

## Supplementary Information: Modeling the finger instability in an expanding cell monolayer

Victoria Tarle<sup>1</sup>, Andrea Ravasio<sup>2</sup>, Vincent Hakim<sup>3</sup>, Nir S. Gov<sup>1</sup>

<sup>1</sup> *Department of Chemical Physics,  
Weizmann Institute of Science, Rehovot 76100, Israel*

<sup>2</sup> *Mechanobiology Institute,  
National University of Singapore, Singapore*

<sup>3</sup> *Laboratoire de Physique Statistique, CNRS,  
Université P et M Curie, Université Paris Diderot,  
Ecole Normale Supérieure, Paris, France*

## I. SIMULATION ALGORITHM

### A. Time integration

Each cell in our simulations is a point-like particle, with coordinates  $(x, y)$ . In our simulation we assume different forces that are applied on the cells, depending on their relative locations and velocities (Eqs.1-6). At each time iteration (with time step  $DT$ ) the movement of each cell is propagated forward in time using a standard Euler method

$$\begin{aligned}\vec{v}(t_i) &= \vec{v}(t_{i-1}) + DT \cdot \vec{a}(t_i) \\ \vec{x}(t_i) &= \vec{x}(t_{i-1}) + DT \cdot \vec{v}(t_i)\end{aligned}\quad (S1)$$

where  $\vec{v}(t_i)$  is the velocity vector of a cell at time  $t_i$  and  $\vec{a}(t_i)$  is the acceleration which is the sum of all the forces acting on the cell at time  $t_i$ .

The boundary condition is periodic in  $y$ -direction, while in the  $x$ -direction we used free boundaries (cells exit the simulation domain).

### B. Algorithm for finding the nearest-neighbors

For the forces calculation (Eqs.1-6) we require to know the list of nearest-neighbors of each cell. Our definition of nearest-neighbors (NN) is the following:

1. A NN is a cell that is at a distance less than some threshold  $Max\_NN\_Dist$ , which was set to be  $70\mu m$  [1].
2. The NN cell is not hidden by another cell.

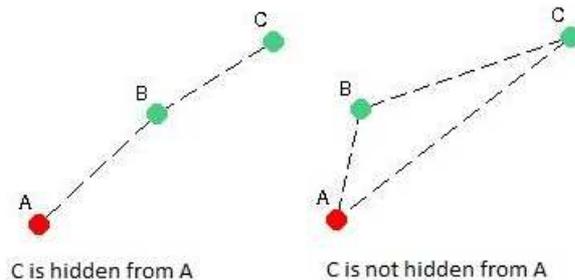


FIG. S1. Illustration of hidden and non-hidden nearest-neighbors (NN).

In Fig.S1, when finding the neighbors of A, the left panel shows cell C that is hidden by cell B, and hence it is not a NN of A (and vice versa). In the right panel C is a NN of A, since it is not hidden. We determine the criterion for being hidden through the angle  $\angle ABC$ : if this angle is larger than a certain value (taken to be  $125^\circ$ ), the further cell is considered hidden, and is not a NN.

For convenience of finding and using NN, the data structure of each cell includes an array of NN indices and an array of their relative angles. Each angle is defined from  $-\pi$  to  $\pi$  (Fig.S2), and the arrays are being ordered according to the angle (from smallest to largest).

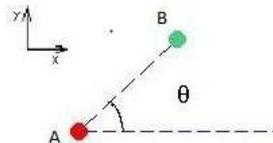


FIG. S2. The angle  $\theta$  is the angle of cell B in A's NN array.

As it is seen from Fig.S3, the NN of a cell is only a small fraction of the cells that are within the threshold distance of  $70\mu m$ , since most of them are hidden by closer cells. The reason for using such a large value of the threshold

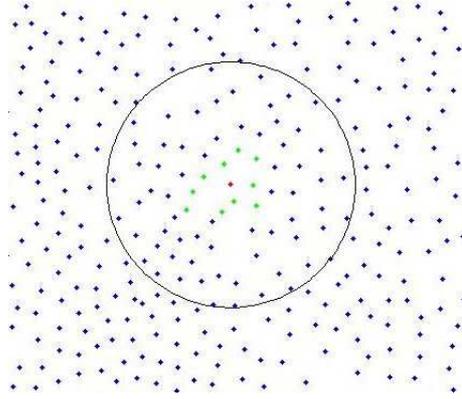


FIG. S3. Red - the cell of interest; Green - NNs of the red cell; Blue - all the rest of the cells. Black circle - the radius of maximum distance of NN ( $70\mu m$ ).

distance is in order to be able to connect cells when they are highly stretched and at low density, as occurs during the finger stretching.

The algorithm of finding NN is as follows:

We first find the candidates to be NN, by making an array of all the cells that are NN of the NN of this cell, from the previous iteration. This saves computing time, by limiting the search for the NN to a smaller group of cells. Since sometimes this is not completely accurate, and especially in cases of fingers merging, once every 50 iteration, all the cells within the threshold radius are being marked as candidates.

Each candidate is then checked by the following procedure:

- Check that the distance is less than maximum distance for NN.
- Calculate the angle  $\theta$  (see Fig.S2).
- If NN array is empty, add this cell to the array ( $\theta$  to angles array, and candidate index to indices array) and finish.
- Otherwise, find where in the angles array the new  $\theta$  should be.
- Repeat for angles larger and smaller than  $\theta$  (for  $|\theta - \theta_{in\ array}| < \frac{\pi}{2}$ ):
  - Calculate the angle between the cell of interest (A in Fig.S1), the candidate and the cell in NN array. The angle  $\angle ABC$  in Fig.S1 where B can be the candidate and C a cell in the existing NN array, or the other way around.
  - If  $\angle ABC > \Theta_{hiding}$  (there is hiding), check which one is closer to A. If candidate is closer - add it and delete the other one from the array. If the existing cell is the closer one - don't add the candidate and finish the procedure.
- If there is no hiding, the candidate is just inserted into the array, in the proper place to keep it sorted by angle.

There can be a situation, where the order in which the candidates are checked for being hidden or not, affects the final outcome. This is shown in Fig.S4: in this case since  $\angle ACD, \angle ADB > \Theta_{hiding}$  but  $\angle ACB < \Theta_{hiding}$ , if the candidate D is checked first, it is discarded, and then B is found not to be hidden. However this is wrong, and can be found by checking at the end of the whole procedure that each NN is mutual, i.e. A is NN of B and B is NN of A. In cases this is found not to occur, the two cells are considered not NN.

### C. Contour Line

In our model certain forces are applied only on the cells at the border contour. Hence we require to find the cells along the contour. The contour line is found using the ordered arrays of NN for each cell. The first cell on the contour is found as the most distant one:  $max(x)$  for right edge, and  $min(x)$  for left edge of the layer. The next cell along the contour, is the NN of the first cell that has an angle bigger than  $\pi/2$  (Fig.S5a). The next cells along the contour are the following cell in NN list of last border cell that was found, and so on (Fig.S5b). The algorithm is stopped when the first cell of the contour is reached (recall that the  $y$ -axis is periodic, hence contour line is periodic).

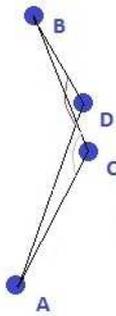


FIG. S4. Special case where checking the candidate D before B leads to the wrong conclusion that B is no hidden.

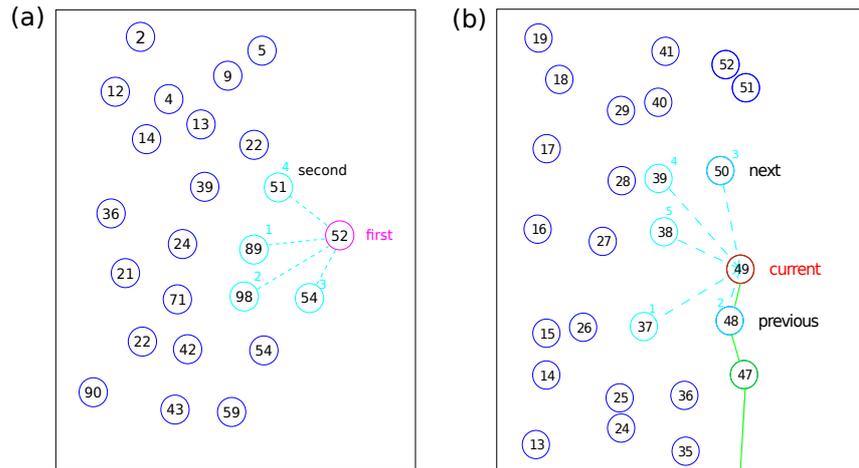


FIG. S5. (a) Example of finding the second cell of the contour: NN of the first cell in contour (52) are marked by light blue. The first NN that has angle bigger than  $\pi/2$  (as defined in S2) is 51 (b) Example of finding the next cell along the contour: The NN array of cell 49 is [37,48,50,39,38] (marked by light blue). Since the previous cell along the contour is 48, the next element in the NN array is 50, and hence it will be the next cell along the contour line array.

### 1. Extending cells

There are cases where a few cells extend in front of the main contour, as shown in Fig.S6. Such cases could in principle represent real single-cell-wide fingers, but due to numerical complications we decided not to treat them as a real part of the border contour. We therefore treated such extending cells only with the forces that are applied to bulk cells, i.e. without the  $F_{border}$  (Eq.7) and  $F_{cable}$ . In the example of Fig.S6 the cells [79, 21] are being extracted from the contour line, because the curvature is not well defined there.

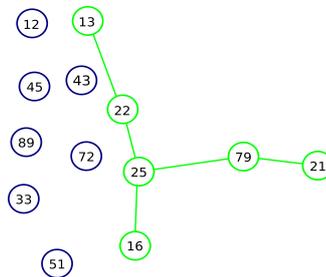


FIG. S6. Example of extending cells ([79, 21]) that are removed from the contour line.

Technically they are found as palindromes in the resulting contour line array. For example in the case of Fig.S6 initially the contour line array will be: [... 16,25,79,21,79,25,22,13 ...], and after deleting the palindrome it will become:

[... 16,25,22,13 ... ].

## 2. Detached groups

Sometimes single cells or even groups of cells become detached completely from the bulk cell layer (Fig.S7).

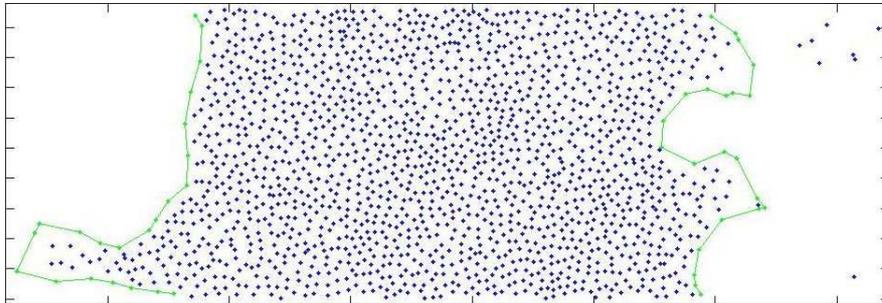


FIG. S7. Example of a detached group of cells (top right corner) from the bulk cell layer.

In this case, a cell in this detached group may initiate a false contour line around this group, instead of correctly identifying the border contour of the main layer. We identify such contours around a detached group by the number of times the contour line crosses the borders of the computation domain in the  $y$ -axis ( $min, mx$  values of  $y$ ): if it crosses an even number of times it is a sign of a detached group. In this case this contour is ignored, and the algorithm for finding the contour line is performed again. This procedure is repeated until all the detached groups are found.

In our calculations detached cells can join the bulk layer.

## D. Contour Line Derivatives

For calculation of the forces on the border cells (Eqs.6,7), we need to calculate the curvature and also up to second derivative of the curvature, so we need second and fourth order derivatives of the contour line. Because of the very discrete nature of the cells along the contour line, calculating these derivatives is not a trivial task. Hence two stages of smoothing are done, to get realistic estimates of the local curvature and it's derivatives.

We begin by calculating the discrete curvature along the contour (coordinate  $s$ ):  $\vec{H} \equiv \frac{d^2\vec{x}}{ds^2}$ . For curvature calculation, we calculate the tangent vector (first derivative):  $\vec{t}_i = \Delta\vec{x}_i/|\Delta\vec{x}_i|$ , where  $\Delta\vec{x}_i = x(x_{i+1} - x_i, y_{i+1} - y_i) = (\Delta x_i, \Delta y_i)$ ;  $\Delta s_i \equiv |\Delta\vec{x}_i| = \sqrt{\Delta x_i^2 + \Delta y_i^2}$ , for:  $i = 1 : n - 1$ . Since the contour is cyclic (as the whole  $y$ -axis), we have:  $\Delta\vec{x}_n = (x_1 - x_n, y_1 - y_n)$ .

Eventually we can write

$$\vec{t}_i = \Delta\vec{x}_i/\Delta s_i \quad (S2)$$

and the second derivative is taken in a similar manner

$$\vec{H}_i = \frac{\vec{t}_i - \vec{t}_{i-1}}{(\Delta s_i + \Delta s_{i-1})/2} \quad (S3)$$

In our simulation, we distinguish between border forces that are applied in regions with either concave or convex curvatures. For this purpose  $H_{mod} = |\vec{H}| \cdot sign(\vec{H})$  is calculated, where  $sign(\vec{H})$  is positive for concave and negative for convex curvature.

The sign of  $H$  is calculated using the fact that the contour is found always in the same direction, counterclockwise which is downward on the left border, and upward on the right border. This means that cells inside the bulk are always to the left from the border tangent vector  $\vec{t}$ . This is written as:  $sign(\vec{H}) = sign((\vec{n} \times \vec{H})_z)$ , using the normal vector:  $\vec{n}_i \equiv \vec{t}_i - \vec{t}_{i-1}$ . This completes the calculation of the function  $H_{mod}(s)$ , where the the coordinate along the contour  $s$  is defined to be:  $s_i = \sum_{k=1}^i \Delta s_k$ .

This curvature function is then smoothed using a weighted Gaussian method, taking as a “weight” the distances between the cells

$$H_{mod,smoothed}[i] = \left( \sum_k e^{-\frac{(s_i - s_k)^2}{2\sigma_G^2}} \right)^{-1} \cdot \sum_k \exp\left(-\frac{(s_i - s_k)^2}{2\sigma^2}\right) H_{mod}[k] \quad (S4)$$

This smoothing procedure is shown in Fig.S8a.

The final curvature vector is then given by

$$\vec{H}_{smoothed} = H_{mod,smoothed} \cdot \frac{\vec{H}}{|\vec{H}|} \cdot \text{sign}(\vec{H}) \quad (S5)$$

This smoothing procedure is also used to calculate the derivatives of the curvature:  $d\vec{H} \equiv \frac{d\vec{H}}{ds}$  and  $dd\vec{H} \equiv \frac{d^2\vec{H}}{ds^2}$ . In the discrete version these are written as:

$$dH_{mod}[i] = \frac{H_{mod}[i+1] - H_{mod}[i]}{\Delta s_i} \quad (S6)$$

$$ddH_{mod}[i] = \frac{dH_{mod}[i] - dH_{mod}[i-1]}{(\Delta s_i + \Delta s_{i-1})/2} \quad (S7)$$

These quantities are then smoothed using the same weighted Gaussian method:

$$ddH_{mod,smoothed}[i] = \left( \sum_k e^{-\frac{(s_i - s_k)^2}{2\sigma_G^2}} \right)^{-1} \cdot \sum_k \exp\left(-\frac{(s_i - s_k)^2}{2\sigma^2}\right) ddH_{mod}[k] \quad (S8)$$

which is then used to give the final vector, with direction of  $\vec{H}$

$$dd\vec{H}_{smoothed} = ddH_{mod,smoothed} \cdot \frac{\vec{H}}{|\vec{H}|} \cdot \text{sign}(\vec{H}) \quad (S9)$$

The smoothing of this derivative is shown in Fig.S8b.

As for the Gaussian Width ( $\sigma_G$ , Eqs.S4,S8) of the smoothing, we tried several values to determine the optimal value (Fig.S9). As a result the value of  $\sigma_G = 35\mu m$  was chosen.

## E. Proliferation

Cell proliferation is described in our simulation by each cell having a finite probability to undergo division into two daughter cells, at each time step. This probability is written as:  $P_{division} = DT/Division\_time$ , where  $DT$  is the computation time step, and  $Division\_time$  is the average time for a cell to divide. This division time depends on the local density of cells (Eq.8), which describes the inhibition of proliferation at high densities [2].

The stochastic division events are decided at each time step by a random number generator, such that if it is smaller than the division probability, division occurs. When division occurs a random angle is selected, and the cell become two cells at a separation of  $5\mu m$  oriented at the selected angle.

## II. FINGER DENSITY AS A FUNCTION OF THE EFFECTIVE BENDING MODULUS $\kappa$ OF THE BORDER CONTOUR

We varied the value of  $\kappa$  (Eqs.4,5) which we expect to change the average separation between fingers, according to [3]:

$$\lambda_{finger} = a_{cell} \sqrt{\frac{\kappa}{F_{max}/H_{max}}} \quad (S10)$$

where  $a_{cell}$  is the typical size of a cell ( $\sim 16\mu m$ ). As shown in Fig.S10 the average density of fingers indeed decreases as  $\kappa$  increases.

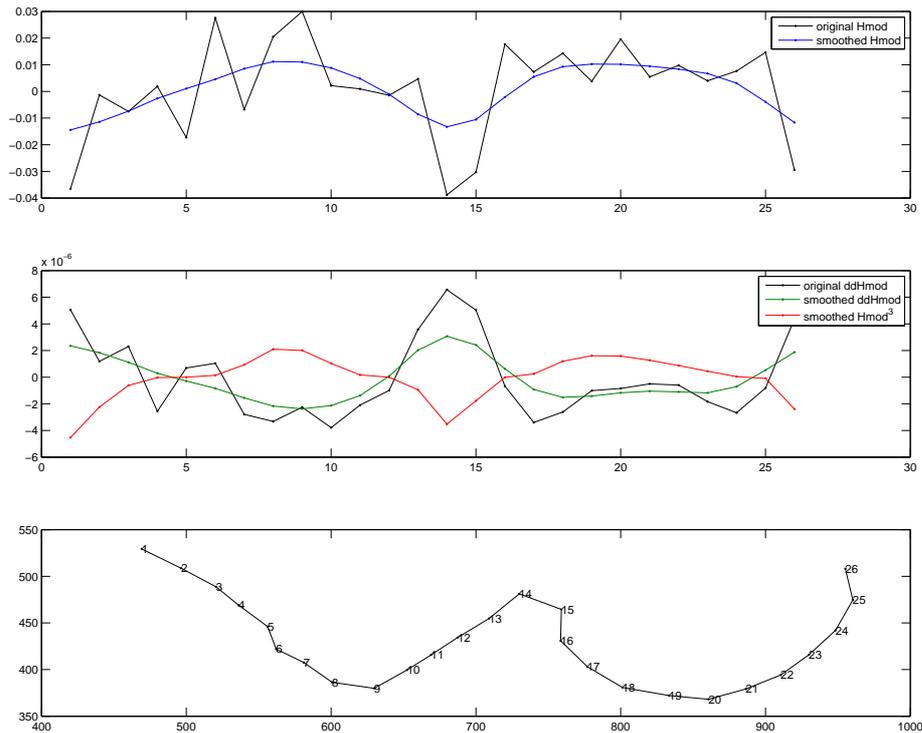


FIG. S8. (a) Illustration of the smoothing procedure of the curvature  $H$  (Eq.S4): black- original  $H_{mod}$  for the contour shown in (c), blue - smoothed  $H_{mod,smoothed}$ . (b) Illustration of the smoothing procedure of the second derivative of the curvature  $ddH$  (Eq.S8): black - original  $ddH_{mod}$  (second derivative) from the contour in (c), green - smoothed  $ddH_{mod,smoothed}$  second derivative, red - smoothed  $\frac{3}{2}H_{mod,smoothed}^3$  (which appears in the border force, Eq.6). (c) The original contour line, with a serial enumeration of the border cells.

### III. LEADER CELL DETACHMENT FOR VERY LARGE VALUES OF $F_{max}$ .

For very large values of  $F_{max}$  in Eq.7, we get detachments of cells from the tips of fingers, as illustrated in Fig.S7. We therefore used a non-linear attractive potential between the cells (third term in Eq.1) to inhibit such processes, but they still occur for very large values of  $F_{max}$ .

### IV. EFFECTS OF CELL PROLIFERATION

We tested the influence of the rate of cell proliferation on the finger formation, by running the simulation without any proliferation. We find (Fig.S11b) that fingers still initiate and grow even under no proliferation. The overall expansion of the layer at long times is of course inhibited under such conditions.

### V. ADJUSTMENT OF THE NOISE TO MATCH THE VELOCITY DISTRIBUTIONS FOR DIFFERENT VALUES OF $\beta$ .

When varying the value of the orientational interaction parameter  $\beta$  (Eq.4), we needed to adjust the value of the noise  $\sigma_0, \sigma_1$  (Eq.2), such that the bulk velocity distribution will still match the measured data. This is shown in Fig.S12.

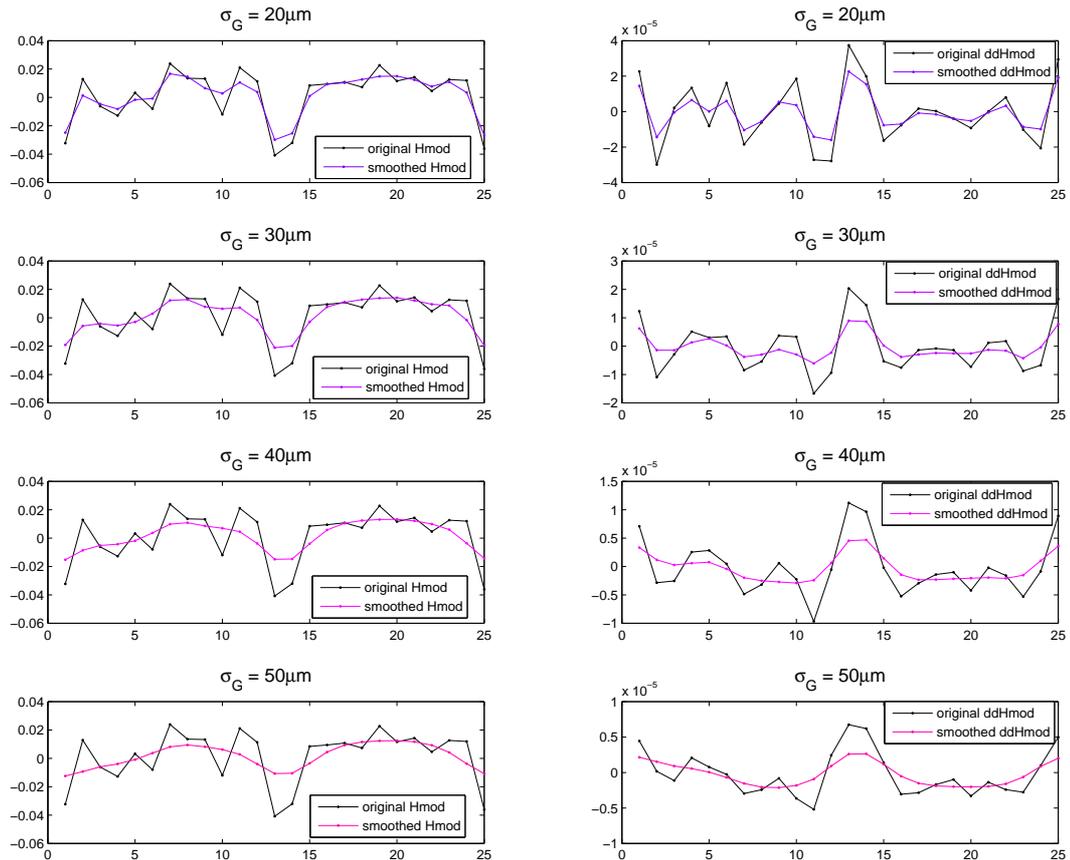


FIG. S9. Smoothing of  $H_{mod}$  with different widths  $\sigma_G$  of the Gaussian weight function (Eqs.S4).

## VI. TANGENT CORRELATION LENGTH ALONG THE CONTOUR, AND THE BULK VELOCITY CORRELATION LENGTH.

We calculated the tangent-tangent correlation along the border contour, which is shown in Fig.S13a for various values of the cable force  $F_{cable}$ . We find that the initial decay of this correlation along the contour can be fitted to an exponential form, from which we can then define the tangent correlation length  $\xi_{tangent}$  (Fig.3c,4c).

Similarly, we calculated the velocity-velocity correlation inside the bulk of the layer (calculated for a layer that is confined on all edges), and plot it as a function of the distance between cells (Fig.S13b).

## VII. LEADER CELLS RECOGNITION

For the automatic counting and calculation of the leader cells' velocity, we used the following recognition criterion to identify as particle as a leader. This identification is based when the (smoothed) curvature function  $H_{mod}$  has a local minimum, and it is lower than a certain threshold value (taken to be  $H_{threshold} = -0.0075\mu\text{m}^{-1}$ ).

Finding the local minima of the curvature is done in the following way: Recall that  $H_{mod}[i]$  is the absolute value of the curvature with a sign that is positive for concave and negative for convex, at point  $i$ . The derivative:  $dH[i] = H_{mod}[i+1] - H_{mod}[i]$ , of  $H_{mod}$  (and since the contour is cyclic  $dH[end] = H_{mod}[1] - H_{mod}[end]$ ). Leaders are those cells for which  $dH[i-1] < 0$  and  $dH[i] > 0$  and  $H_{mod}[i] < H_{threshold}$ , where  $H_{threshold} < 0$ .

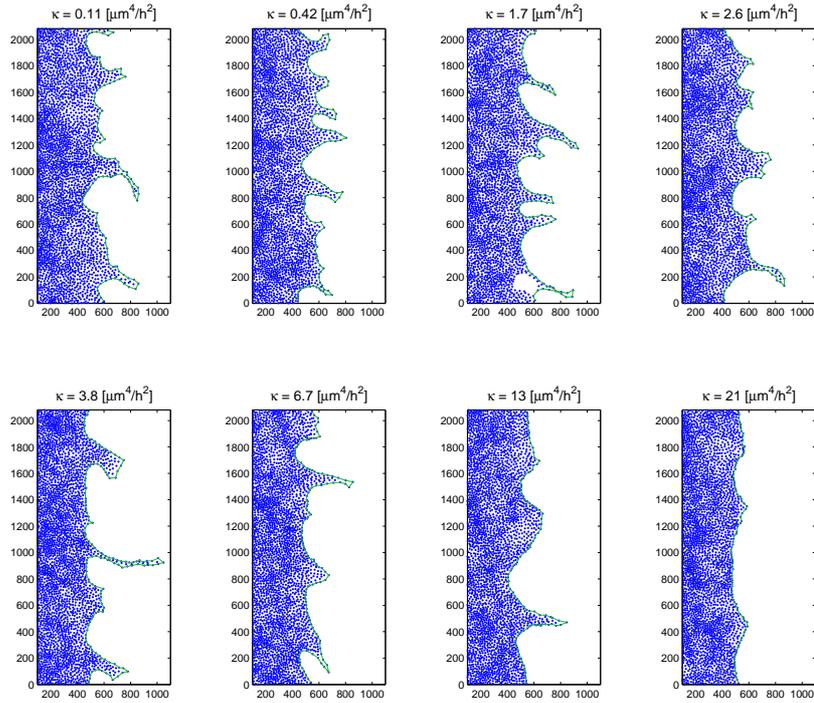


FIG. S10. Snapshots of the expansion of the layer and formation of fingers (at  $t = 25\text{hr}$ ), for different values of the effective bending modulus of the layer border  $\kappa$ . As  $\kappa$  increases we see that the density of fingers decrease, and their rate of growth decreases too, as expected from the curvature-motility feedback [3].

### VIII. PROPAGATING WAVES OF ACCELERATION.

In order to look for propagating waves, of the sort reported in [4–6], we plotted the kymographs of the  $x$ -component of the velocity averaged over the whole length in the  $y$ -direction (Fig.S14a,c). This was done for a case without fingers formation (Fig.S14a, case 2 of Fig.9), as well as with fingers (Fig.S14c, case 1 of Fig.9). The oscillations that are observed, correspond to periodic accelerations and deceleration of the expanding edge. By plotting the acceleration along the  $x$ -direction ( $\dot{v}$ , Eq.4), these waves become clearer. The wave velocity that we obtain from the slopes of these kymographs is  $2 - 3\mu\text{m}/\text{min}$ , which is roughly 2 – 3 times faster than the velocity of propagation observed in [4, 5]. It is likely that the properties of these waves are highly sensitive to the form of the inter-cellular potential, which is not known with high accuracy. While we do not explore further these waves in the present work, we are satisfied that the model we propose does have the capacity to exhibit such phenomena (in this context see also [7]).

### IX. EXPERIMENTAL METHODS

Wound-model in vitro experiments MDCK (madin-darby canine kidney) cells were cultured at  $37^\circ\text{C}$  in a humidified incubator with 5%  $\text{CO}_2$  in DMEM (Life Technologies) supplemented with 10% Fetal Bovine Serum (FBS; Life Technologies). Trypsin/EDTA (Life Technologies) subculture of cells was performed every 2/3 days when cells reached 80% confluence. Fibronectin (Sigma) was reconstituted in deionized water at a stock concentration of  $1\text{mg}/\text{ml}$  and serial dilutions of fibronectin were also prepared in deionized water. Surface of a cell culture treated 6-well plate was activated by a mild dose of UV light for 15 min to promote protein adsorption.  $1.5\text{ml}$  of fibronectin solution of varying concentration (1, 5, 10, 20, 40,  $80\mu\text{g}/\text{ml}$ ) was added to each well and incubated for 1h at  $37^\circ\text{C}$ . After washing away the excess of protein (3x), wells were passivated by 0.2% pluronic acid solution (Sigma) for 1 h at  $37^\circ\text{C}$ . Meanwhile, large cuts of PDMS membrane were adhered to a bacteriological petri dish and submerged in 0.2% pluronic acid solution for 1h at  $37^\circ\text{C}$  to passivate their sides. Both wells and PDMS cuts were washed with deionized water (5x) and allowed to dry for 10 min. PDMS cuts were carefully placed into the wells and allowed to adhere to the surface

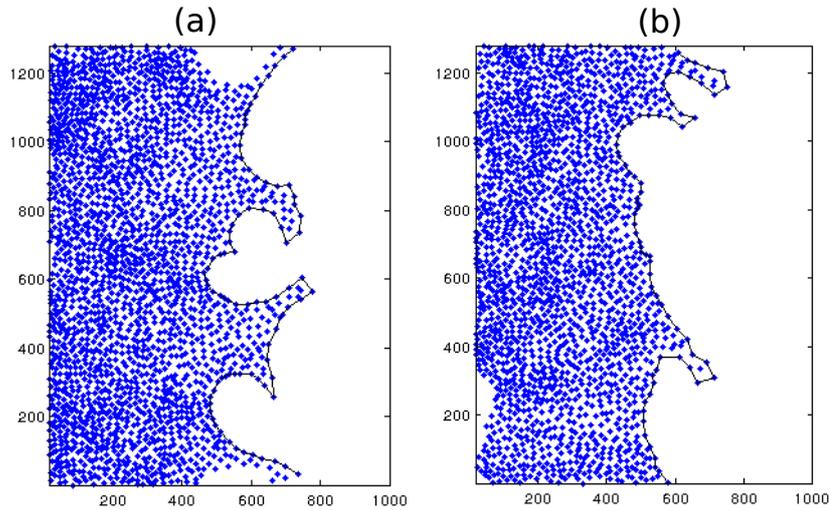


FIG. S11. (a) Snap-shot of an evolving cell layer, using the normal conditions (parameters given in Table 1), at  $t = 22$ hr. (b) Same as (a) but without cell proliferation.

for 1h at  $37^{\circ}\text{C}$ . Thereafter cells were seeded into the wells and cultured overnight to form a confluent monolayer. After careful removal of the PDMS cuts, samples were placed into a cell culture observation system (Nikon BioStation CT) and collective migration of cells were visualized by phase contrast at 4x. All live-cell experiments have been conducted at  $37^{\circ}\text{C}$  and 5%  $\text{CO}_2$ . The image segmentation of phase contrast images was performed using in-house MATLAB software. The algorithm applies a range filter on the raw image. The range filter gives high response at region with large intensity variations and low response at region with homogeneous intensity. Range filter was used to generate a binary mask corresponding to the epithelial tissue. After segmentation, the contour of the epithelial gap was extracted from the binary mask and average displacement of the tissue was calculated.

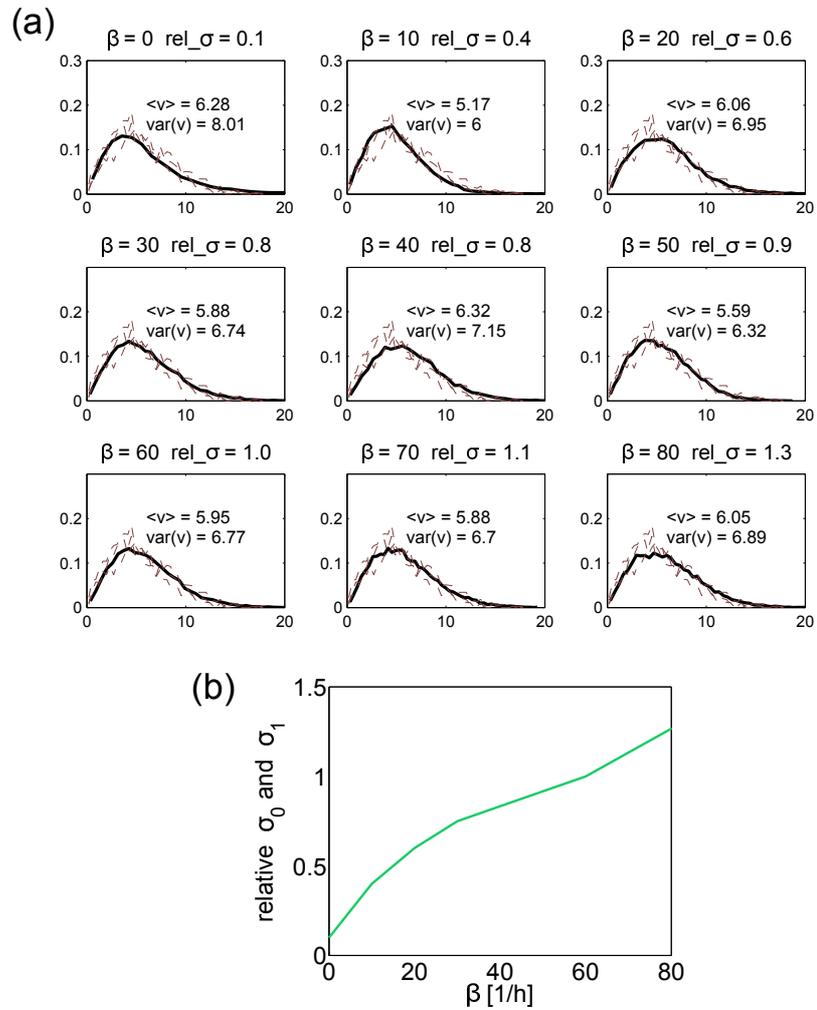


FIG. S12. (a) Velocity distributions for different values of  $\beta$ : the narrow lines give the experimental data [1], and the bold line gives the simulated distribution. Above each panel the value of  $\beta$  and the scaling factor of the noise  $\sigma$ , are given. (b) The scaling factors of the noise from (a) are plotted as a function of  $\beta$ .

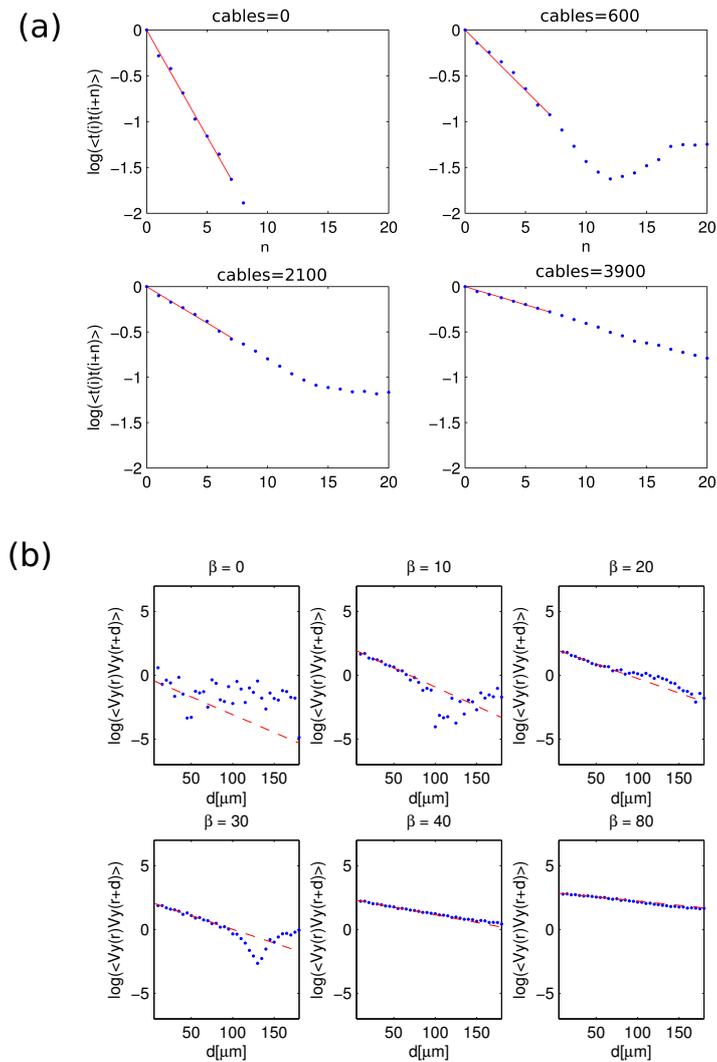


FIG. S13. (a) Tangent correlations along the border contour, for various values of the cable force  $F_{cable}$  (denoted above each panel). The simulation results (points) are fitted to an exponential decay (red line), whose slope give the correlation length  $\xi_{tangent}$ . (b) Velocity correlations inside the bulk, for various values of the orientational interaction parameter  $\beta$  (denoted above each panel). The simulation results (points) are fitted to an exponential decay (red line), whose slope give the correlation length  $\xi_{vv}$ .

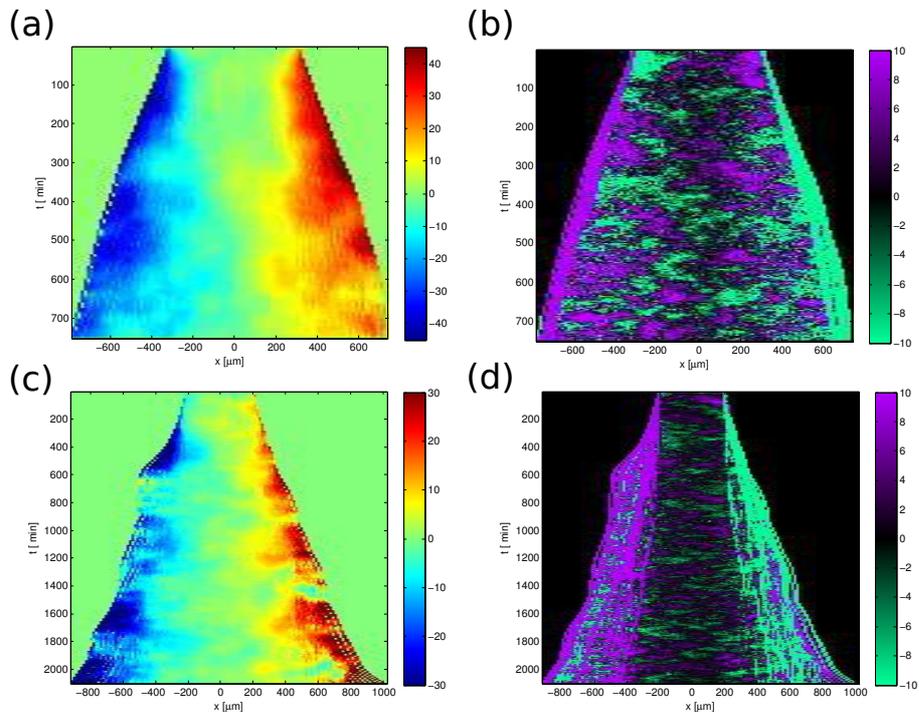


FIG. S14. (a) Kymograph of the velocity along the  $x$ -direction, averaged over the  $y$ -strips of the cell layer (color scale in units of  $\mu m/h$ ). This monolayer has no fingers formation, corresponding to case (2) of Fig.9. (b) Kymograph of the acceleration  $\dot{v}$  (Eq.4) along the  $x$ -direction, for the system in (a) (color scale in units of  $\mu m/h^2$ ). (c)-(d) Kymographs as in (a)-(b) but for a system that forms fingers (case (1) of Fig.9). There is clearly a central region of the layer that exhibits propagating waves, flanked by the fingers on either side.

- 
- [1] N. Sepúlveda, L. Petitjean, O. Cochet, E. Grasland-Mongrain, P. Silberzan, and V. Hakim, *PLoS computational biology* **9**, e1002944 (2013).
  - [2] A. Puliafito, L. Hufnagel, P. Neveu, S. Streichan, A. Sigal, D. K. Fygenson, and B. I. Shraiman, *Proceedings of the National Academy of Sciences* **109**, 739 (2012).
  - [3] S. Mark, R. Shlomovitz, N. S. Gov, M. Poujade, E. Grasland-Mongrain, and P. Silberzan, *Biophysical journal* **98**, 361 (2010).
  - [4] A. Zaritsky, D. Kaplan, I. Hecht, S. Natan, L. Wolf, N. S. Gov, E. Ben-Jacob, and I. Tsarfaty, *PLoS computational biology* **10**, e1003747 (2014).
  - [5] X. Serra-Picamal, V. Conte, R. Vincent, E. Anon, D. T. Tambe, E. Bazellieres, J. P. Butler, J. J. Fredberg, and X. Trepat, *Nature Physics* **8**, 628 (2012).
  - [6] Y. Matsubayashi, W. Razzell, and P. Martin, *Journal of cell science* **124**, 1017 (2011).
  - [7] M. Deforet, V. Hakim, H. Yevick, G. Duclos, and P. Silberzan, *Nature communications* **5**, 3747 (2014).