

Supplementary Data

Supplementary Methods

Level Set Segmentation

We adapt the region based energy model, known as the Chan-Vese Level Set Model, for segmentation, because this method can detect objects without sharp edges, and it allows for automatic change of topology. Some segmentation results are shown in Supplementary Figure 1. If the signal to noise ratio is low, the cell boundaries are not easy to segment, and the cell detection may fail. An example is the central part denoted by yellow circles in Supplementary Figure 1 D E F, where some cells can not be properly segmented and detected. Our proposed method tries to exploit the spatio-temporal redundancy in the data to overcome the errors in cell detection and segmentation.

Finding the Seed (First) Cell Pair

Let us assume that the first image frame has N cells, and the second image frame has M cells. According to the distance function in Equation (1), we can get the distance value $D_L(c_i, c_j)$ of every cell pair (c_i, c_j) , and pick the most similar cell pair (c_a, c_b) that satisfies

$$D_L(c_a, c_b) = \min_{c_i, c_j} \{D_L(c_i, c_j), i = 1, \dots, n, j = 1, \dots, m\} \quad (2)$$

Alternatively, we could find the top few most similar cell pairs instead.

Growing Cell Correspondences from a Seed Pair

Given a matching cell pair, we now describe the process to compute matches for the neighboring cells. This distance function is composed of two parts: the normalized difference of the edge lengths and the difference of the orientation angles between the edges. Let c_i and c_j be two correctly matched cells respectively at time instants t and $t+1$, and c_m and c_n be two neighboring cells around them respectively. Then the distance measure function is

$$D_N(c_i, c_j; c_m, c_n) = \lambda_1 \cdot \frac{|l_{c_m, c_i}(t) - l_{c_n, c_j}(t+1)|}{l_{c_m, c_i}(t)} + \lambda_2 \cdot \frac{|A_{c_i}(t) - A_{c_j}(t)|}{A_{c_i}(t)} + \lambda_3 \cdot \frac{|\theta_{c_m, c_i}(t) - \theta_{c_n, c_j}(t+1)|}{\theta_{c_m, c_i}(t)} \quad (3)$$

If the distance is small, c_m and c_n are identified as matching cells, and the process continues. This leads to a recursive procedure for matching cells from a seed to the neighbors. Note that this distance does not include summing over all the cells in the neighborhood unlike Equation (1) in the main paper. This allows us to find correspondences even when the images may be significantly corrupted.

The process is illustrated in Figure 3, where the seed pair (1,1) identified in C and D is used to grow the tracking process spatially using the local graph matching method. By distance function Equation (3), we can get the correspondence between the neighboring cells (numbers 2-8) of the seed pair (1, 1), based on which we can find the correspondences among the next layer of neighboring cells (numbers 9-21).

Cell Division Detection

Here we provide a summary of how cell divisions are detected in the local graph matching approach. As an example, consider the local graphs A and B illustrated in Figure 4. Formally, the procedure for detecting the cell division can be expressed as follows:

1. Find the corresponding cell pair (M, A) using the local graph matching technique.
2. Check whether the area difference of (M, A) exceeds half the area size of a parent cell; if true, then (M, A) is a candidate pair of a parent cell and a sibling cell.
3. Search in the neighborhood of the daughter candidate A to find the other daughter candidate (sibling) cell B ;
4. Compare the sibling cell candidates A and B ; if the sum of their areas is about the same as the parent cell M , then we can say that M divides into two daughter cells A and B . We can also check the distance between A and B , which should be within about half the average distance between two neighboring adult cells.

Correspondence across Image Slices

Using the local graph matching methods described above, we can compute correspondences across slices S_k and S_{k+1} at the same time instants, i.e., correspondences within the same stack. The distance measure function is

$$D_S(c_i, c_j) = \frac{\lambda_1}{M} \cdot \sum_{c_m \in N(c_i), c_n \in N(c_j)} \frac{|l_{c_m, c_i}(S_k) - l_{c_n, c_j}(S_{k+1})|}{l_{c_m, c_i}(S_k)} + \lambda_2 \cdot \frac{|A_{c_i}(S_k) - A_{c_j}(S_{k+1})|}{A_{c_i}(S_k)} \\ + \frac{\lambda_3}{M} \cdot \sum_{c_m \in N(c_i), c_n \in N(c_j)} \frac{|\theta_{c_m, c_i}(S_k) - \theta_{c_n, c_j}(S_{k+1})|}{\theta_{c_m, c_i}(S_k)} + \lambda_4 \cdot \frac{\|\underline{P}_{c_i}(S_k) - \underline{P}_{c_j}(S_{k+1})\|}{\Delta} \quad (m = n = 1, \dots, M)$$
(4)

Fusion of Spatial and Temporal Correspondences

In the tracking process, we build a correspondence matrix C to denote whether two cells c_i, c_j in two images (may be at different time points or different slices at the same time point) are the same cell or not. The number of rows in C is equal to the number of detected cells in one image, while the number of columns is the number of detected cells in the other image. If the two cells are the same, we set the correspondence value $C_{i,j}$ to 1; otherwise we set it to 0. If cell c_i splits into cells c_{j_1} and c_{j_2} , we set C_{i,j_1} and C_{i,j_2} to 1 at the same time.

Without loss of generality, let us consider the k -th slice and the immediate top and bottom slices for describing the fusion process. This can be repeated for all k , except the top- and bottom-most slices in each stack. For the top-most slice, we will use fuse with the immediate lower slice, while for the bottom-most slice we will fuse with the immediate upper slice. Let us use $\mathbf{C}(t, t+1; k)$ denotes the cells' correspondence across time instants t and $t+1$ at slice S_k , and the correspondence across slices S_k and S_{k+1} at time t as $\mathbf{C}(k, k+1; t)$. The fusion process is composed of the upward fusion matrix $\mathbf{C}^u(t, t+1; k)$ and downward fusion matrix $\mathbf{C}^d(t, t+1; k)$, which can be proved to be products of the individual correspondence matrices as follows (see Figure 6):

$$\begin{cases} \mathbf{C}^u(t, t+1; k) = \mathbf{C}(k, k-1; t) \times \mathbf{C}(t, t+1; k-1) \times \mathbf{C}(k-1, k; t+1) \\ \mathbf{C}^d(t, t+1; k) = \mathbf{C}(k, k+1; t) \times \mathbf{C}(t, t+1; k+1) \times \mathbf{C}(k+1, k; t+1) \end{cases} \quad (5)$$

The final set of the correspondences at slice S_k , denoted by $\mathbf{C}^*(t, t+1; k)$, is obtained by the ‘union operator 1’, i.e.,

$$\mathbf{C}^*(t, t+1; k) = \mathbf{C}(t, t+1; k) \bar{\cup} \mathbf{C}^u(t, t+1; k) \bar{\cup} \mathbf{C}^d(t, t+1; k),$$

which can be expressed algorithmically as follows:

- $\mathbf{C}^*(t, t+1; k) = \mathbf{C}(t, t+1; k);$
- For any new cell pair identified by either $\mathbf{C}^u(t, t+1; k)$ or $\mathbf{C}^d(t, t+1; k)$ or both, add this pair to $\mathbf{C}^*(t, t+1; k)$

With this fusion process, we can find new cell pairs, which are not identified by tracking in a single slice tracking, but can be found through the adjacent slices. With this process, we can track most of the properly segmented cells. The effectiveness of the fusion process in identifying additional cells is shown in Table 2.

Final Cell Lineage Computation

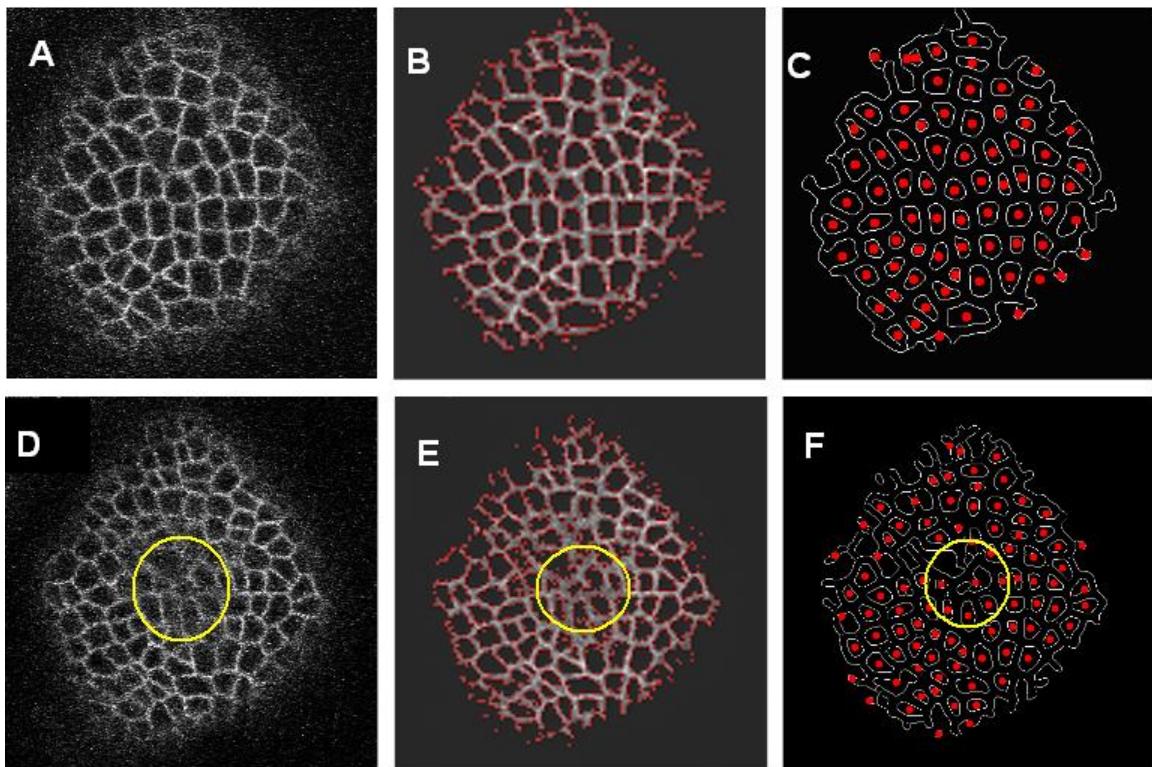
Let us assume that t_1 denotes the first time instant. Then the lineage from the first time instant t_1 to any time instant t ($t > t_1$), i.e. $\mathbf{L}(t_1, t)$, can be mathematically represented as follows

$$\mathbf{L}(t_1, t) = \bigcup_{i=1,2,3, t-i \geq t_1} \mathbf{L}(t_1, t-i) \mathbf{C}^*(t-i, t) \quad (6)$$

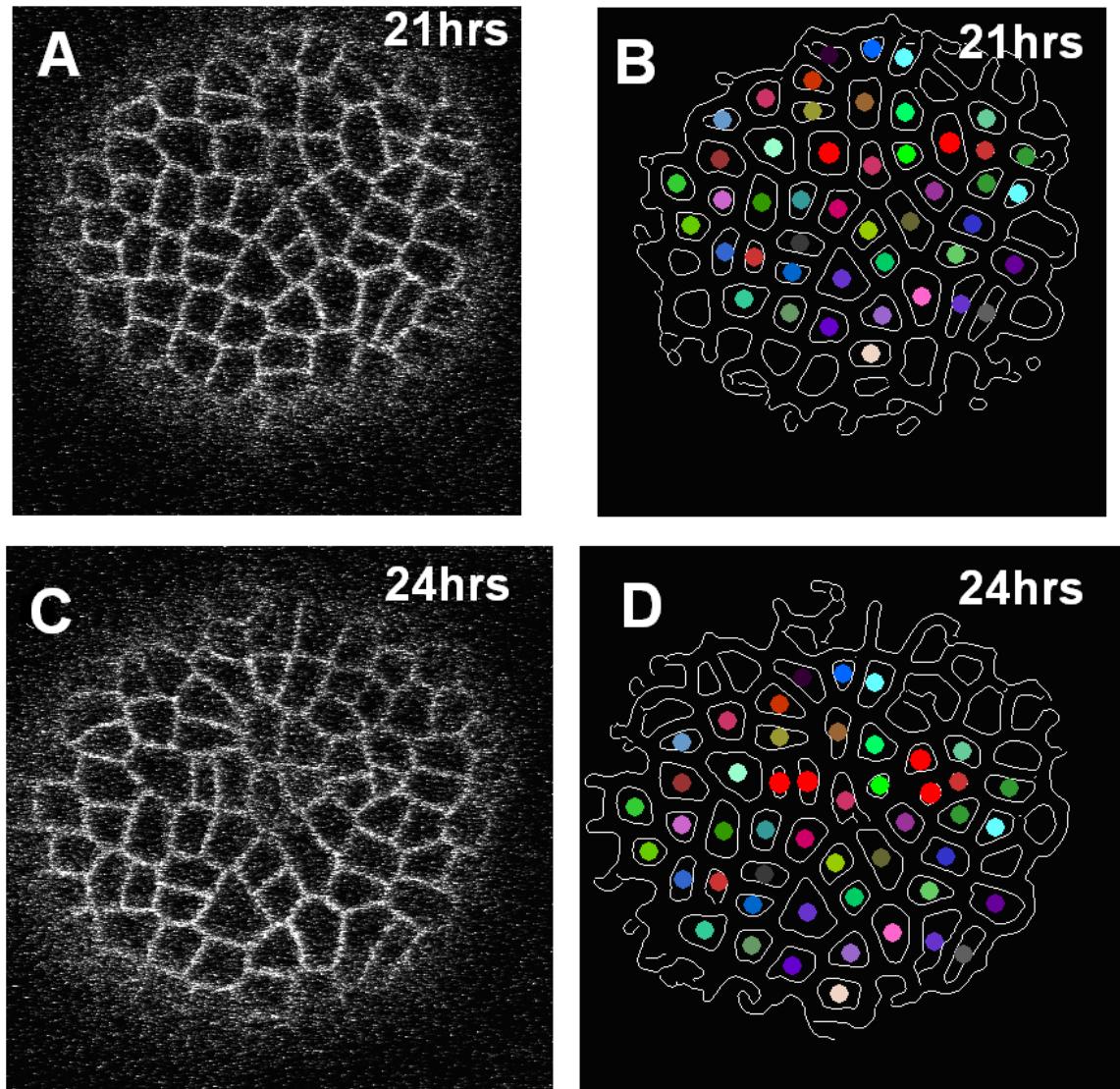
where \mathbf{C}^* denotes the output of the fusion process and this ‘union operator’ $\bar{\cup}$ will combine the lineage results from those 3 paths (see Figure 8) based on a majority voting process. (Since the lineages are computed for the cells after the fusion process of the previous section, we drop the index k in the correspondence matrix.) In this process, a matched cell pair is accepted if the matching result agrees in a majority of the paths. To be specific, a pair that is matched in all 3 paths should agree in at least two, a pair matched in two paths should agree, while a pair matched in only one path will be accepted. The reason why a cell pair may not be matched along all paths is because not all cells may be imaged or detected at each time instant. This process is explained

diagrammatically in Figure 8. An example of the computed cell lineages along 24 time instants (72hours) is presented in Figure 5 and Supplementary Figure 4.

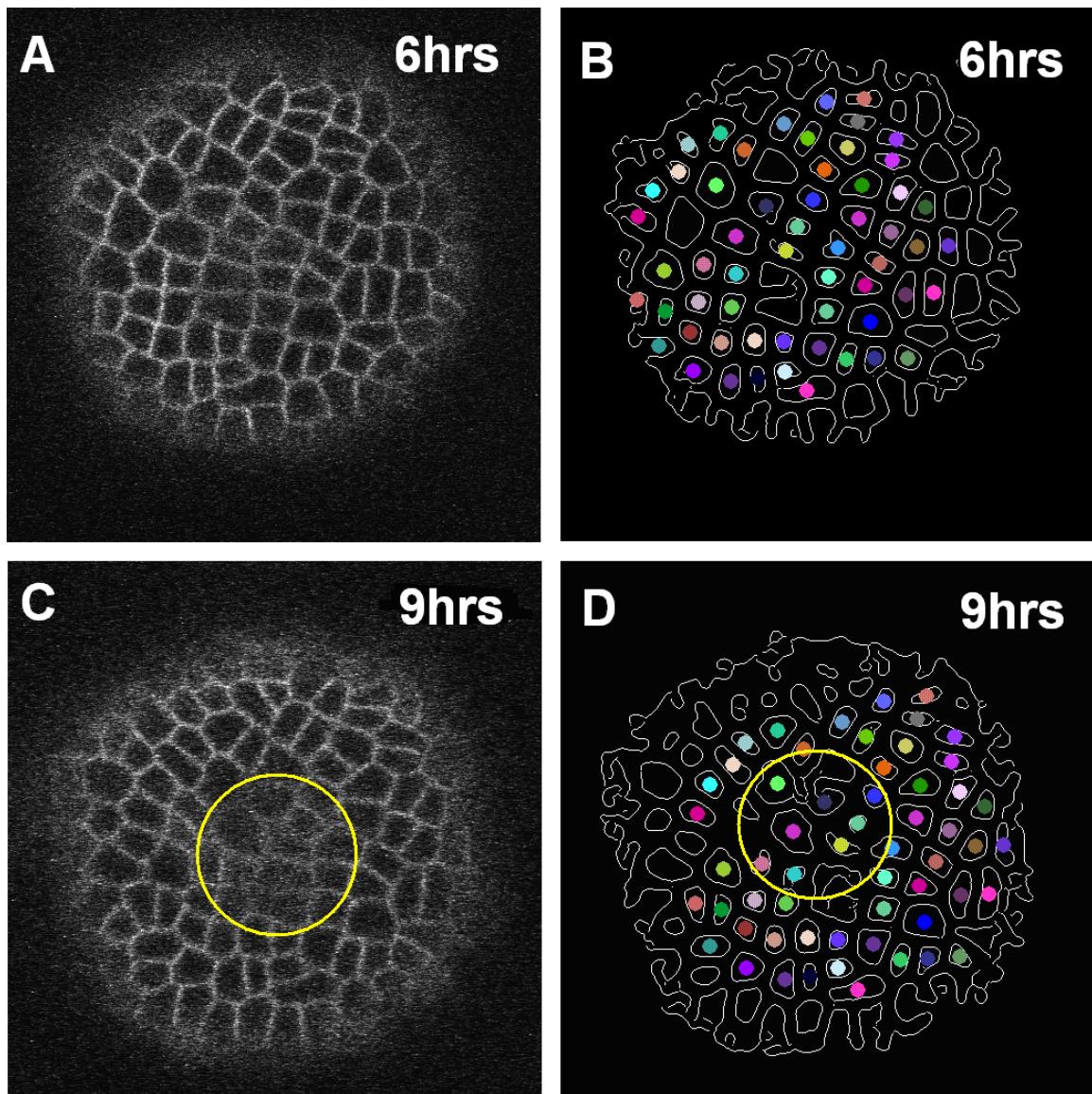
Supplementary Figures



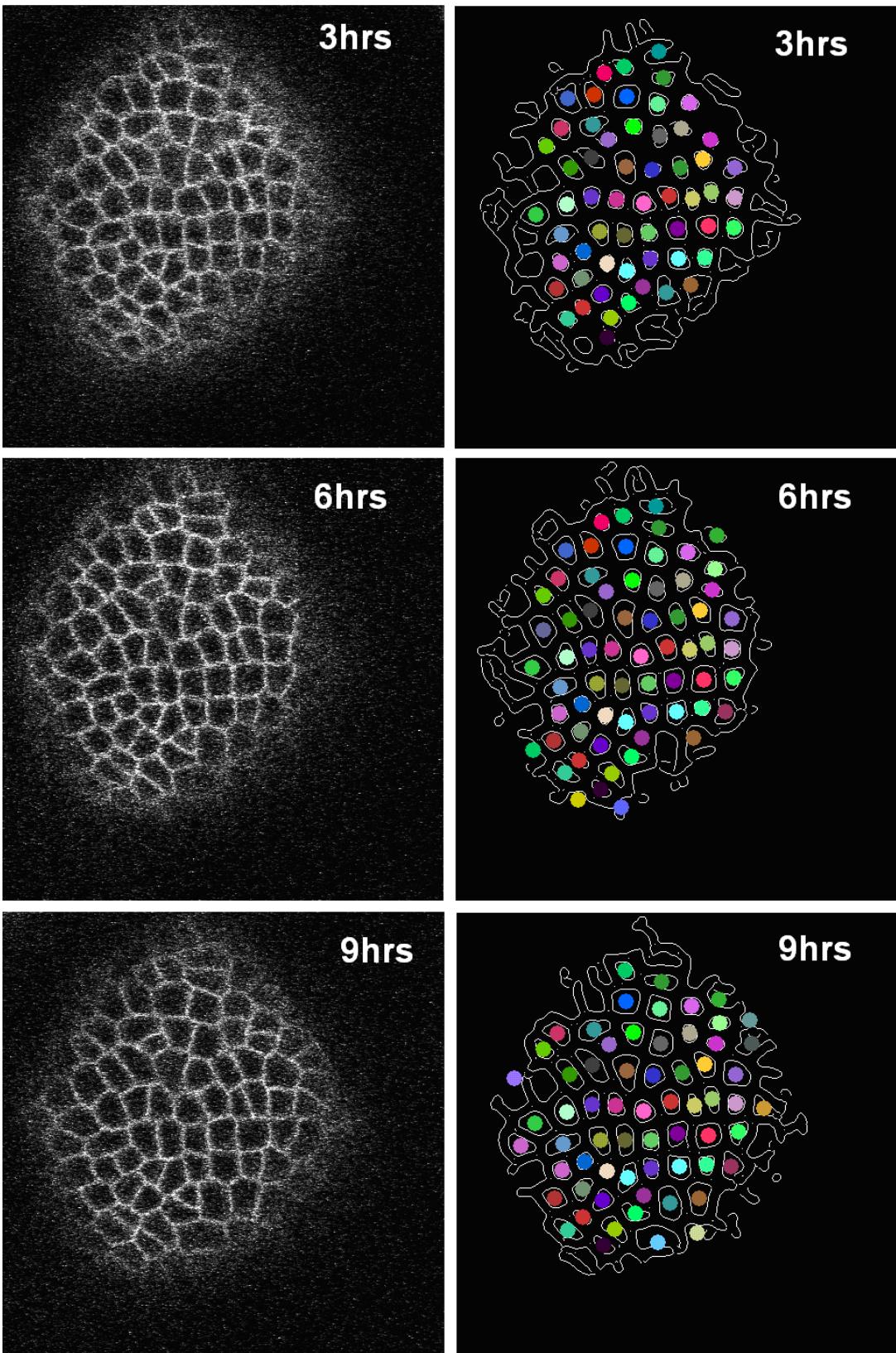
Supplementary Figure 1: Segmentation of SAM cells. Segmentation results (B and E) of the original confocal cross sections acquired at different depths (A and D), and cell centroid detection (C and F) from the segmentation. Note that the cells in the central part of the image (yellow circle) are not segmented properly because of noise.



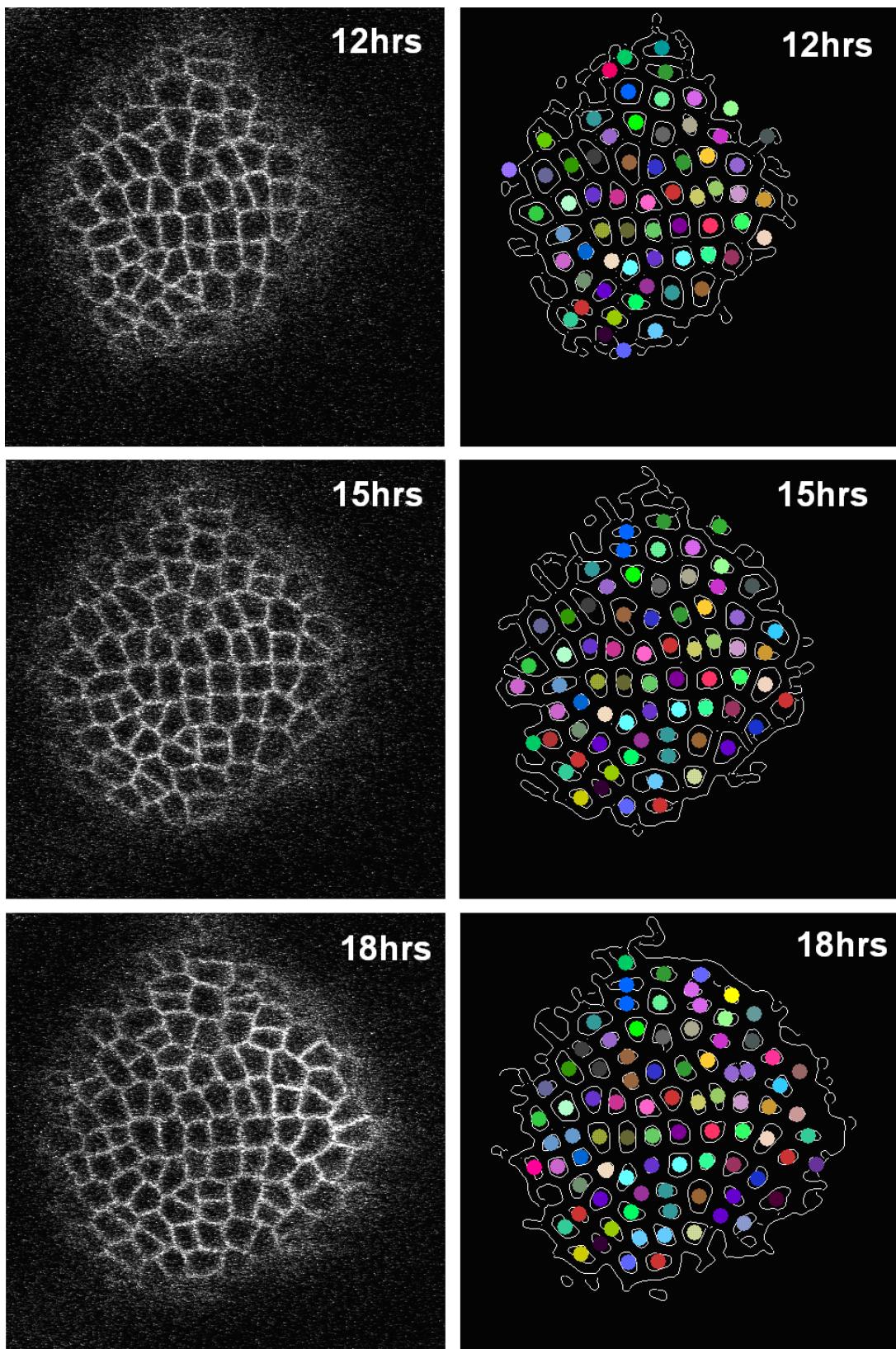
Supplementary Figure 2: Tracking output (right panel) showing 2 cell division events (shown in red color) on a data set from second SAM.



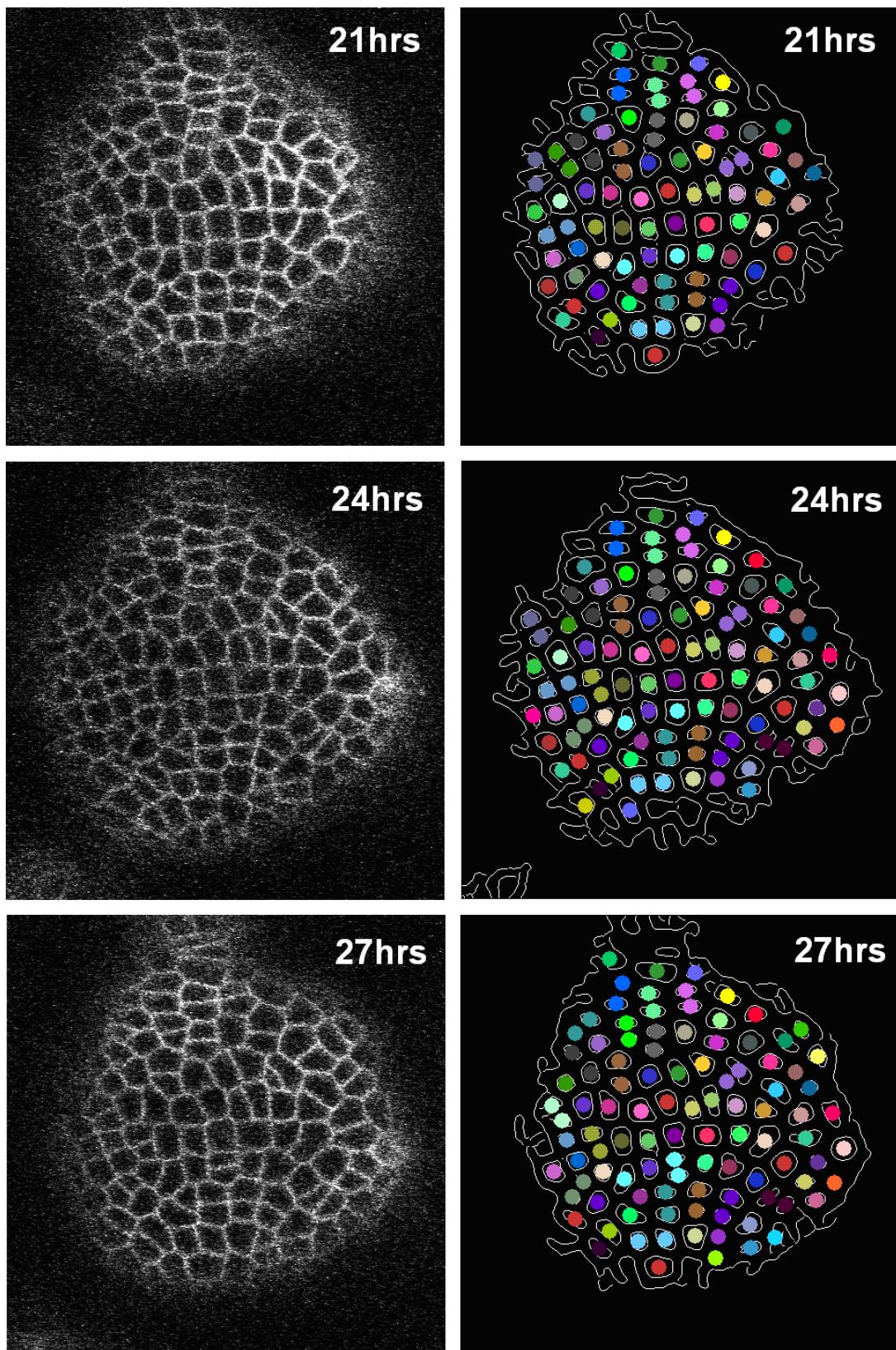
Supplementary Figure 3: Tracking output (B and D) on a noisy confocal cross sections of SAMs (A and C) (some cells within the red circle are not imaged properly).



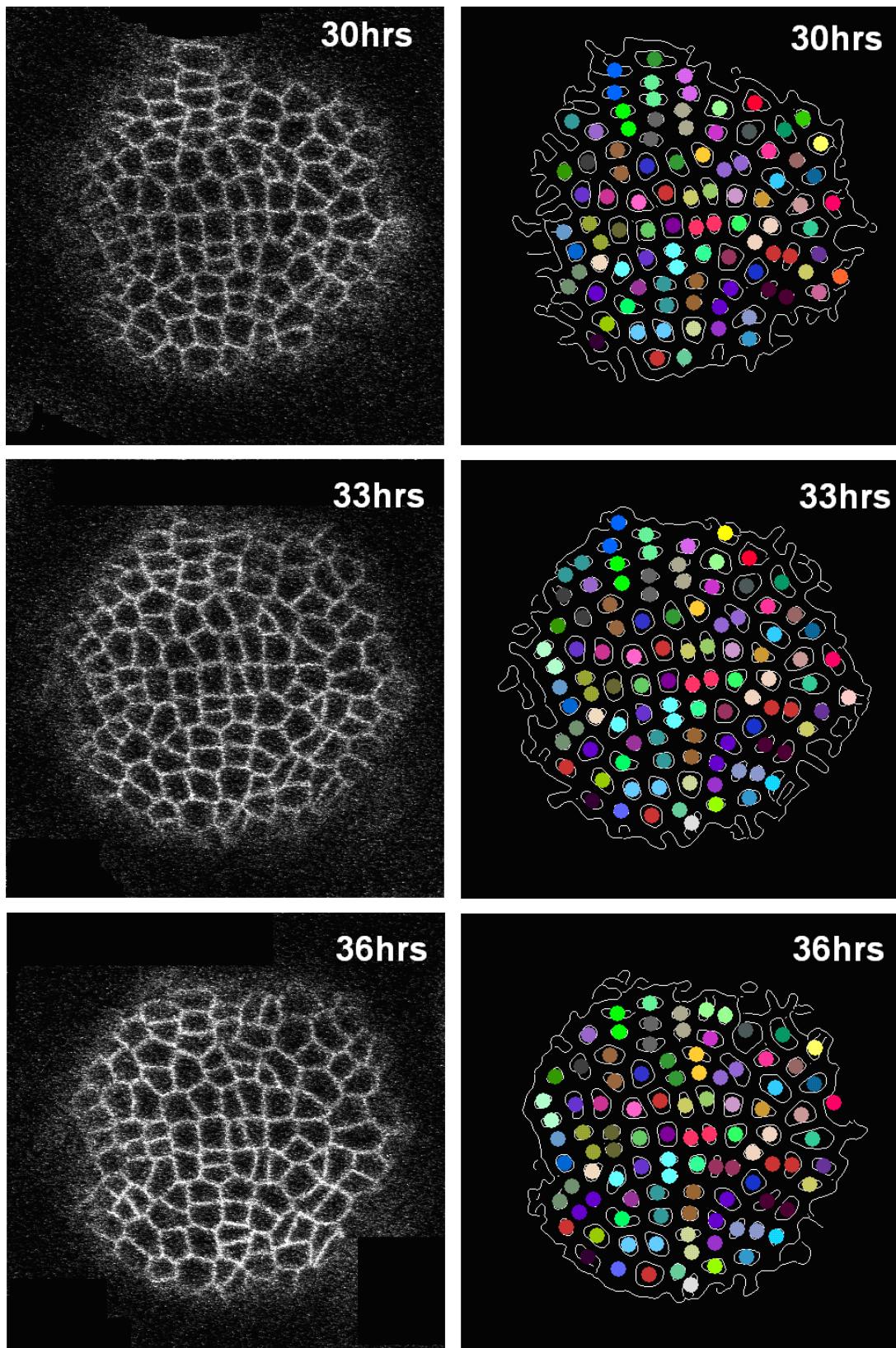
Supplementary Figure 4 (to be continued)



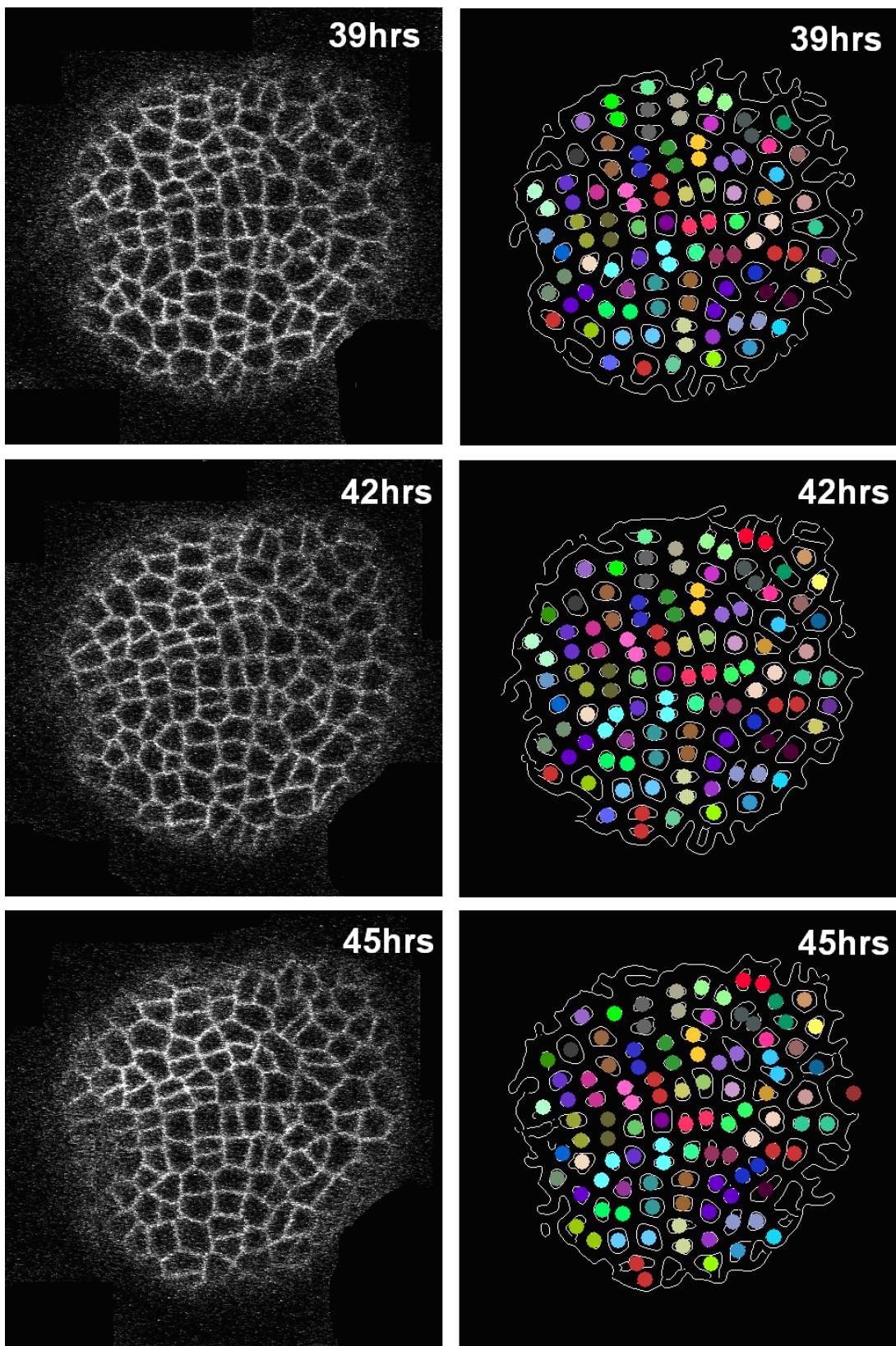
Supplementary Figure 4 (to be continued)



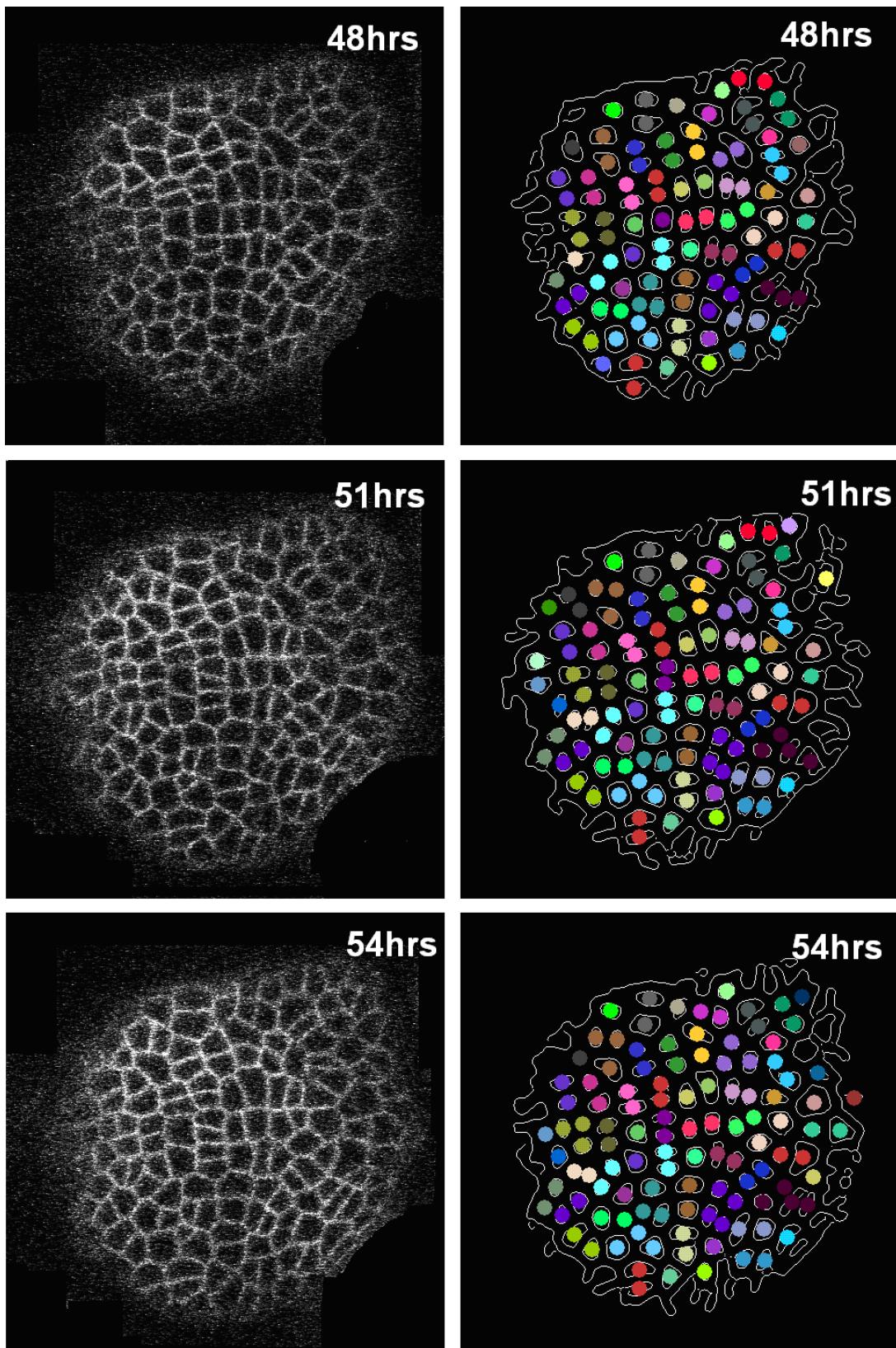
Supplementary Figure 4 (to be continued)



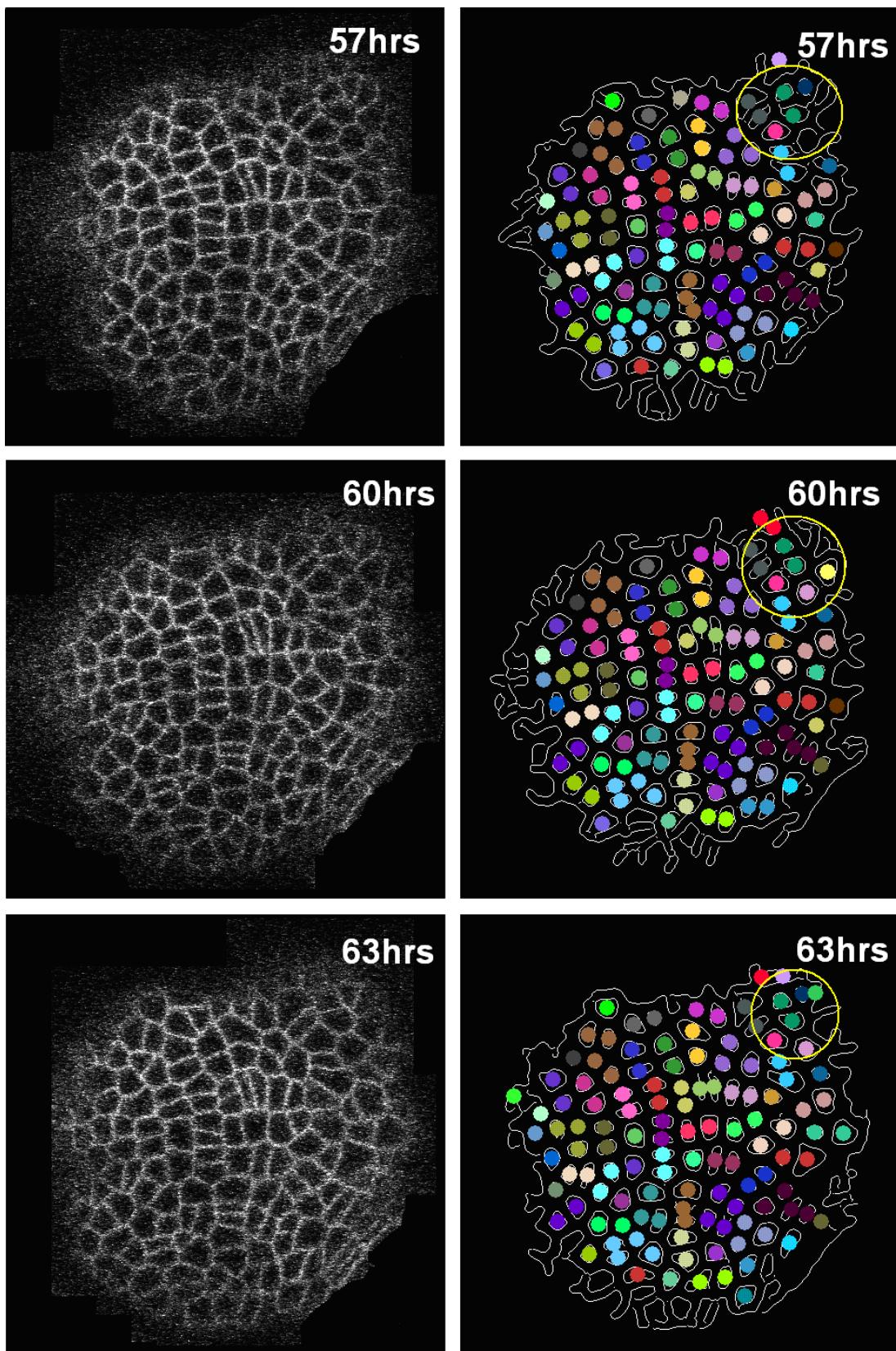
Supplementary Figure 4 (to be continued)



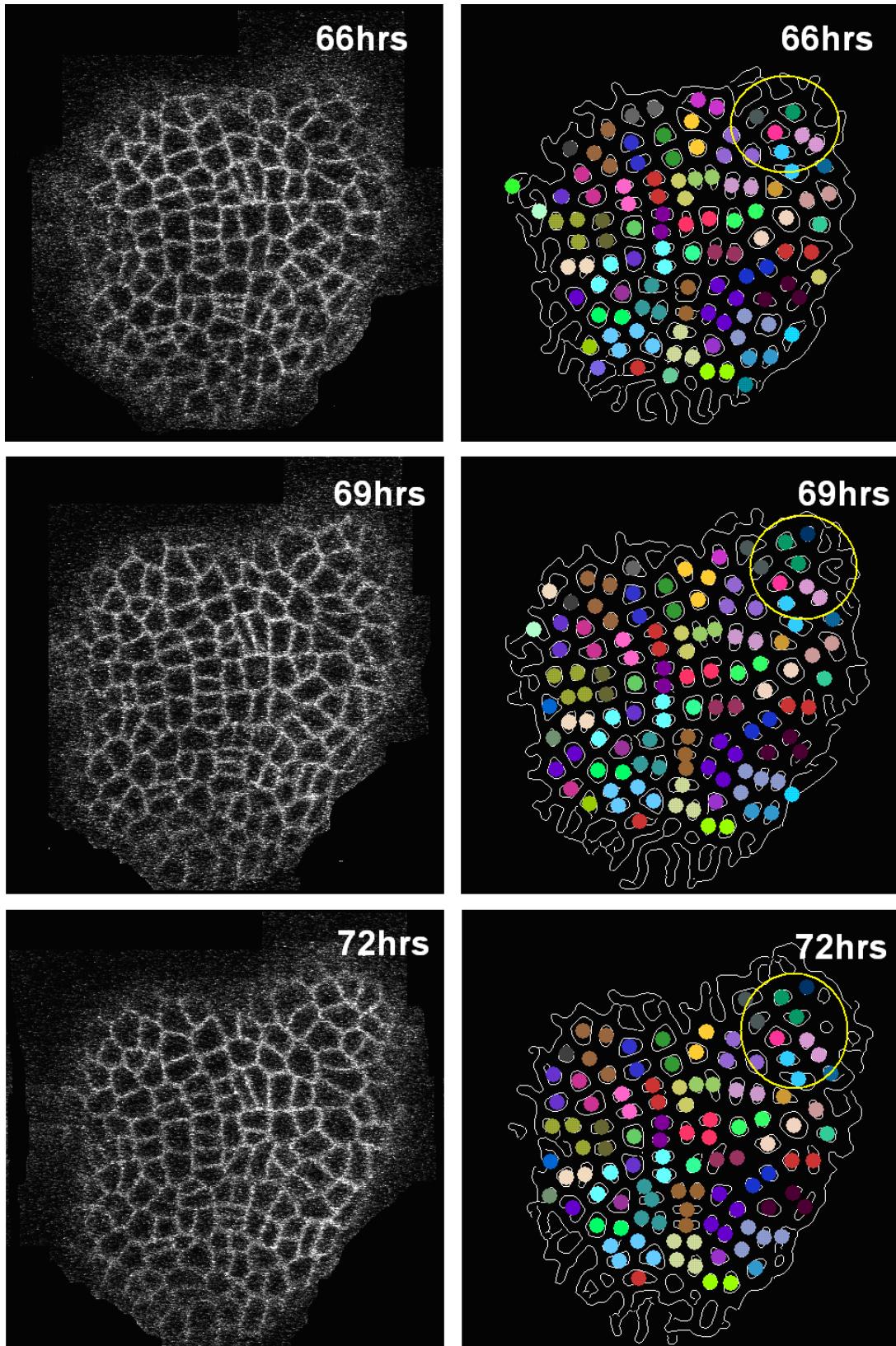
Supplementary Figure 4 (to be continued)



Supplementary Figure 4 (to be continued)

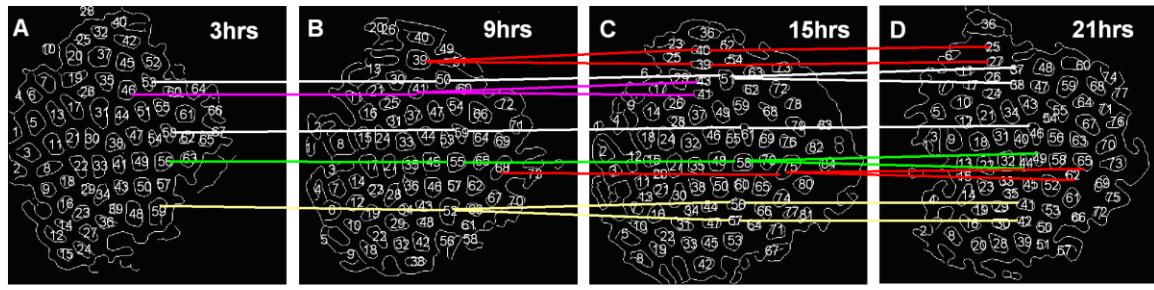


Supplementary Figure 4 (to be continued)

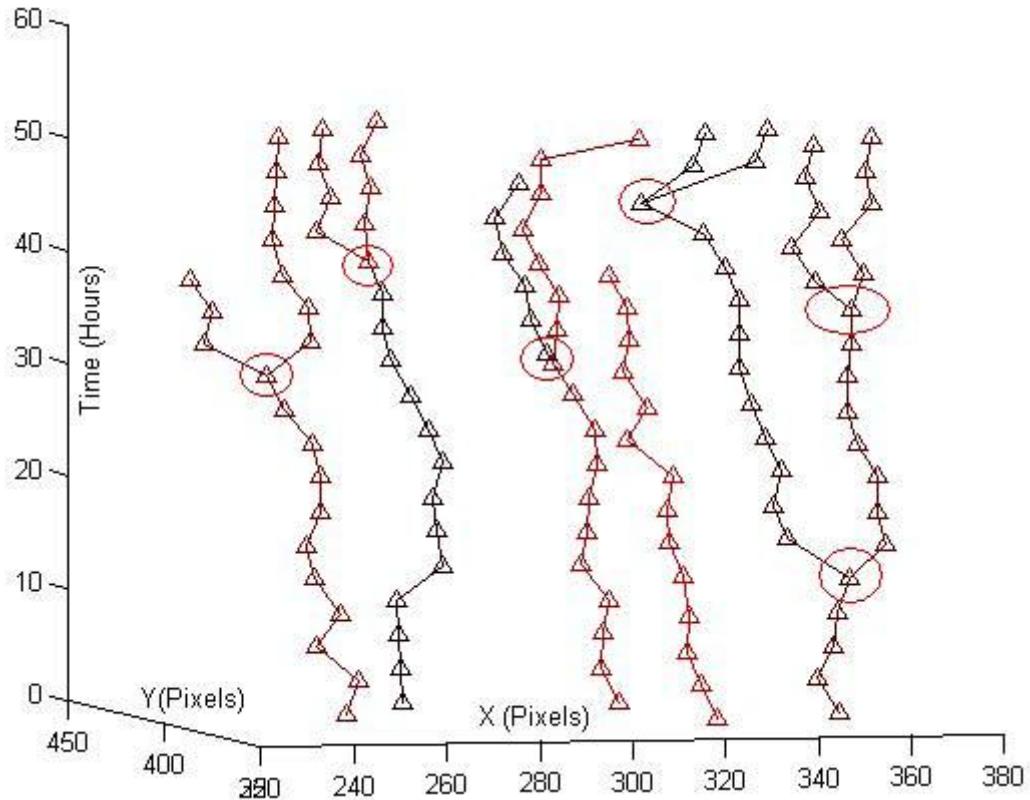


Supplementary Figure 4 (to be continued)

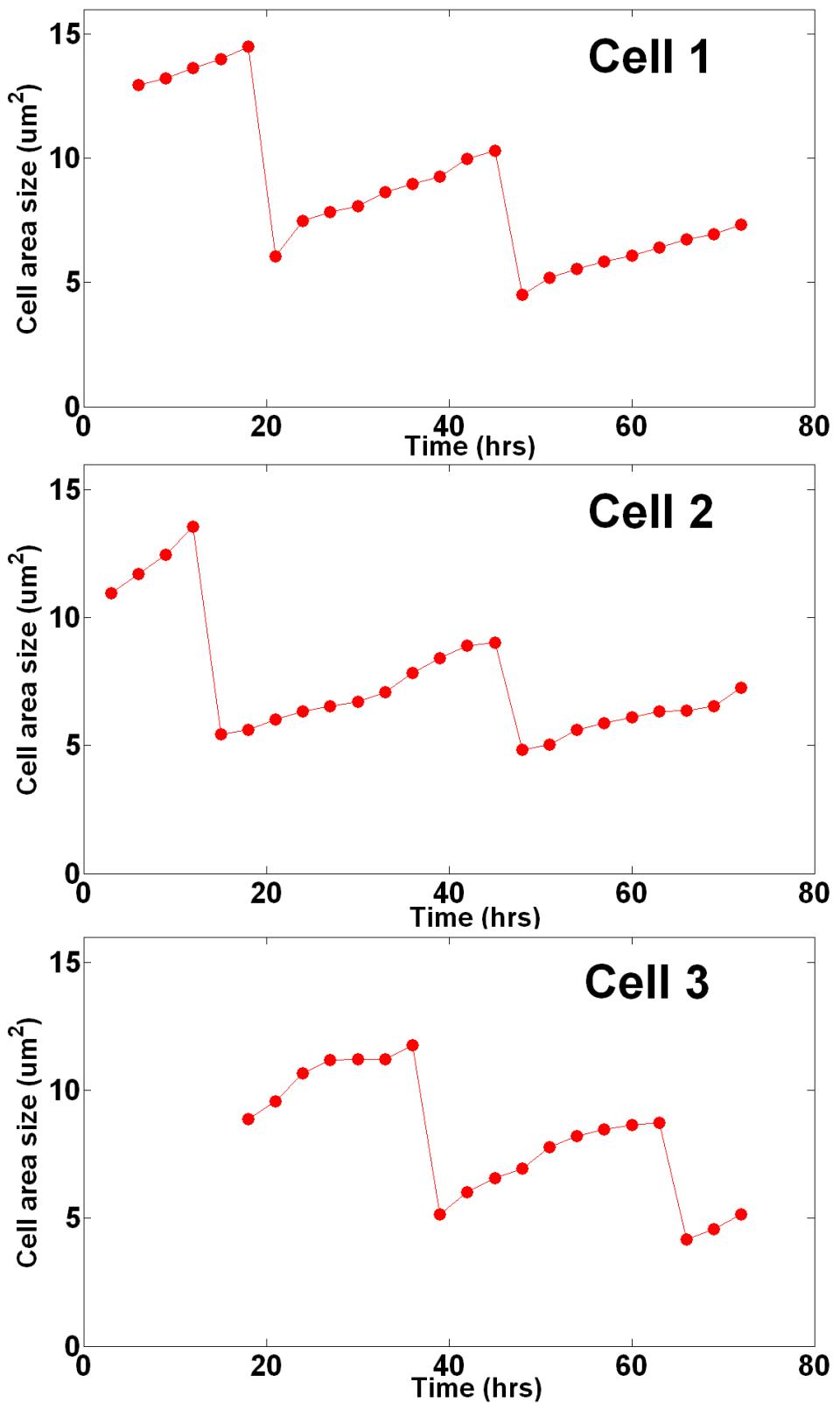
Supplementary Figure 4: A complete time series showing consistency in cell tracking. The computed cell lineages of SAM cells, across 24 time instants with successive time points separated by approximately 3 hours (a total of 72 hours) have been shown. Left panels show confocal images and the right panels show corresponding computationally-segmented images. Cells of the individual lineages are color coded, with the primordial regions denoted by yellow circles.



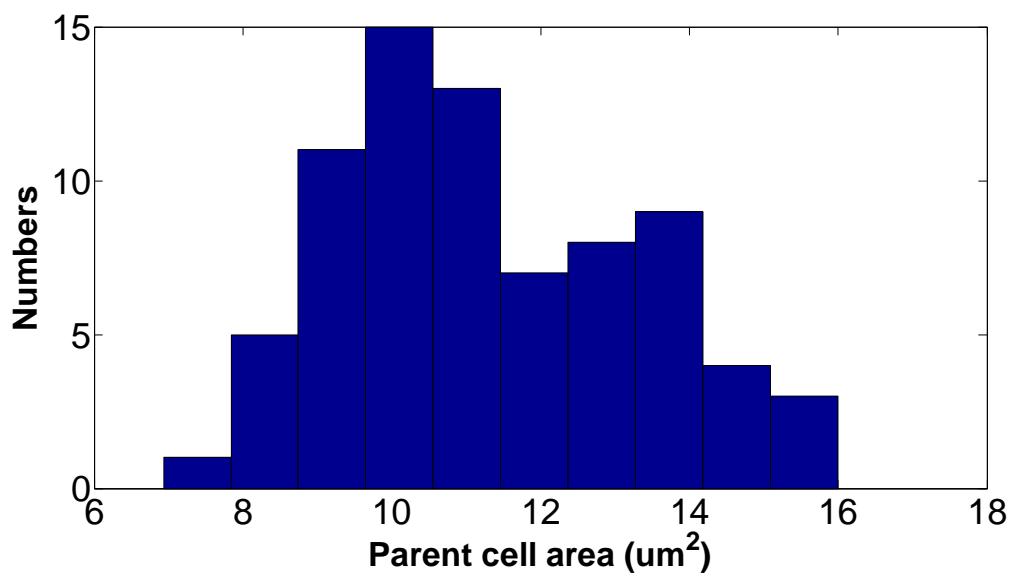
Supplementary Figure 5: Cell lineage tree. The cell lineages along 4 time instants are denoted for different cells denoted by color-coded lines. Every cell has been denoted with a number and the divided daughter cells are denoted by the same number.



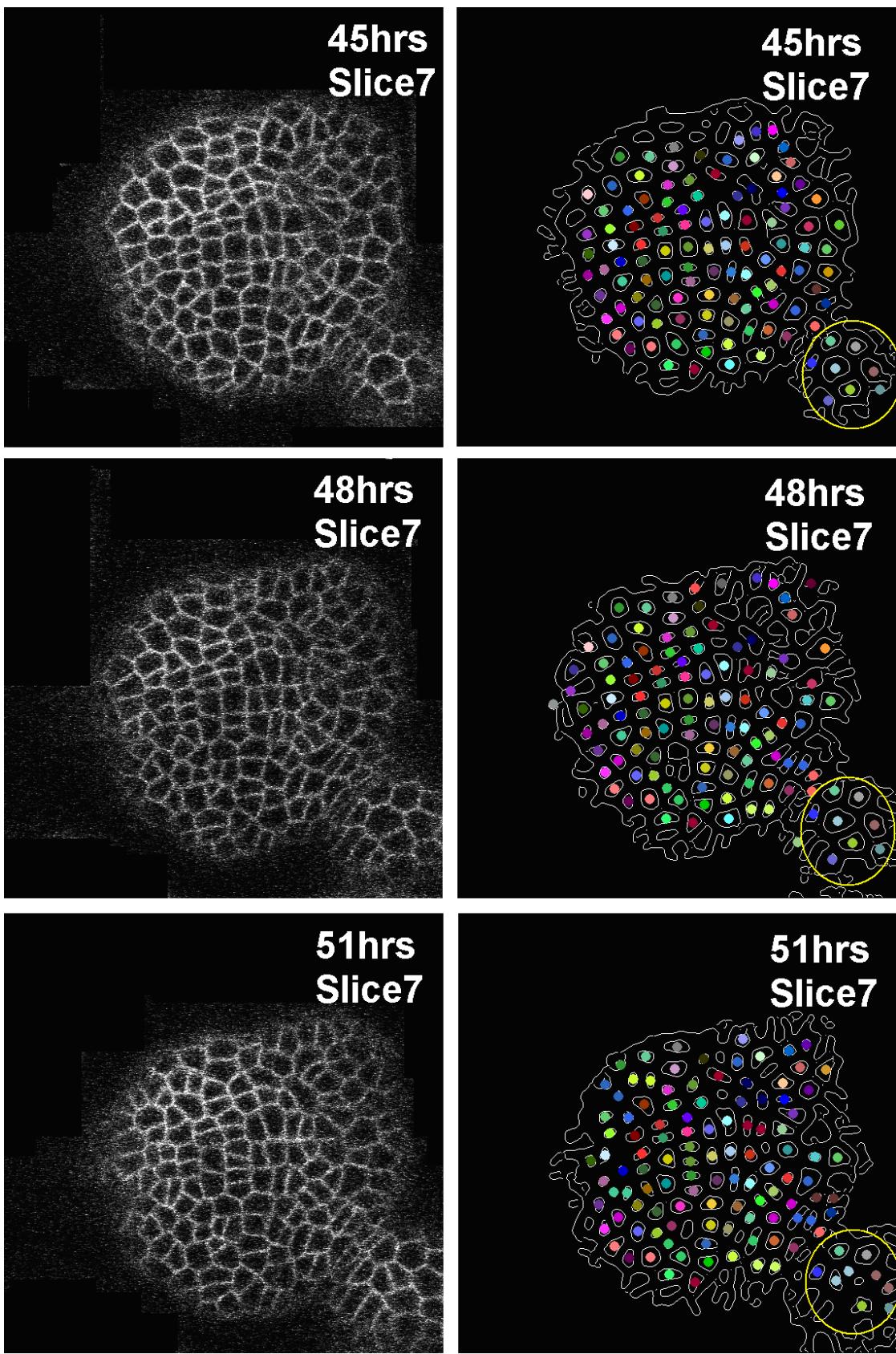
Supplementary Figure 6: Representation of a 3D cell lineage tree. The tracks of 5 cells along 18 time instants (i.e., 54 hours) have been shown to reveal 6 cell division events and one of them dividing twice.



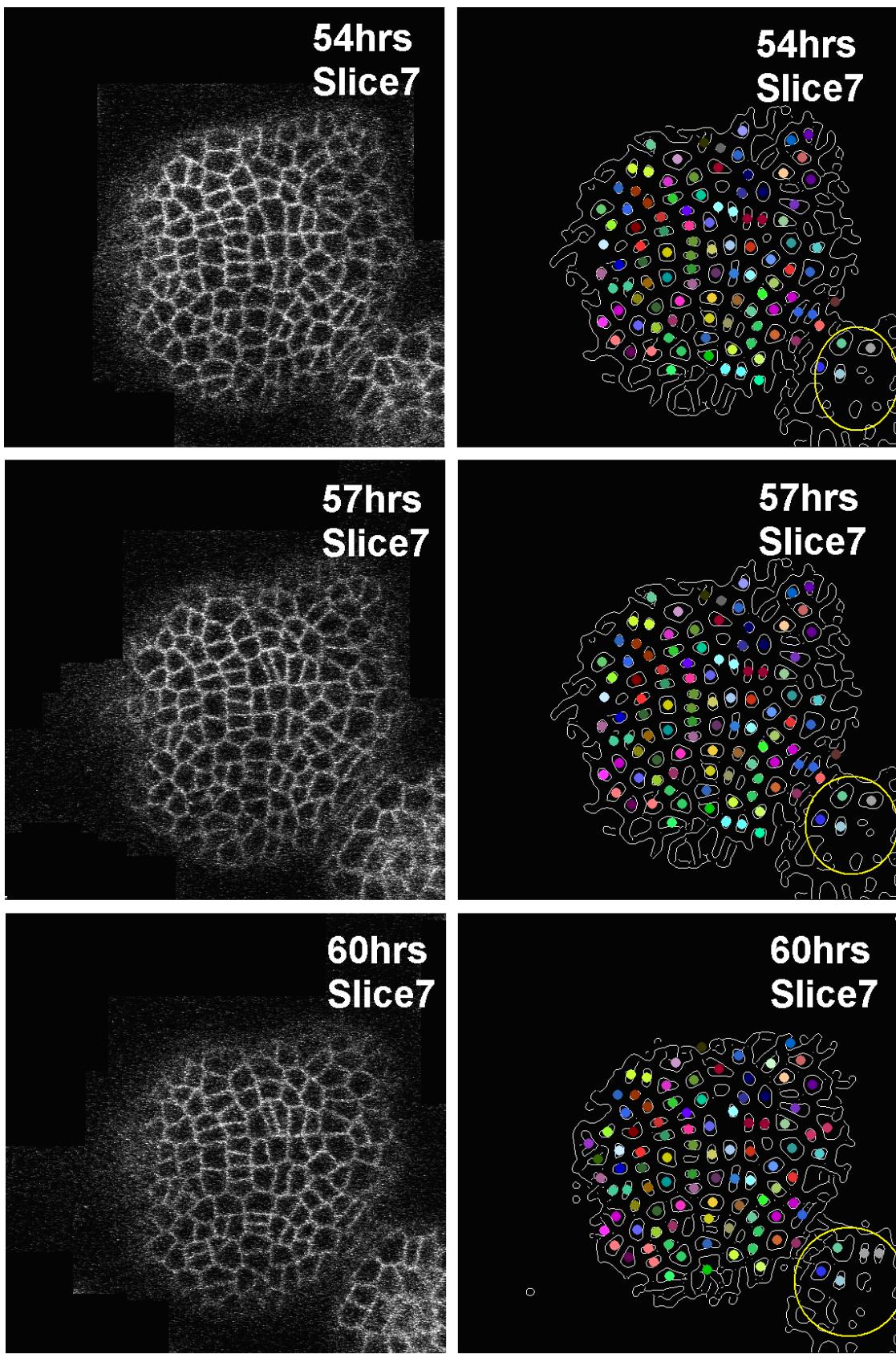
Supplementary Figure 7: Cell growth dynamics of representative examples of three cells prior to and after cell division events.



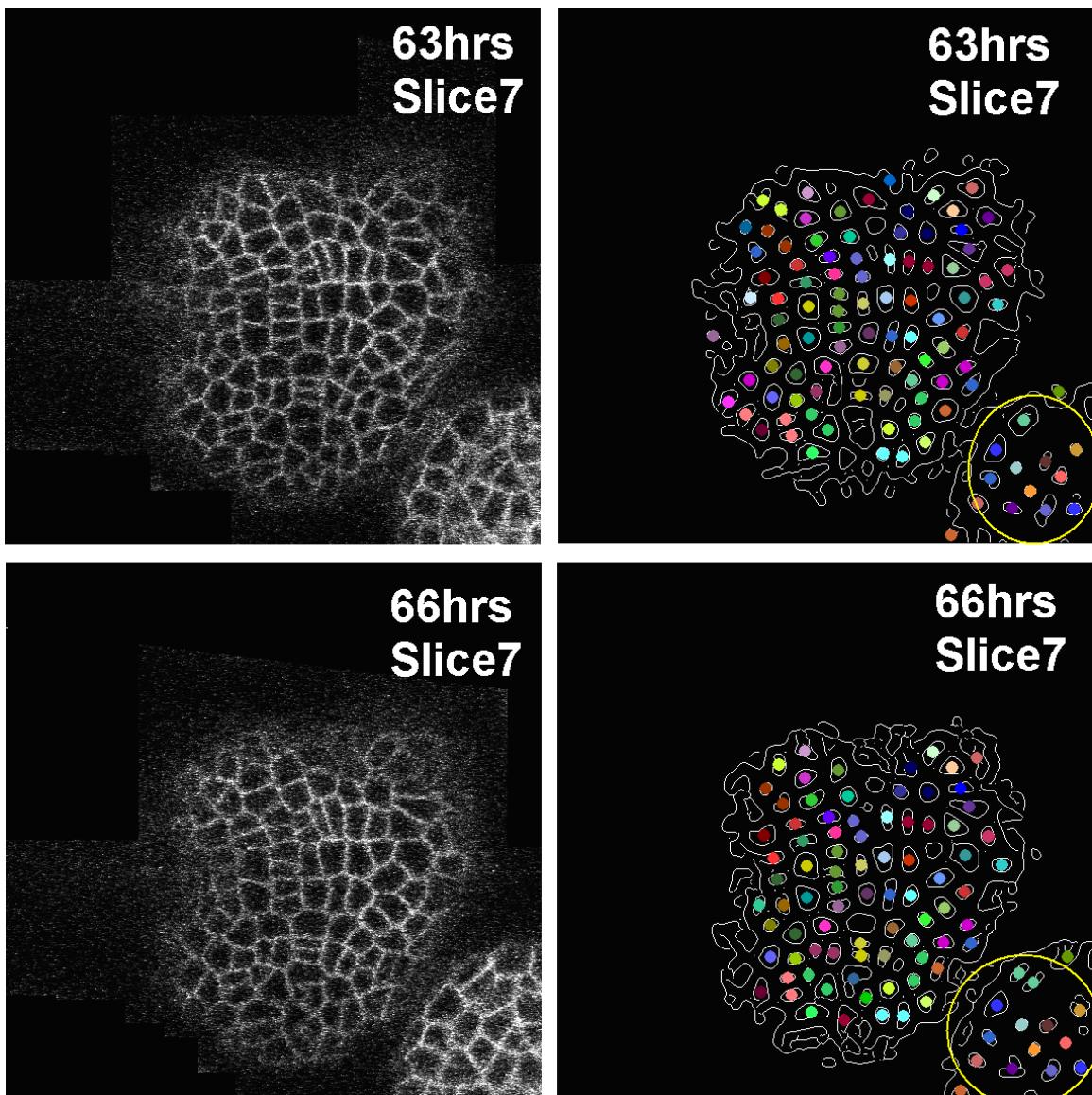
Supplementary Figure 8: A histogram showing variation in cell size of parent cells at the time of cell division.



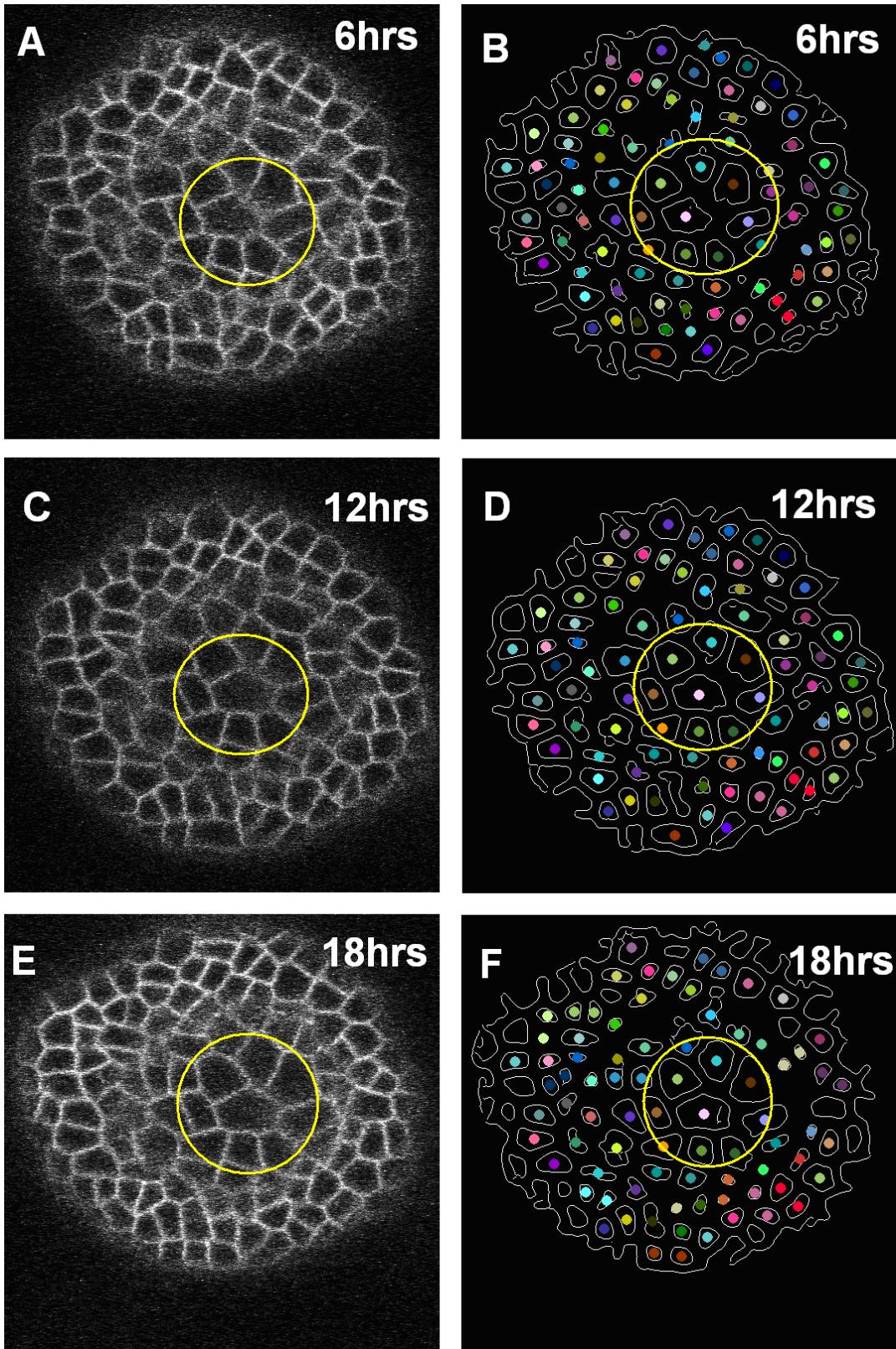
Supplementary Figure 9 (to be continued)



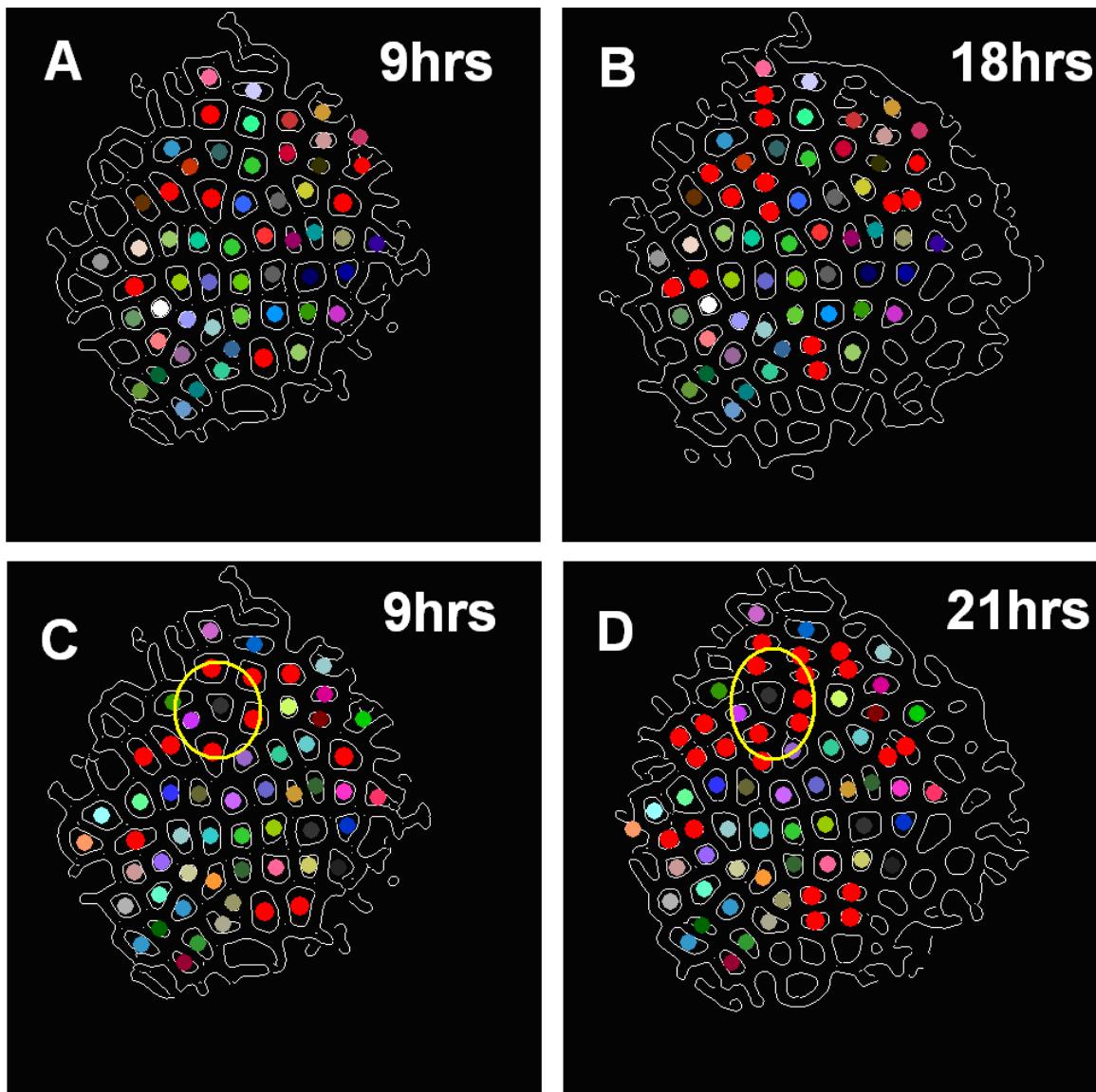
Supplementary Figure 9 (to be continued)



Supplementary Figure 9: Performance of cell tracker in high curvature regions. The cell lineage tracks (right panel) of cells that become part of the developing organ primordia are shown for a time lapse data of 24hours.



Supplementary Figure 10: Tracking output for cells of the L2 layer denoted by yellow circles.



Supplementary Figure 11: Cell tracking results for time interval of 9hours and 12hours.
In C and D there are 4 neighboring cells divided simultaneously in the yellow circled
region.