Biophysical Journal, Volume *113*

Supplemental Information

A Versatile Framework for Simulating the Dynamic Mechanical Structure of Cytoskeletal Networks

Simon L. Freedman, Shiladitya Banerjee, Glen M. Hocky, and Aaron R. Dinner

S1 Calculation of crosslinker head position during binding and unbinding

In this section, we describe how we update the binding state ($I_{1(2)}$ in Eq. 2) and position ($\vec{r}_{1(2)}$) of a crosslinker head. The binding states and positions of the two heads of a crosslinker are coupled only through the potential energy (Eq. 2).

We first discuss binding. An unbound crosslinker head with position $\vec{r}_u$ can attempt to bind to the closest point on each nearby filament link. Let $\vec{l}_i = \vec{r}_i - \vec{r}_{i-1}$, where $\vec{r}_i$ is the position of the i^{th} bead on the filament to which the link belongs. Then, we propose a bound state with binding point

$$\vec{r}_b = \begin{cases} \vec{r}_{i-1} & |\vec{l}_i| = 0 \text{ or } p \leq 0 \\ \vec{r}_i & p \geq 1 \\ \vec{r}_{i-1} + p\vec{l}_i & \text{otherwise} \end{cases} \tag{S1}$$

where $p = (\vec{r}_u - \vec{r}_i) \cdot \vec{l}_i$. Eq. S1 can be interpreted easily in a reference frame in which $\vec{l}_i$ is oriented vertically (Fig. S1A): if $\vec{r}_u$ is below the link, $\vec{r}_b = \vec{r}_{i-1}$; if it is above the filament then $\vec{r}_b = \vec{r}_i$; otherwise $\vec{r}_b$ is the intersection of $\vec{l}_i$ with the line perpendicular to $\vec{l}_i$ that passes through $\vec{r}_u$. If $|\vec{r}_b - \vec{r}_u| < r_c$, the changes in binding state and position are accepted with probability $(k_{xl}^{\text{on}} \Delta t) P_{xl,i}^{\text{off}\to\text{on}}$ (see main text, Crosslinkers).

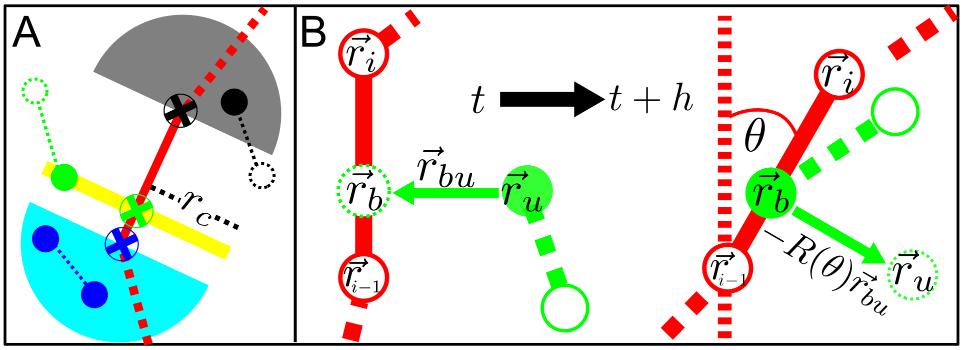

Figure S1: Position of crosslinker head upon binding or unbinding. (A) Any crosslinker head in the aqua, yellow, and gray areas (such as the filled blue, green, and black circles) can bind to the blue, green, and black binding points (circles with crosses), respectively. (B) The process by which a crosslinker generates an unbinding point (r_u) at time $t+h$ using its original displacement at time t when it snapped to the binding point r_b.

For unbinding, we do the following. At the time of binding (t), we record the displacement vector, $\vec{r}_{bu} = \vec{r}_b(t) - \vec{r}_u(t)$, and the vector connecting the ends of the filament link, $\vec{l}_i(t) = \vec{r}_i(t) - \vec{r}_{i-1}(t)$. At the time that we attempt unbinding ($t+h$), we determine the angle of rotation of the filament link:

$$\theta = \arccos\left(\frac{\vec{l}_i(t) \cdot \vec{l}_i(t+h)}{|\vec{l}_i(t)||\vec{l}_i(t+h)|}\right). \tag{S2}$$

Then, the position to which the crosslinker head tries to jump is

$$\vec{r}_u(t+h) = \vec{r}_b(t+h) - \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \vec{r}_{bu}(t) \tag{S3}$$

as shown in Fig. S1B. This jump is accepted with probability $(k_{xl}^{\text{off}}\Delta t)P_{xl,i}^{\text{on}\to\text{off}}$. The motivation for this scheme is that it ensures that a head that jumps onto (off) a filament link returns to its original position if it unbinds (rebinds) immediately. Detailed balanced consistent with Eq. 2 can thus be satisfied through the acceptance probabilities $(k_{xl}^{\text{on}}\Delta t)P_{xl,i}^{\text{off}\to\text{on}}$ and $(k_{xl}^{\text{off}}\Delta t)P_{xl,i}^{\text{on}\to\text{off}}$.

S2 Relaxation times scales

In this section, we present data on filament and network time scales that inform our choices of sampling frequencies.

S2.1 Decorrelation of filament angles

The evaluations of persistence length in Actin filaments exhibit predicted spatial and temporal fluctuations in the main text average over independent configurations of filaments. To determine the amount of time between independent configurations in a trajectory of a single filament, we evaluated the integrated autocorrelation time of the angles θ_i for $i \in [2\dots20]$ between links along a 21 bead filament. Fig. S2A shows the autocorrelation

$$R(\theta, s) = \frac{\langle\theta(t)\theta(t+s)\rangle - \langle\theta(t)\rangle^2}{\langle\theta(t)^2\rangle - \langle\theta(t)\rangle^2} \tag{S4}$$

where s is the time between realizations and the angle brackets represent an average over all 19 angles and all 1900 saved configurations. Fig. S2B shows the integrated autocorrelation time τ as a function of the simulation cutoff time t_{final}, where

$$\tau(\theta) = \int_0^{t_{\text{final}}} R(\theta, s)ds. \tag{S5}$$

For all choices of t_{final}, $\tau < 2$ s and therefore configurations that are separated by at least 2 s should be independent realizations with respect to angles between subsequent filament links.

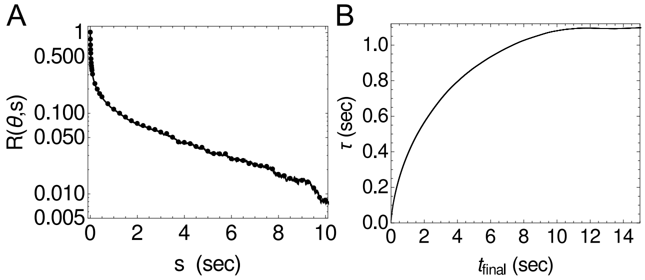

Figure S2: Estimation of the characteristic decorrelation time for persistence length measurements. (A) Decorrelation of angles between filament links for a 21 bead filament with $k_a = 1$ pN/μm, $l_a = 1$ μm, and $\kappa_B = 0.068$ pNμm^2. (B) Measurement of the integrated autocorrelation time τ for different values of the cutoff time t_{final}.

S2.2 Shear relaxation times

One extra parameter that must be set for shear simulations is the relaxation time (t_{relax})—i.e., the minimum time between strain steps for responses to be history independent. We probed this question computationally by determining if the parameter of interest (total potential energy of filaments and crosslinkers) varied significantly for different periods of relaxation between steps of $\Delta\gamma = 0.001$. Fig. S3 shows that while very small t_{relax} values do yield higher energies at equivalent strains, as t_{relax} is increased, the curves collapse for identical strains. In the shear simulations in the main text (Tunable elastic behavior of crosslinked filament networks), $t_{\mathrm{relax}} = 1$ ms (yellow curve).

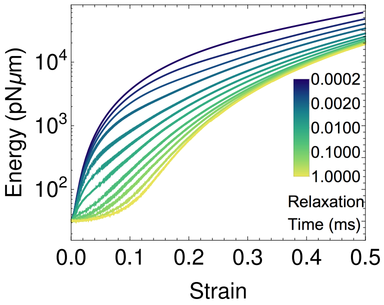

Figure S3: Total potential energy as a function of strain for various relaxation times. Simulation parameters, are otherwise identical to the shear simulations in the main text.

S3 Comparison with Cytosim

Cyotsim is a freely available C++ software package developed to simulate active polymer networks and described in (1). While AFINES shares many of the same features, for clarity we enumerate the technical differences.

- **The filament model.** AFINES uses a bead spring chain and Cytosim uses a chain constrained via Lagrange multipliers.

- **Attachment of motors and crosslinkers.** Cytosim uses a continuous-time Monte Carlo procedure (the Gillespie algorithm (2)) to calculate when a motor should attempt attachment to a filament, while AFINES attempts with the probability computed for each discrete timestep of fixed duration. In Cytosim, the attachment of a motor to a filament is not dependent on the distance from the filament, other than that it must be below a threshold, whereas in AFINES, a closer motor has a higher probability of attachment, due to detailed balance considerations.

- **Detachment of motors and crosslinkers.** Cytosim has a force dependent detachment of crosslinkers. This was not a necessary detail to reproduce the benchmarks shown in the results section, and detailed balance would require altering the motor and crosslinker dynamics, so we have not included it in the present version. We plan in the future to understand how this detail effects cytoskeletal networks in general and add it as an option to AFINES.

- **Capabilities present in one and not the other.** AFINES implements network shearing. Cytosim implements filament polymerization and depolymerization, microtubule asters, and spherical geometries.

To compare the two packages, we have used Cytosim to run the benchmarks associated with filament fluctuations (Fig. S4) and motility assays (Fig. S6, below). For the filament fluctuation benchmarks, shown in Fig. S4, we find that while Cytosim is able to yield nearly the correct persistence length of filaments, at long segment lengths it performs worse than AFINES, perhaps because it uses linearized versions of the angle forces (1).

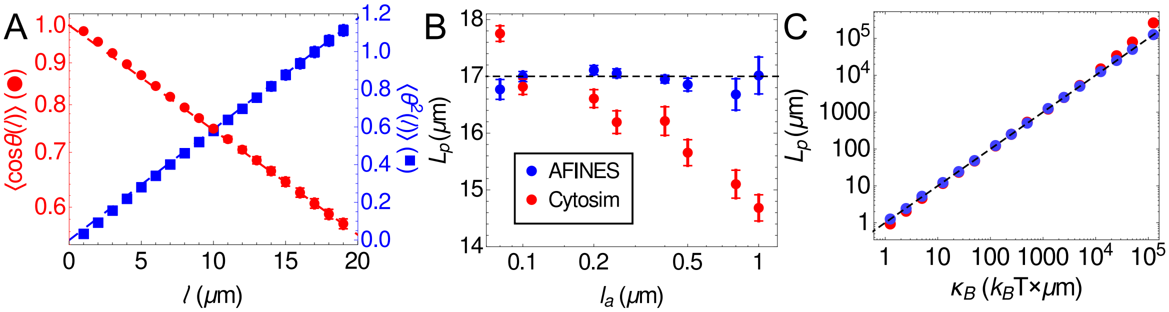

Figure S4: Measurements of persistence length for Cytosim filaments (red) compared with the same measurements for AFINES (blue). (A) Cosine correlation function and $\Delta\theta^2$ correlation function for 20 Cytosim fibers with $L_p = 17$ μm fluctuating for 2000 seconds. See Section 4.1 of the main text for details. (B) Measurement of L_p as function of segment length, l_a, using the fit to the first 5 data points of $\langle \Delta\theta^2 \rangle$ in (A). (C) Measurement of L_p as a function of input bending modulus for Cytosim and AFINES. Colors are the same as panel B.

S4 Parsing the energy in sheared networks

To further examine the source of the energy scalings shown in Fig. 5D, we measure the fraction of the total energy density w from each of its sources in the network, the stretching energy of filaments, the stretching energy of crosslinkers, and the bending energy of filaments, as shown in Fig. S5. In general, we find that shearing the network stretches and bends actin filaments, and also stretches crosslinkers, as in Fig. S5A-C. Fig. S5B-C show that, as crosslinkers become more stiff, more of the energy from the strain is concentrated on the filaments. Fig. S5A shows that for crosslinkers, the trend is not monotonic. When $k_{xl} < 100$ pN/μm, increasing crosslinker stiffness results in more energy in the crosslinkers, and in this regime, the scaling of $w(\gamma)$ increases monotonically. However, for $k_{xl} \geq 100$ pN/μm, the trend reverses, and the strain energy density concentrates on the filaments more than the crosslinkers, as seen in Fig. S5D-E. In this regime, the scaling of $w(\gamma)$ plateaus near the value 3.5, reflecting the prediction for the differential shear modulus in a strain controlled rheology experiment, $G = d^2w/d\gamma^2 \propto \gamma^{3/2}$ (3).

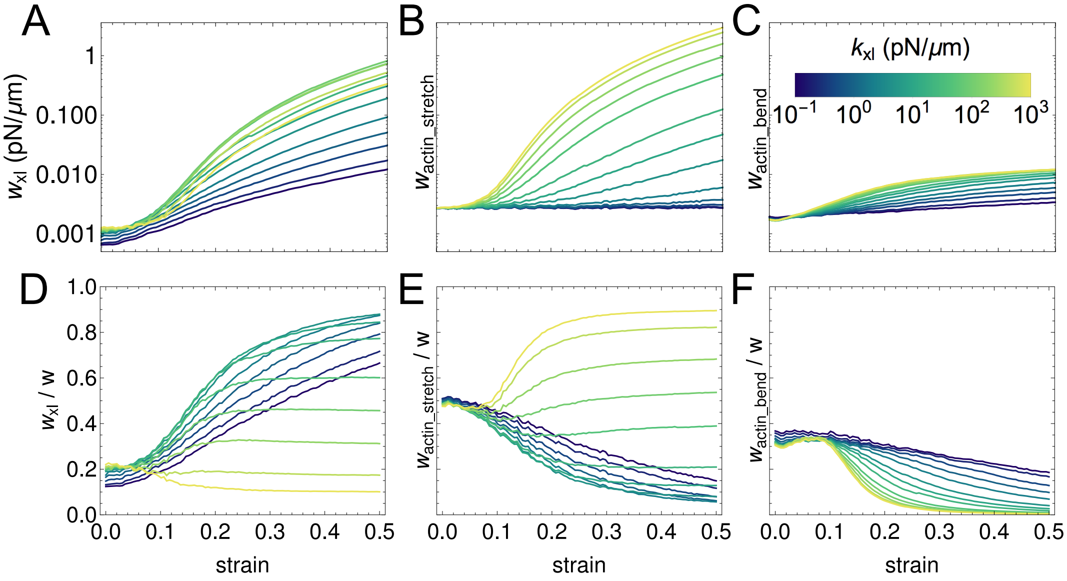

Figure S5: Absolute (A-C) and relative (D-F) energy contributions from crosslinkers stretching (A, D), filaments stretching (B, E), and filaments bending (C, F) for the sheared network discussed in Tunable elastic behavior of crosslinked filament networks.

S5 Comparison with Cytosim for motility assays

We also used Cytosim to simulate the motility assays described in the main text (Ensembles of motors interacting with individual filaments simulate actin motility assays). The results, shown in Fig. S6, are generally congruent with the results from AFINES in Fig. 6. We find that increasing motor density, filament length, and duty ratio increase longitudinal motion and decrease transverse motion of the filament (Fig. S6B), and makes the filament move more ballistically (Fig. S6C). Furthermore, the path persistence length plots (Fig. S6D) are nearly identical to the measurements obtained using AFINES. Thus, it is reassuring that the two models agree to this extent despite the differences in filament and binding implementations.

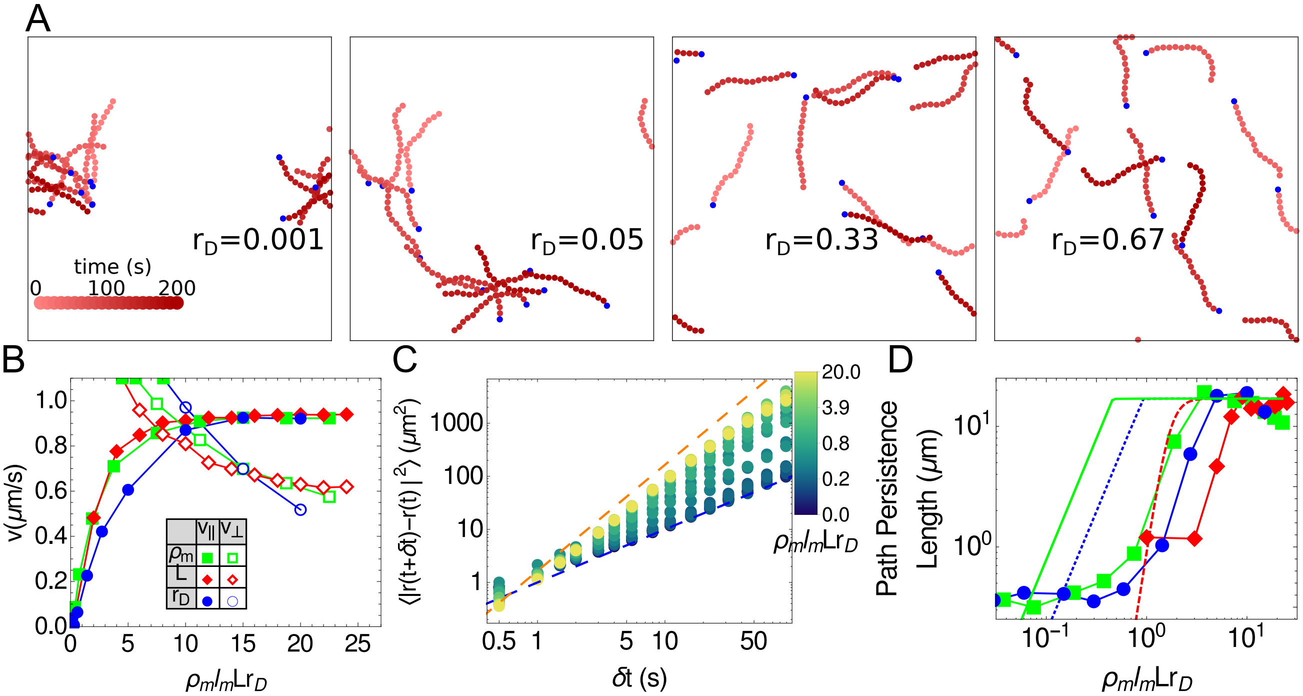

Figure S6: Motility measurements at varying motor density, filament length, and duty ratio generated using Cytosim. For a detailed description of this calculation see main text, Ensembles of motors interacting with individual filaments simulate actin motility assays.

S6 Procedure for quantifying contractility

An actin assay can be considered contractile if it has regions to which most of the actin aggregates. In an experiment with a limited field of view, the net flux of actin into the field of view is positive when the system is contractile. This flux corresponds mathematically to a negative value for the integral of the divergence of the velocity field over the area (4, 5). However, in our simulations, all particles' positions are known and there is no flux of material into or out of the simulation region owing to the periodic boundary condition. Thus the total divergence obtained by integrating over the simulation box must be zero. Nevertheless, we can still compute the density-weighted divergence to quantify contractility, as we now describe.

To ensure that the divergence is well-defined at all points, we first interpolate a continuous velocity field. When the data are experimental images, the velocity field is determined using Particle Image Velocimetry (PIV). Here, we take a similar approach, with the advantage that positions of actin beads are a direct output of the simulation, analogous to tracer particles in experiments. To this end, for each filament bead i with position $\vec{r}_i(t)$ at time t, we calculate the velocity by forward finite difference:

$$\vec{v}_i(\vec{r}_i, t) = \frac{\vec{r}_i(t+h) - \vec{r}_i(t)}{h}, \tag{S6}$$

where h is a suitable amount of time to characterize motion. We calculate the average velocity of each $(5\ \mu\text{m})^2$ bin. Similarly to PIV, we lower the noise further by setting a threshold, and only consider bins with at least n actin beads. We then interpolate the bin values with Gaussian radial basis functions (RBFs):

$$\vec{v}(\vec{r}) = \sum_{k=1}^{M} \vec{w}_k e^{-(|\vec{r}-\vec{r}_k|/\epsilon)^2} \tag{S7}$$

where M is the number of bins with at least n actin beads, ϵ is a constant related to the width of the Gaussian RBFs, and $\vec{w}_k$ are their weights. The optimal value for ϵ is generally close to the value of the average distance between RBFs (6); we found $\epsilon = 5$ μm and a threshold of $n = 10$ yielded a robust interpolation across many different actin structures. We use the scipy.interpolate.Rbf Python package to determine the weights (6). We calculate the divergence of the resulting field $dv_x(\vec{r})/dx + dv_y(\vec{r})/dy$ by using finite difference approximations for the derivatives of Eq. S7. Examples of this velocity field and the local divergence are shown in Fig. 7C and Fig. S7C.

As noted above, given $\nabla \cdot \vec{v}$, we quantify the contractility by the density weighted divergence, $\int \rho_a \langle \nabla \cdot \vec{v} \rangle dA$. In Fig. S7E we show an example where the density weighting has the effect of significantly increasing the magnitude of the areas with negative divergence. To understand how the contractility varies with length scale, we replace the integral with the sum over square regions

$$\sum_{k} \rho_a(\vec{r}_k) \langle \nabla \cdot \vec{v} \rangle_k dA \tag{S8}$$

and vary the size of the regions, $dA = dxdy$ (Fig. S7F). For the maximum size $dA = (50$ μm$)^2$ (yellow curve), the density weighted divergence fluctuates around 0 as expected from the zero actin flux. However for region sizes $dA \leq (10$ μm$)^2$, the values are consistently negative, indicating contractility; the curves decrease to a minimum before plateauing closer to 0, as seen in experiment (5). We also show, in Fig. S7G, that the trend of this order parameter is independent of the time scale h used to calculate the velocity in Eq. S6.

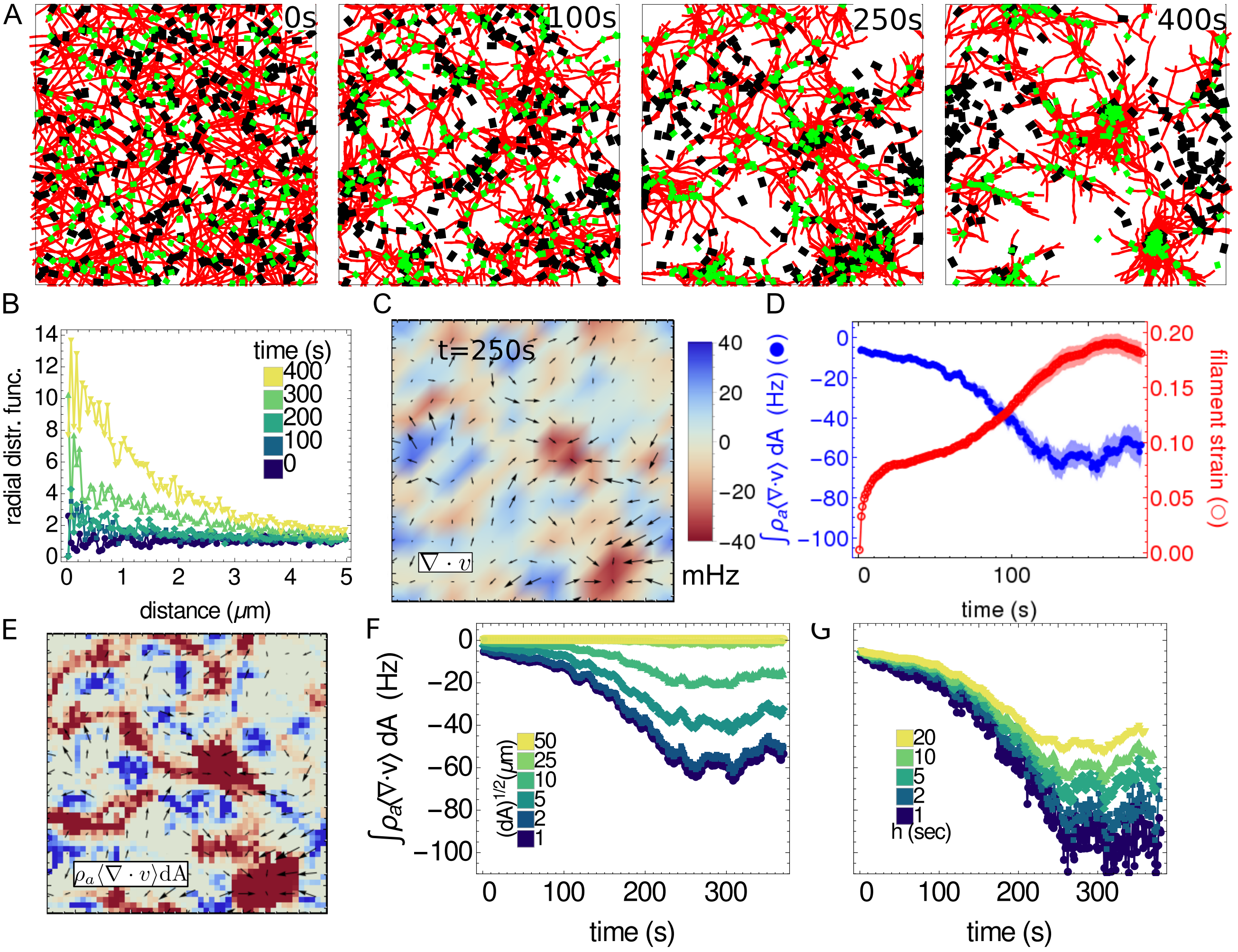

Figure S7: Calculation of density weighted divergence for a simulated contractile actomyosin network. (A-D) Identical to Fig. 7, but with $k_m^{\rm off} = 10\ {\rm s}^{-1}$, $k_m^{\rm end} = 1\ {\rm s}^{-1}$, and $\rho_m = 1\ \mu{\rm m}^{-2}$. In (A), all filaments and 10% of motors and crosslinkers are shown. (E) Same as (C), but the color is weighted by the actin density ρ_a. (F) Dependence of the density weighted divergence on the patch size used for integration, $dA = dxdy$, with $h = 10$ s. (G) Dependence of the density weighted divergence on the time scale h used in calculating the velocity of actin v in Eq. S6 with $dx = dy = 1\ \mu$m.

Supporting References

[1] Nedelec, F., and D. Foethke, 2007. Collective Langevin dynamics of flexible cytoskeletal fibers. New J. Phys. 9:427.

[2] Gillespie, D. T., 1977. Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81:2340–2361.

[3] Gardel, M., J. Shin, F. MacKintosh, L. Mahadevan, P. Matsudaira, and D. Weitz, 2004. Elastic behavior of cross-linked and bundled actin networks. Science 304:1301–1305.

[4] Murrell, M. P., and M. L. Gardel, 2012. F-actin buckling coordinates contractility and severing in a biomimetic actomyosin cortex. P. Natl. Acad. Sci. USA 109:20820–20825.

[5] Murrell, M., and M. L. Gardel, 2014. Actomyosin sliding is attenuated in contractile biomimetic cortices. Mol. Biol. Cell 25:1845–1853.

[6] Hetland, R., and J. Travers, 2001. SciPy: Open source scientific tools for Python: rbf - Radial basis functions for interpolation/smoothing scattered Nd data.