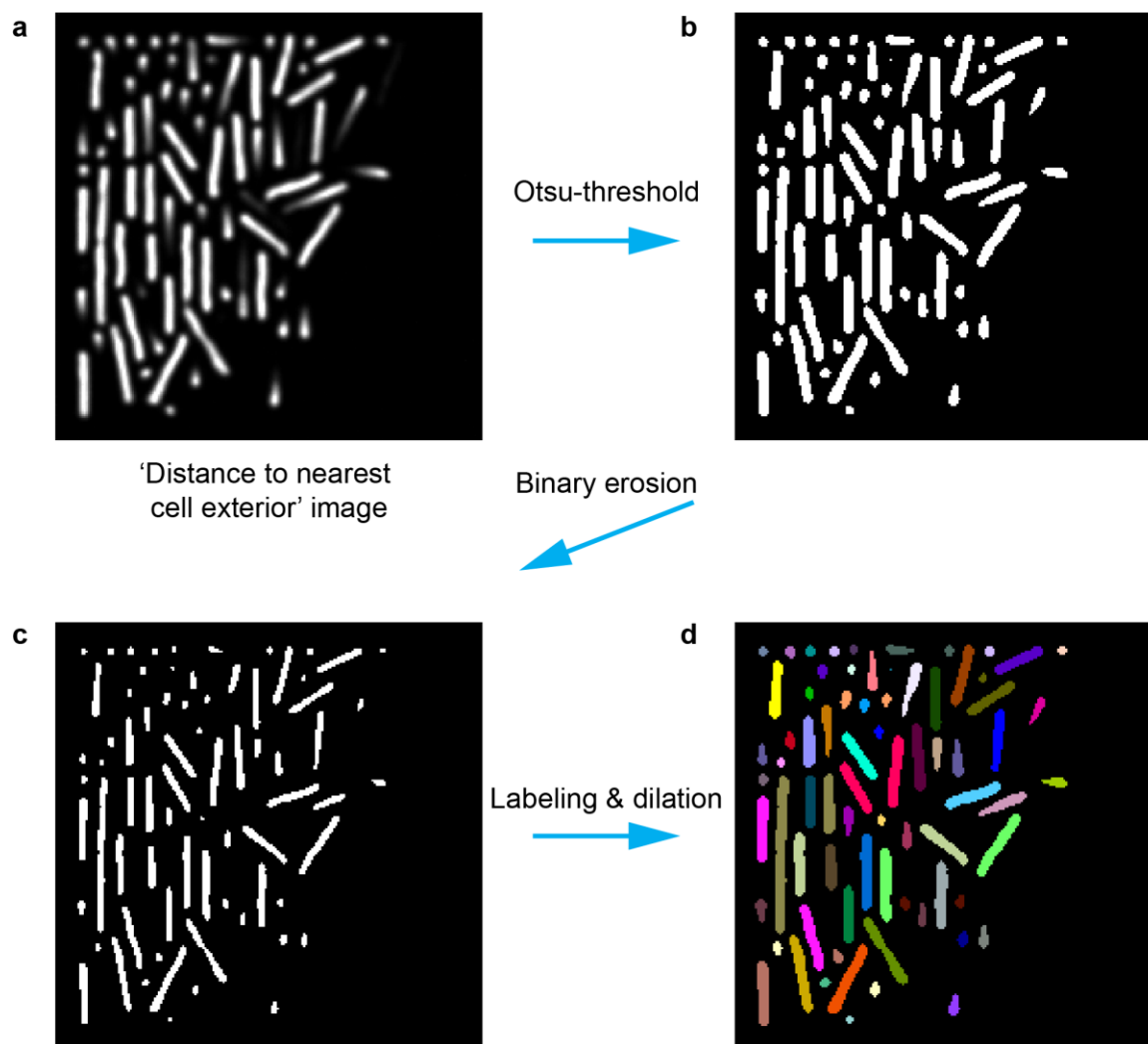
28

29 **Supplementary Figure 1. Segmentation results on Zhang et al. 2020 $t = 600$ min data using**
30 **different training data simulation parameters.**¹ (a) Same 2D cross section shown in Zhang et
31 al. 2020 **Figure 4** $t = 600$ min. Noted the images have been rotated 90 degrees and flipped
32 horizontally in this figure. (b) Manual annotation result. (c) Segmentation result produced by
33 BCM3D 2.0 using cell diameter and cell length (d, l) parameters (0.6 μm , 2 μm) in simulation.
34 (d) Segmentation result produced by BCM3D 2.0 using cell diameter and cell length (d, l)

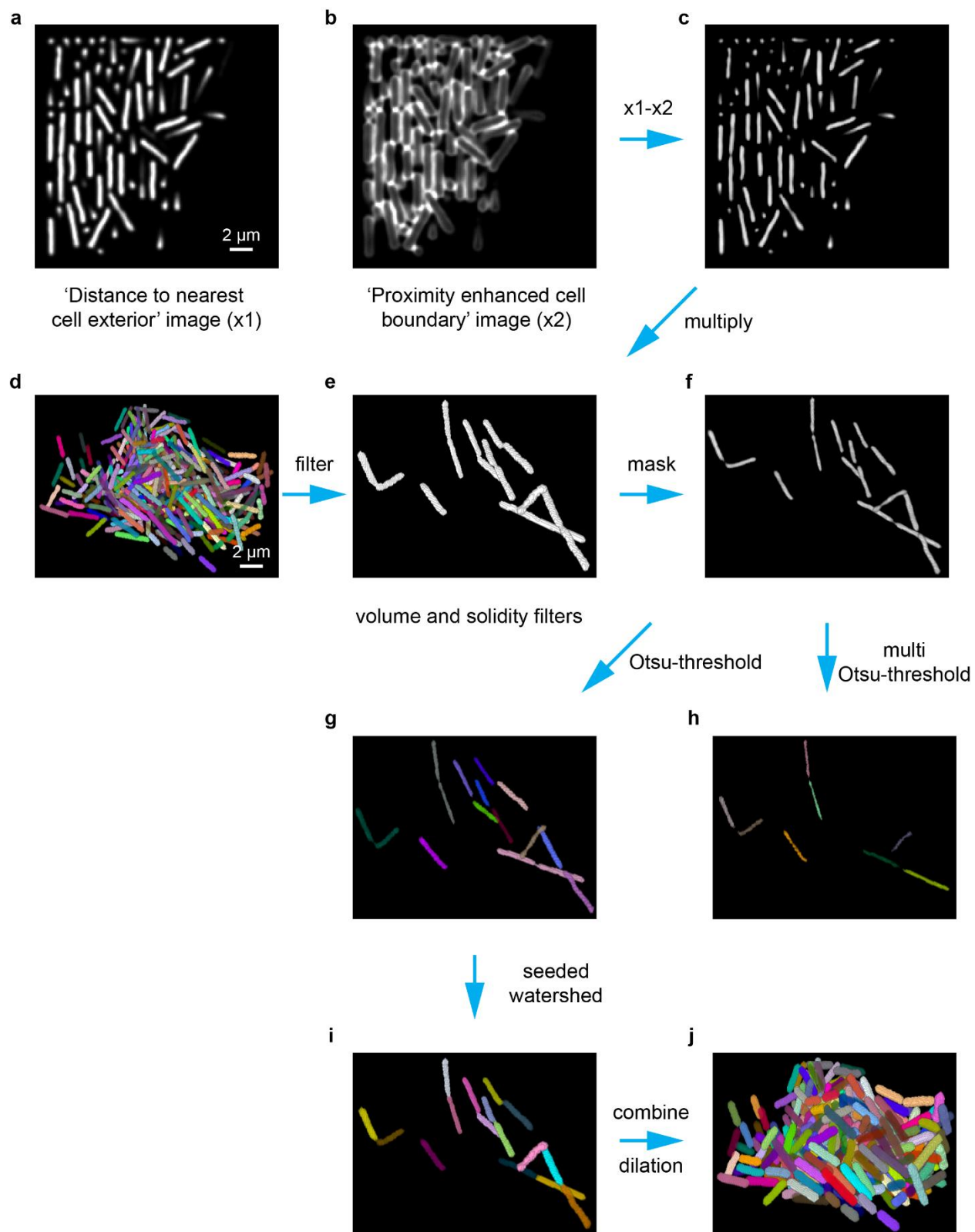
35 parameters ($1\text{ }\mu\text{m}$, $3\mu\text{m}$) in simulation.



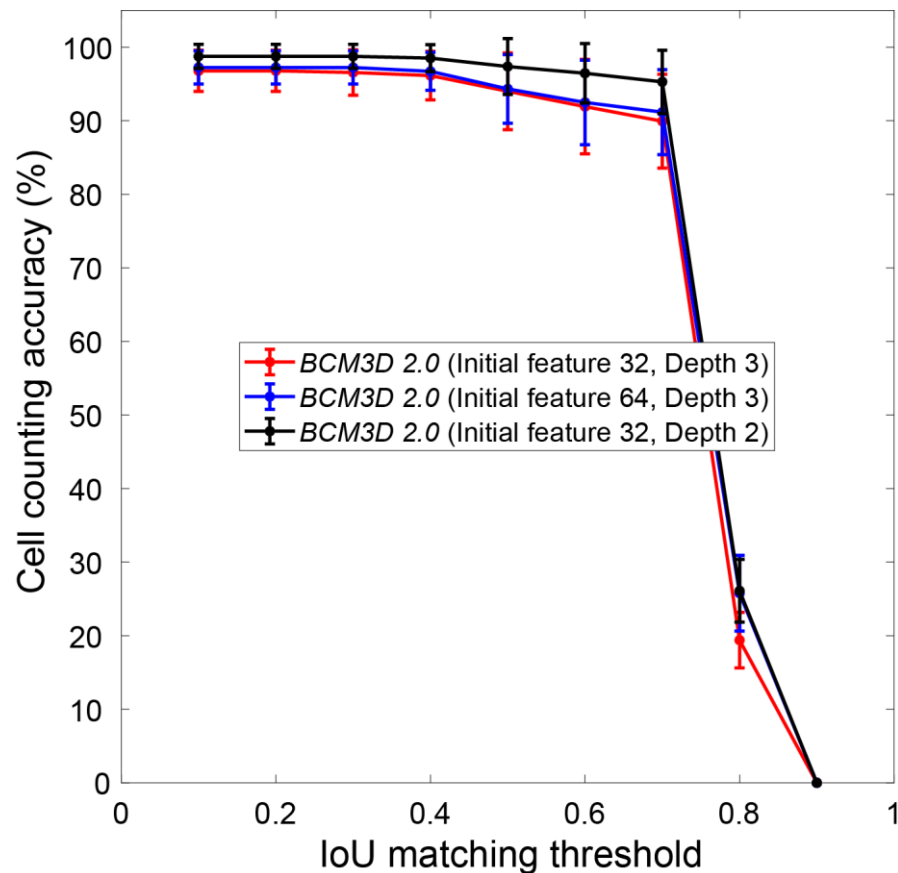
Supplementary Figure 2. Schematic of generating intermediate image representations. (a) Ground truth cell positions. (b) Binary maps based on the ground truth. (c) Images of distance to the nearest voxel not belonging to this cell. (d) A steeper gradient in distance values is obtained by raising each voxel value in panel c to the third power. (e) Smooth images to get 'distance to nearest cell exterior' image representation. (f) Obtain cell boundary by subtracting (c) from (b). (g) Highlight boundary areas that are close to other cells by calculating reciprocal of distance to the nearest cell. (h) Multiply (f) and (g). The inset shows small holes between two cells' boundary. (i) Small holes (inset) are removed in the 'proximity enhanced cell boundary' image by morphological closing and Gaussian blurring.



Supplementary Figure 3. Post-processing step 1: Thresholding of CNN-produced ‘distance to nearest cell exterior’ image. (a) CNN-produced ‘distance to the nearest cell exterior’ image. (b) Apply Otsu-threshold to obtain binary images. (c) To split clusters that are only connected by one or two voxels, the boundary voxels of each object were set to zero (binary erosion). (d) Identify individual cell objects by labeling connected regions and then add back boundary voxels that were erased in panel c.

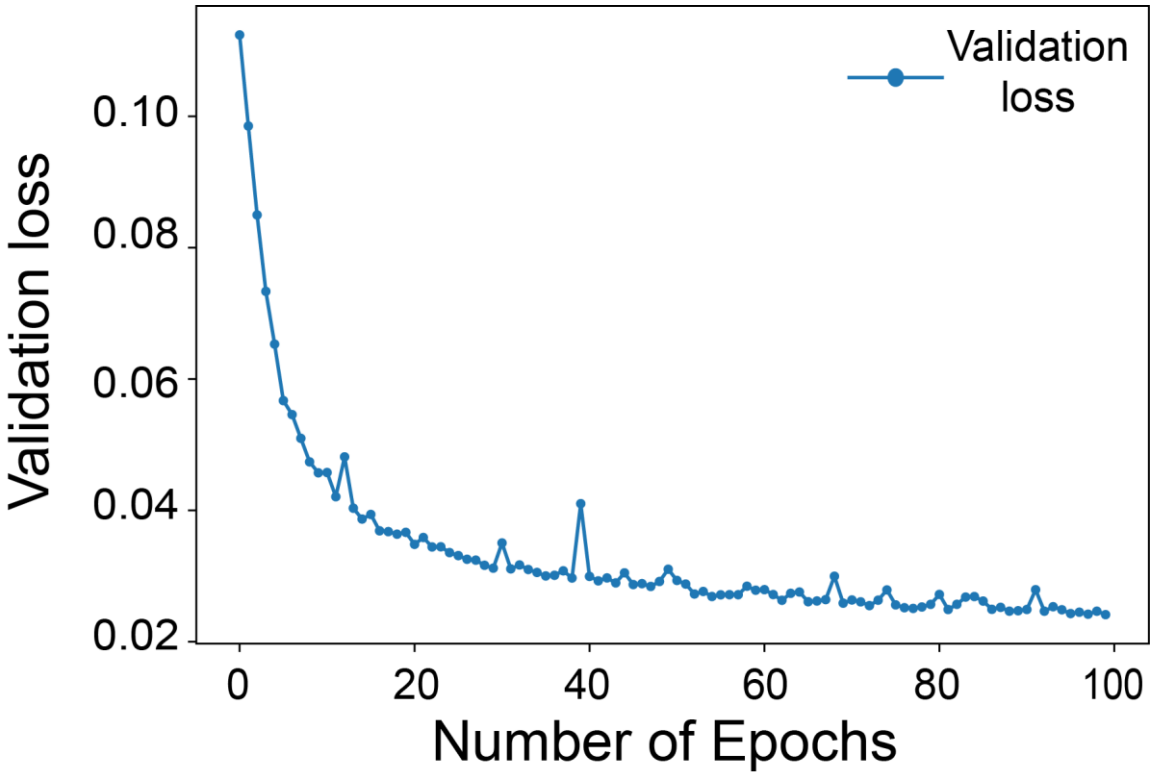


Supplementary Figure 4. Post-processing step 2: Combining of intermediate image representations. (a) CNN-produced ‘distance to nearest cell exterior’ image. (b) CNN-produced ‘proximity enhanced cell boundary’ image. (c) Generate difference map by subtracting (b) from (a) and set negative values to zero. (d) Segmentation results from the process describe in **Figure S3**. (e) Identify objects that need further processing using volume and solidity filters (see Methods). Identified objects are outlined in a binary mask. (f) Mask image in panel c by multiplying it with image in panel e. (g) To split under-segmented clusters, apply seeded-watershed to image in panel f. Seeds are obtained by applying Otsu-thresholding to the image in panel f. (h) If there still are under-segmented clusters in the image in panel g, additional seeds are obtained by applying multi-level Otsu-thresholding the image in panel f. (i) Combine segmented objects from images in panels g and h. (j) Combine (i) and (d) to get final segmentation results.



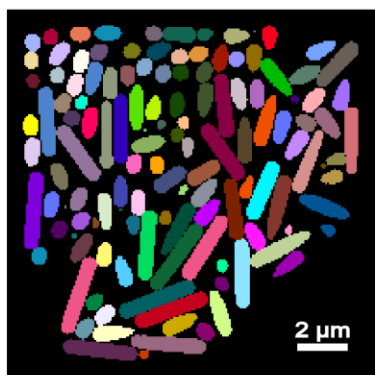
Supplementary Figure 5. Comparison of segmentation accuracies achieved by BCM3D 2.0 using different hyperparameter combinations. The networks were trained with U-Net backbone depth 3, 32 initial features, U-Net depth 3, 64 initial features, and U-Net depth 2, 32

initial features respectively. Segmentation accuracy is parameterized in terms of cell counting accuracy (y axis) and IoU matching threshold (x axis). Each data point is the average of N=10 independent biofilm images. Data are presented as mean values \pm one standard deviation.



Supplementary Figure 6. Validation loss over number of epochs in training. A U-Net network architecture depth of 2, a convolution kernel size of 3, 32 initial feature maps was trained for 100 epochs. Twenty simulated biofilm images with randomly selected cell densities and signal-to-background ratio were used to train the model.

a GT for cytosolic fluorescence



b BCM3D 2.0



c BCM3D 1.0 (CNN + LCuts)



d Omnipose



e Cellpose 2.0



f GT for membrane fluorescence



g BCM3D 2.0



h BCM3D 1.0 (CNN + LCuts)



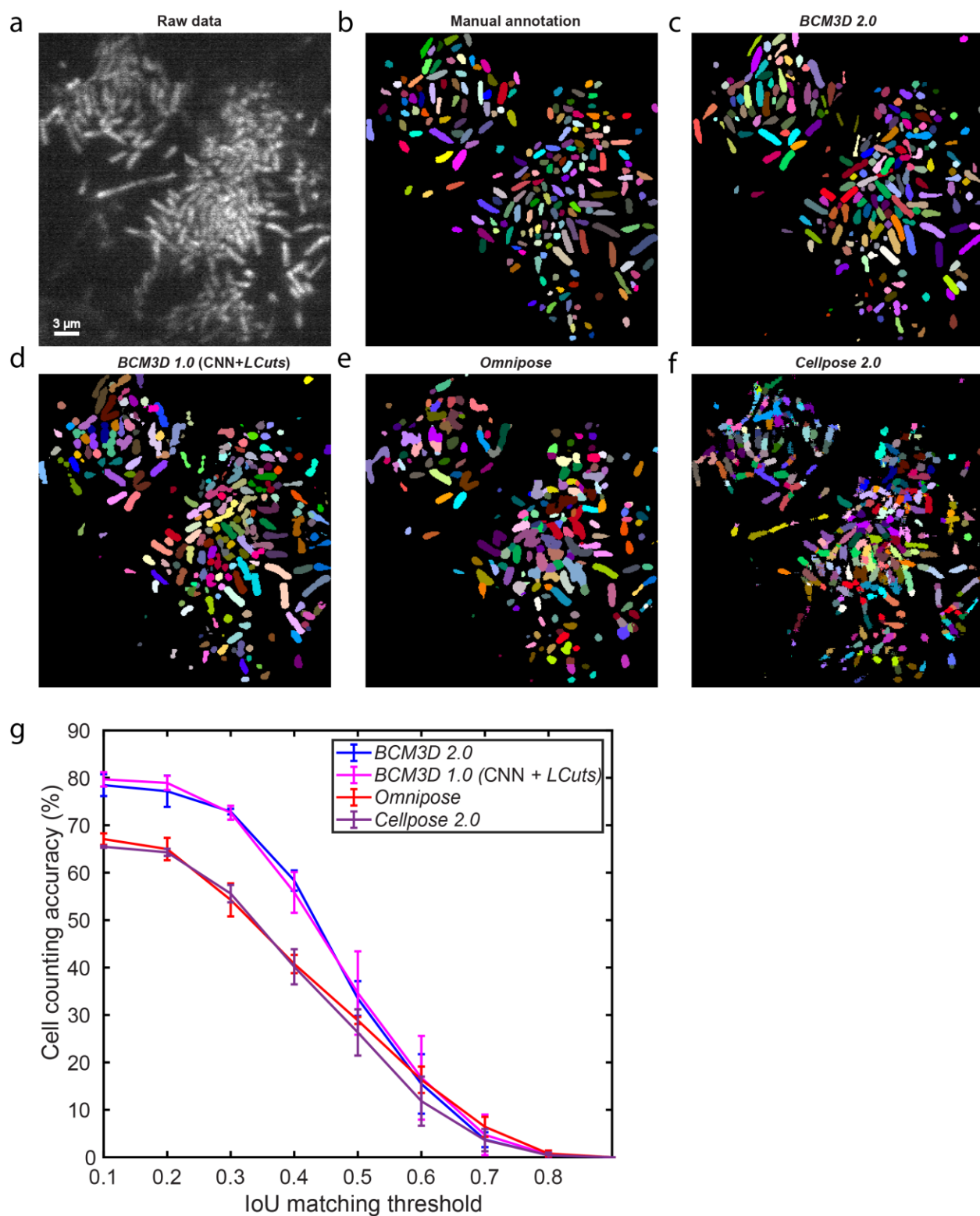
i Omnipose



j Cellpose 2.0

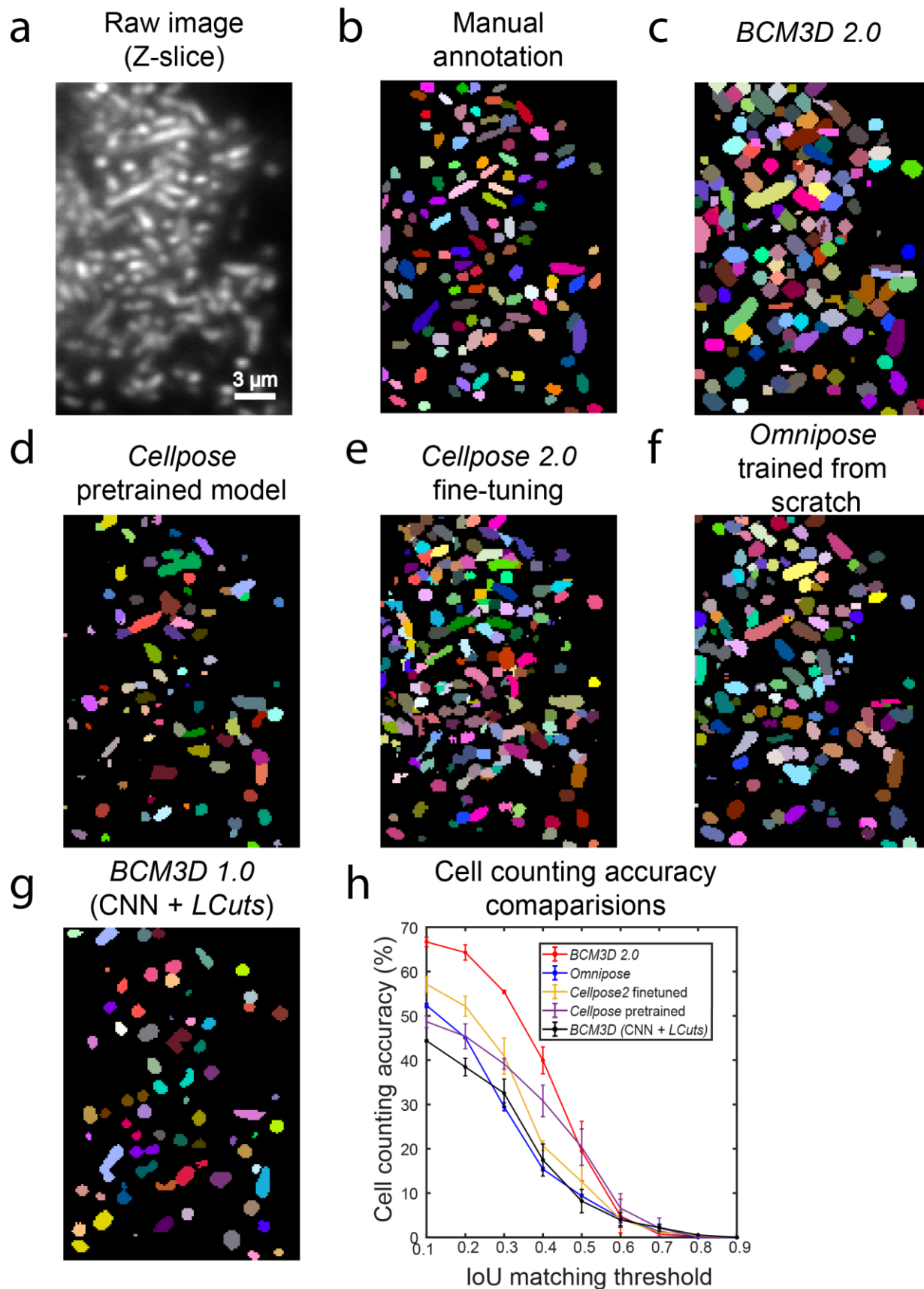


Supplementary Figure 7. Visualization of segmentation results for simulated datasets used for Figure 2e and 2f. (a, f) 2D cross section through the ground truth cell volumes used to generate the simulated images in **Figure 2b** and **2d**. (b-e) Segmentation results produced by *BCM3D 2.0*, *BCM3D 1.0* (CNN+LCuts), *Omnipose*, and *Cellpose 2.0* respectively for cytosol-labeled biofilms. (g-j) Segmentation results produced by *BCM3D 2.0*, *BCM3D 1.0* (CNN+LCuts), *Omnipose*, and *Cellpose 2.0* respectively for membrane-labeled biofilms.

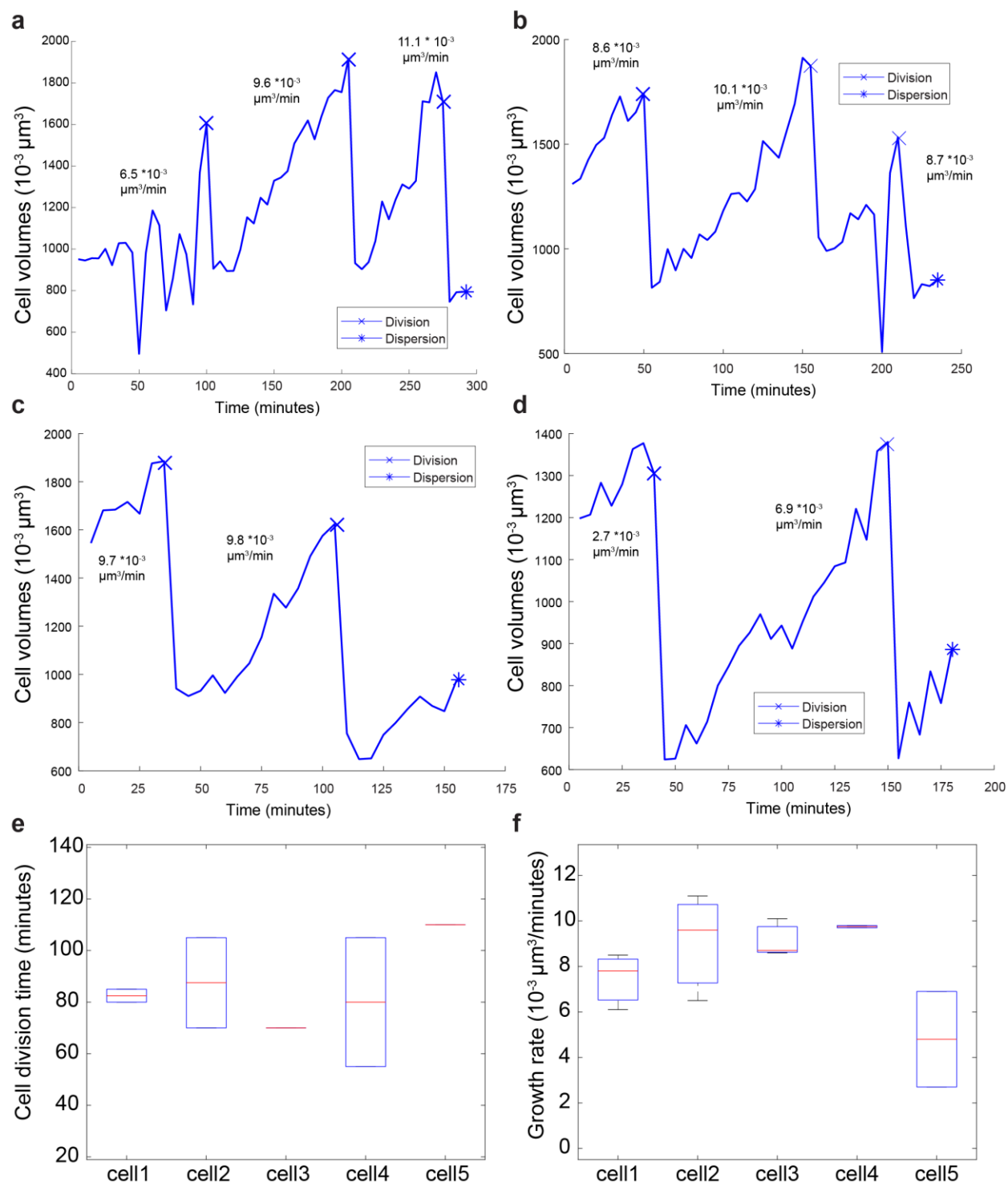


Supplementary Figure 8. Comparison of segmentation results on Zhang et al. 2020 $t = 600$ min data.¹ Noted the images have been rotated 90 degrees and flipped horizontally in this figure. (a) Same 2D cross section shown in Zhang et al. 2020 **Figure 4** $t = 600$ min. (b)

102 Manual annotation result. (c) Segmentation result produced by *BCM3D 2.0*. (d) Segmentation
103 result produced by *BCM3D 1.0 CNN + LCuts*. (e) Segmentation result produced by *Omnipose*
104 trained from scratch for up to 800 epochs using the same simulated training dataset that *BCM3D*
105 *2.0* was trained on. (f) Segmentation result produced by *Cellpose 2.0* fine-tuned for up to 250
106 epochs using the randomly selected 2D slices of the same simulated training dataset that *BCM3D*
107 *2.0* was trained on. (g) Segmentation accuracy is parameterized in terms of cell counting
108 accuracy (y-axis) and IoU matching threshold (x-axis). Each data point is the average of the cell
109 counting accuracies calculated using annotation maps traced by $N = 3$ researchers. Data are
110 presented as mean values \pm one standard deviation indicated by error bars.



Supplementary Figure 9. Performance comparisons on *S. oneidensis* dataset in Figure 3. (a) Small 2D cross section selected from the large 3D biofilm shown in **Figure 3**. (b) Manual annotation result. (c) Segmentation result produced by *BCM3D 2.0*. (d) Segmentation result produced by the pretrained *Cellpose* model. (e) Segmentation result produced by *Cellpose2* fine-tuned using the simulated training data from this work. *Cellpose2* can be used for quickly prototyping new specialist models from a generalist model with minimal new training data. We note that *Cellpose2* only uses 2D data for training and fine-tuning, so thirty different 2D slices in all three dimensions of the 3D training data were selected to fine-tune the network. (f) Segmentation result produced by *Omnipose* trained from scratch for up to 800 epochs using the same simulated training dataset that *BCM3D 2.0* was trained on. Validation loss was observed to stop decreasing after approximately 600 epochs. (g) Segmentation result produced by *BCM3D 1.0 CNN + LCuts* (h) Segmentation accuracy is parameterized in terms of cell counting accuracy (y-axis) and IoU matching threshold (x-axis). Each data point is the average of the cell counting accuracies calculated using annotation maps traced by $N=2$ researchers. Data are presented as mean values \pm one standard deviation indicated by error bars.



Supplementary Figure 10. (a-f) Cell volumes over time for four additional cells. The solid line represents a manually annotated trajectory where cell division and dispersion events are detected manually and indicated on the trajectories. Single-cell growth rate was measured by calculating the slope of a line that connects both the beginning and end points for each trajectory. (e) Box plot

of cell division time for the selected five cells (four in this figure plus one in Figure 5). (f) Box plot of single-cell growth rate for the selected five cells.

Supplementary Methods

Thresholding of CNN-produced ‘distance to nearest cell exterior’ images (Supplementary Figure 3)

Thresholding of CNN-produced ‘distance to nearest cell exterior’ images is the first step of the post-processing pipeline. Predicted ‘distance to nearest cell exterior’ images were first normalized by a simple percentile-based normalization method (Supplementary Figure 3a), which we define for an input u as

$$N(u; p_{low}, p_{high}) = \frac{u - perc(u, p_{low})}{perc(u, p_{high}) - perc(u, p_{low})}$$

Where $perc(u, p)$ is the p -th percentile of all voxel values of u . We typically use $p_{low} = 3$ and $p_{high} \in (99.5, 99.9)$. After applying Otsu-thresholding to the ‘distance to nearest cell exterior’ image to obtain a binary image (Supplementary Figure 3b), connected voxel clusters can be isolated and identified as single cell objects by labeling connected regions. To split clusters that are only connected by one or two voxels, the boundary voxels of each object were set to zero before labeling connected regions (Supplementary Figure 3c). After labeling, the erased boundary voxels were added back to each object (Supplementary Figure 3d). A conservative size-exclusion filter was applied: small objects with volume smaller than the radius cubed of the targeted cells were considered background noise and filtered out.

Post-processing of initial thresholding result (Supplementary Figure 3) by introducing ‘proximity enhanced cell boundary’ images (Supplementary Figure 4)

Thresholding of the ‘distance to nearest cell exterior’ image produces a binary image (background = 0, cell = 1), where groups of connected, non-zero voxels identify individual cells in most cases. However, when cells are touching, they are often not segmented as individuals, but remain part of the same voxel cluster (undersegmentation). On the other hand, a small part of a cell may be erroneously identified as another cell (oversegmentation). To address these errors and further improve the segmentation accuracy, we included ‘proximity enhanced cell boundary’ image, seeded watershed, and multi-level Otsu thresholding to post-process the binary images obtained from the normalized ‘distance to nearest cell exterior’ images (**Supplementary Figure 4**).

Step 1. Objects that need further processing were found by evaluating its volume and solidity, i.e., the volume to convex volume ratio. Here, volume is defined as the number of voxels occupied by an object. Convex volume is defined as the number of voxels of a convex hull, which is the smallest convex polygon that encloses an object. The upper limit was found by using the interquartile rule, i.e. the upper limit is quartile 3 (Q3) plus 1.5 times interquartile range (IQR). If an object's volume or solidity is larger than the upper limit, it will be singled out for further processing. All these objects together generate a new binary image (**Supplementary Figure 4e**).

Step 2. To identify and delineate individual cells in the undersegmented connected voxel clusters, CNN-produced ‘proximity enhanced cell boundary’ images were first normalized by the same percentile-based normalization method. We used seeded watershed after combining ‘distance to nearest cell exterior’ images and ‘proximity enhanced cell boundary’ images. Specifically, we generated a difference map by subtracting the ‘proximity enhanced cell boundary’ image from the ‘distance to nearest cell exterior’ image and then set all negative valued voxels to zero (**Supplementary Figure 4a, b, c**). This difference map was then multiplied by the binary image

generated in Step 1 (**Supplementary Figure 4f**). Then, the resulting image was segmented by seeded watershed. Seeds were obtained by Otsu thresholding of the difference map and seeds with a volume smaller than 30 voxels were removed (**Supplementary Figure 4g**). These new objects were again evaluated by their volume and solidity; if there still exist unmatched objects, a multi-level Otsu thresholding will be applied to further generate seeds. Unlike simple Otsu thresholding, which only separates voxels into two classes, foreground and background, multi-level Otsu calculates several thresholds, determined by the number of desired classes. Here, we used the following five classes: background, transition area between background and cell border, cell border, transition area between cell border and cell interior, cell interior. Seeds were extracted by using the third and the fourth threshold successively, i.e. the threshold between the third and the fourth classes (cell border, transition area between cell border and cell interior), and the threshold between the fourth and the fifth classes (transition area between cell border and cell interior, cell interior). The same size filter was used to remove unreasonable small seeds (**Supplementary Figure 4h**).

Step 3: All post-process objects were added together (**Supplementary Figure 4i**) and combined with the initially reasonable objects, i.e., the objects haven't been singled out in Step 1. A conservative size-exclusion filter was applied: small objects with volume 10 times smaller than the upper limit volume were considered unreasonable small parts and filtered out. Since the 'distance to nearest cell exterior' images were confined to the cell interior, we dilated each object by 1-2 voxels to increase the cell volumes using standard morphological dilation (**Supplementary Figure 4j**).

Supplementary References

205 1 Zhang, M. *et al.* Non-invasive single-cell morphometry in living bacterial biofilms. *Nature*
206 *communications* **11**, 6151, doi:10.1038/s41467-020-19866-8 (2020).
207