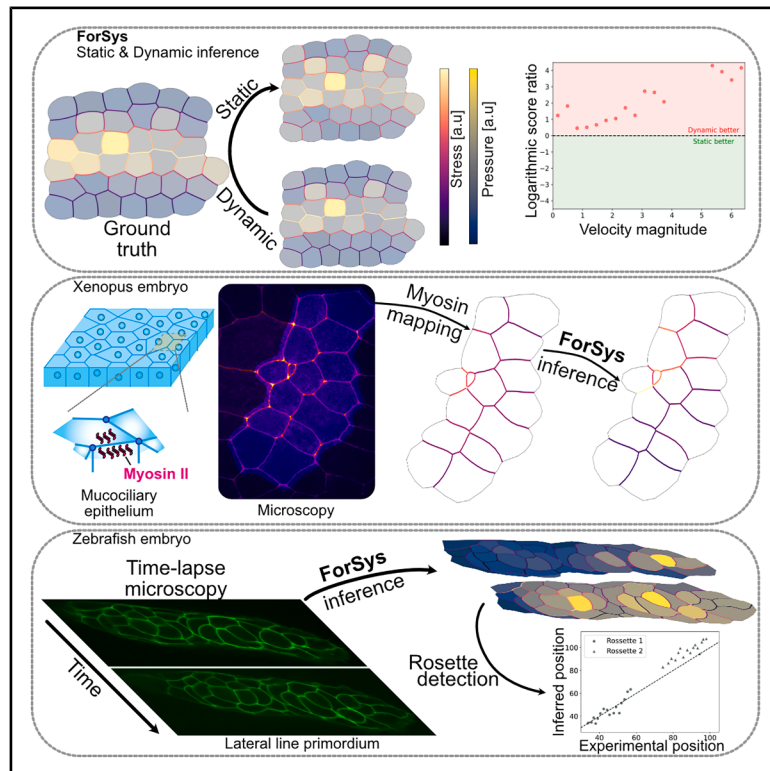


ForSys: Non-invasive stress inference from time-lapse microscopy

Graphical abstract



Authors

Augusto Borges,
 Jerónimo R. Miranda-Rodríguez,
 Alberto S. Ceccarelli, Guilherme Ventura,
 Jakub Sedzinski, Hernán López-Schier,
 Osvaldo Chara

Correspondence

osvaldo.chara@nottingham.ac.uk

In brief

Optical imaging

Highlights

- ForSys is an open-source tool to infer the mechanical states of dynamically evolving tissues
- ForSys dynamic modality outperforms static modality to recapitulate *in silico* ground truths
- ForSys predicts mechanical asymmetries linked to cell-type specific adhesion in the neuromasts
- ForSys is usable through a fiji-supported GUI, a command line interface, or through Python code



Article

ForSys: Non-invasive stress inference from time-lapse microscopy

Augusto Borges,^{1,2} Jerónimo R. Miranda-Rodríguez,^{1,3} Alberto S. Ceccarelli,^{4,5} Guilherme Ventura,⁶ Jakub Sedzinski,^{6,7} Hernán López-Schier,^{1,2,8} and Osvaldo Chara^{5,9,10,*}

¹Unit Sensory Biology and Organogenesis, Helmholtz Zentrum München, Munich, Germany

²Graduate School of Quantitative Biosciences, Ludwig Maximilian University, Munich, Germany

³Instituto de Neurobiología, Universidad Nacional Autónoma de México (UNAM), Boulevard Juriquilla 3001, Juriquilla, México

⁴Systems Biology Group, Institute of Physics of Liquids and Biological Systems, University of La Plata, La Plata, Argentina

⁵School of Biosciences, University of Nottingham, Sutton Bonington Campus, LE12 Nottingham, UK

⁶The Novo Nordisk Foundation Center for Stem Cell Medicine (reNEW), University of Copenhagen, Blegdamsvej 3B, 2200 Copenhagen, Denmark

⁷Department of Biomedical Sciences, University of Copenhagen, Copenhagen, Denmark

⁸Division of Science, New York University Abu Dhabi, Abu Dhabi, Saadiyat Island, United Arab Emirates

⁹Instituto de Tecnología, Universidad Argentina de la Empresa, Buenos Aires, Argentina

¹⁰Lead contact

*Correspondence: osvaldo.chara@nottingham.ac.uk

<https://doi.org/10.1016/j.isci.2025.113685>

SUMMARY

During tissue development and regeneration, cells interpret and exert mechanical forces that are challenging to measure *in vivo*. Stress inference algorithms have thus emerged as powerful tools to estimate tissue stresses. Yet, effectively incorporating tissue dynamics into these algorithms remains elusive. Here, we introduce ForSys, a Python-based software that infers intercellular stresses and intracellular pressures from time-lapse microscopy. After validation, we applied ForSys to the migrating zebrafish lateral-line primordium, revealing increased stress during the cell rounding that precedes mitosis and accurately predicting the onset of epithelial rosetogenesis. We further used ForSys to study neuromast development and uncovered mechanical asymmetries linked to cell type-specific adhesion. The software performs both static and dynamic stress inference, supports command-line use, scripting, and a user-friendly graphical interface within Fiji, and accepts segmentation inputs from EPySeg and Cellpose. This versatility of ForSys enables the analysis of spatiotemporal patterns of mechanical forces during tissue morphogenesis *in vivo*.

INTRODUCTION

Recent advances in experimental techniques have reignited interest in exploring the mechanical properties of biological tissues, commonly referred to as tissue rheology. These methods have facilitated precise and quantitative measurements of tissue mechanical parameters. For example, implanted deformable magnetic droplets have been used to determine the elastic properties along the zebrafish anteroposterior axis during body elongation^{1,2} and presomitic mesoderm differentiation.³ Similarly, the application of optical traps has enabled the controlled deformation of cell membranes, thereby facilitating the study of viscoelastic properties during *Drosophila* development.⁴ Laser ablation experiments have also been employed in various systems to probe cortical tension by measuring the recoil of cell junctions upon laser cutting.^{5–7} Despite their importance, these experimental methods often come with significant drawbacks. They can be costly and necessitate specialized equipment, posing implementation challenges for many researchers. Moreover, these techniques might not be conducive to long-term imaging, poten-

tially disrupting the normal development and, in some cases, leading to the destruction of the sample. Hence, there is a pressing need for alternative approaches to overcome these limitations while still delivering accurate and non-invasive measurements of tissue rheology.

Computational methods offer a promising alternative solution, enabling the cost-effective and straightforward implementation of tissue mechanical characterization *in vivo*.⁸ Inference techniques that use readily available microscopy images can infer the effective stress of a system based on the geometry of the cellular junctions. A key aspect of this approach centers on tricellular junctions (TJ), where three cells converge.⁹ The underlying framework relies on one major assumption: mechanical equilibrium is maintained at each TJ. The strength of these models lies in their simplicity, reducing the estimation of intercellular stresses to the solution of an overdetermined system of linear equations.^{10,11} One of the first implementations of the force-inference approach has been CellFIT,¹⁰ which enables the estimation of stresses from microscopy images. While CellFIT provides accurate estimates in static tissues, its



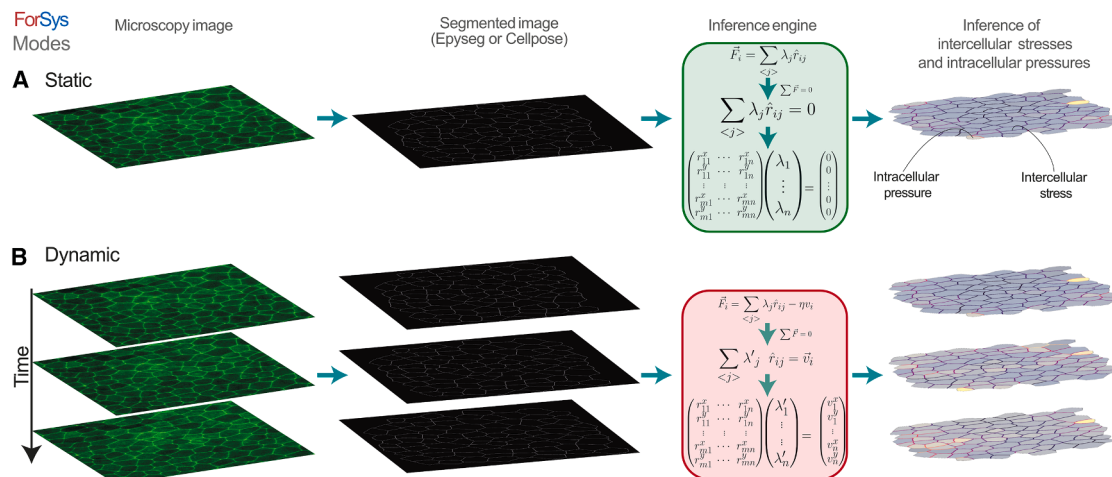


Figure 1. Force inference modalities of ForSys

(A) The static inference is performed on a microscopy image by creating a segmented representation of the tissue. Then, ForSys reads it and builds the system of equations according to the geometrical properties of the tissue, assuming that each vertex in the tissue is in mechanical equilibrium. Lastly, the system is solved, and the intracellular pressures and intercellular stresses are inferred.

(B) Similarly, the dynamical inference uses a time series of images to add dynamical information to the system of equations used in the static case, by assuming an overdamped regime. A time mesh is generated from the succession of microscopy images, and pivot vertices are tracked through time. These are the vertices at which three or more edges meet. Then, the velocity of these vertices from frame to frame is used to modify the system of equations, allowing non-static tissues to be analyzed by stress inference. Also see [Figures S1–S4](#).

applicability to dynamic tissues is limited. Also, although recent techniques using time series data offer improvements,¹² a computational tool capable of dynamic stress inference has been lacking.

To address this gap, here we introduce ForSys, an open-source Python-based inference algorithm specifically developed to tackle the complexities of dynamical stress inference from time series experiments. By inputting microscopy images segmented with tools such as EPySeg¹³ or Cellpose,¹⁴ ForSys performs stress inference in either static or dynamic mode via Python scripting, a Command Line Interface, or a Fiji-based Graphical User Interface. ForSys utilizes the geometry and local velocity of cell junctions to extract the spatiotemporal stress distribution *in vivo*, providing accurate estimations of a tissue's mechanical state.

RESULTS

ForSys: A python-based open-source software to infer mechanical stress in tissues

ForSys enables the inference of intercellular mechanical stress and intracellular pressure of tissues. It takes the two-dimensional (2D) segmentation of an image, which delineates cell outlines, as its input. The input could be a skeletonized representation of the tissue (such as one generated with EPySeg¹³) or a labeled mask (for instance, generated with Cellpose¹⁴). ForSys then conceptualizes the entire tissue as a polygonal structure. In this structure, each polygon represents a cell, with edges connecting vertices (see “the conceptual model behind ForSys” in the [STAR Methods](#) section).

ForSys operates in two distinct modes contingent upon the input ([Figure 1](#)). When supplied with a singular segmentation

of a static image, the software engages its static mode ([Figure 1A](#)) (see “static stress inference” in the [STAR Methods](#) section). In this mode, a stress inference is applied to a single image. Conversely, if the input comprises the segmentation of a time series dataset, ForSys presents the option to function in its dynamic mode (see “the dynamic inference case” in the [STAR Methods](#) section) ([Figure 1B](#)). This mode involves the extraction of temporal trajectories for vertices from the microscopy time series, thereby incorporating corresponding vertex velocities to refine stress inference.

To improve the usability of our method, we implemented three different ways to use the software after installing the package. First, users can access the software through Python scripting by importing the ForSys package ([Figure S1](#)). Second, a Command Line Interface is available for those comfortable working in a shell environment such as Bash or PowerShell (for Linux or Windows users, respectively), eliminating the need to write Python code ([Figure S2A](#)). Finally, we developed a Graphical User Interface that integrates with Fiji ([Figure S2B](#)), providing an accessible option for users who prefer not to work with code or command-line environments. Detailed installation and usage instructions are provided in the [STAR Methods](#) section, “interacting with the ForSys software”, as well as in the software documentation.

ForSys infers *in silico* stresses accurately in static equilibrium

To assess ForSys's performance against existing tools, we utilized as a ground truth simulations generated by a vertex model implemented in Surface Evolver via seapipy (see “seapipy: generation of *in silico* tissues to validate ForSys” in the [STAR Methods](#) section and [Figure S3](#)), analyzed it using our software,

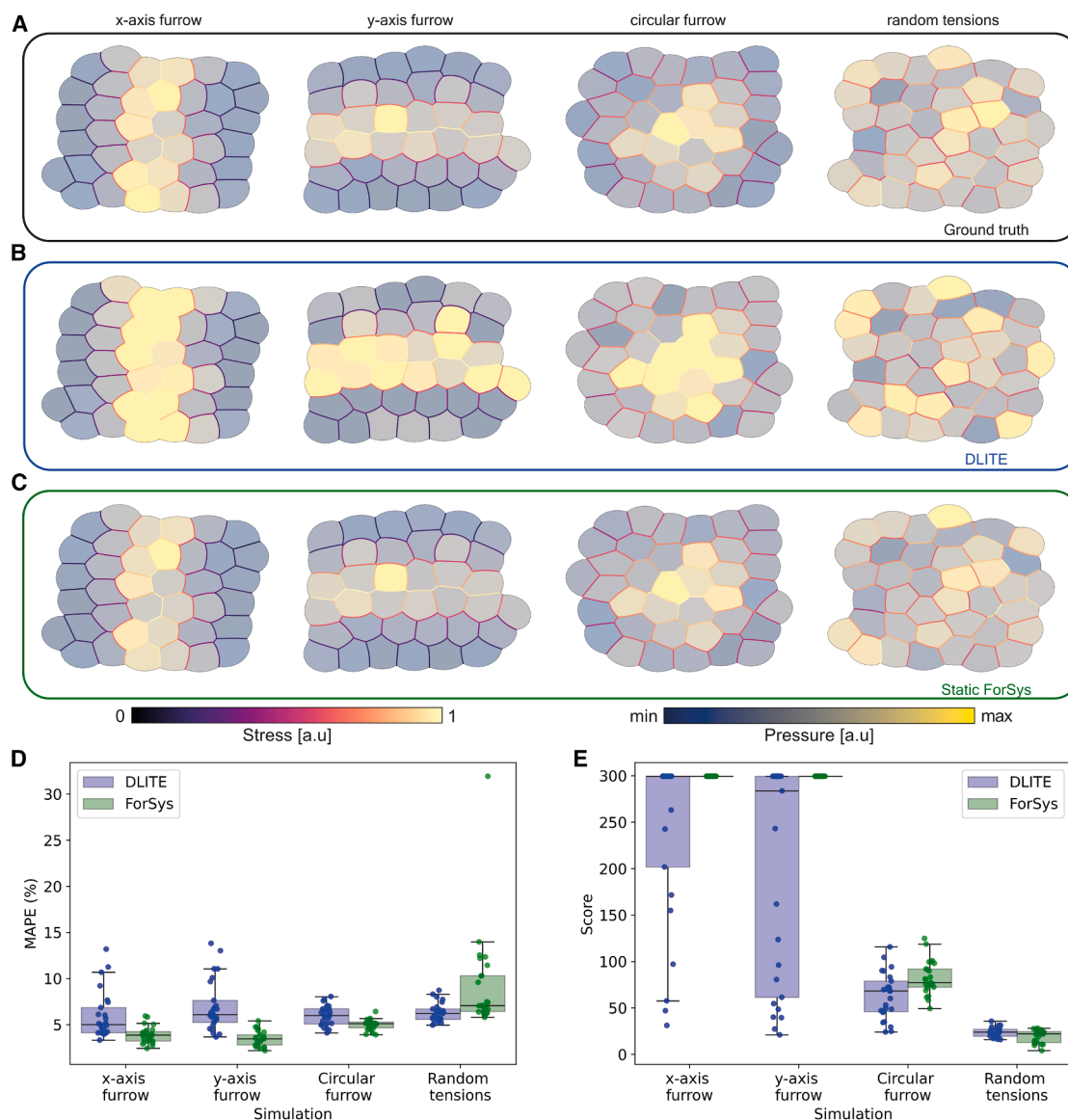


Figure 2. In silico validation of ForSys for tissues in static equilibrium

Four different conditions were generated with seapipy to benchmark ForSys under the static equilibrium condition. Each column shows a representative replicate per condition at the final frame ($t = 24$). The ground truth (A) can be compared to the values for the DLITE predictions (B) and the static ForSys (C). The three rows shown correspond to the final frame of the simulation. The color bar above the last two panels shows the order of the colormap for both the stresses and the pressures. Pressures in the cells are represented with transparency for improved visualization. The mean absolute percentage error (MAPE) (D) and the saturated score function (E) for all simulations ($N = 25$) are represented in two boxplots, DLITE and static inference with ForSys, paired by condition. (see [STAR Methods “evaluating goodness of fit”](#)) Dots show the result for individual repetitions. In (D) and (E), each boxplot shows the median value as a horizontal bar and first to third interquartile ranges as boxes; the upper whisker is either $1.5 \times$ the interquartile range or the maximum value (whichever is the smallest) and the lower whisker is either $1.5 \times$ the interquartile range or the minimum value (whichever is the biggest). The pressure colormaps for the ground truth shown in panel (A), expressed as (minimum, maximum), are from left to right (0.0551, 0.3622), (0.0351, 0.3887), (0.0332, 0.0878), and (0.0289, 0.0924). For panel (B), they are (−1.17, 3.06), (−1.17, 3.01), (0.21, 1.78), and (0.38, 1.44). For panel (C) they are (−0.36, 0.71), (−0.45, 1), (−0.20, 0.38), and (−0.31, 0.31). Also see [Figures S5](#) and [S6](#).

and compared the results with outputs from previously published methods, focusing specifically on CellFIT¹⁰ and DLITE.¹² Given that both tools yield similar results ([Figures S5A](#) and [S5B](#)), we opted for DLITE implementation due to its open-source nature, enabling a direct comparison with tissue stresses extracted from Surface Evolver outputs.

We selected the final time-point ($t = 24$) of simulations generated from four different conditions to compare the ground truth from the Surface Evolver output ([Figure 2A](#)), DLITE’s estimation ([Figure 2B](#)), and ForSys in its static modality ([Figure 2C](#)). In all cases, the predicted intercellular stresses and intracellular pressures closely matched the ground truth. Moreover, both stress

inference methods exhibited a high degree of accuracy and precision, as reflected by a low mean absolute percentage error (MAPE) ($<10\%$) (Figure 2D) and a high saturated score function (~ 30) (Figures 2E and S5C). Importantly, ForSys showed a significantly lower MAPE ($p = 1e-05$; $p = 1e-8$; $p = 0.01$, for the x-furrow, y-furrow, and circular furrow, respectively), higher saturated score ($p < 0.001$; $p < 1e-4$; $p < 0.01$ for the x-furrow, y-furrow, and circular furrow, respectively), and smaller inter-quartile range than DLITE, for all cases except the random tensions (See the “statistical estimators” section in STAR Methods for details).

These results indicate that ForSys’s static modality yields higher accuracy and precision estimations than DLITE while effectively capturing the *in silico*-generated ground truth spatial distributions in static equilibrium. Consequently, only ForSys in its static modality will be used hereafter for comparison with a static solution.

ForSys stress inference in dynamical tissues outperforms static methods *in silico*

With the aim of inferring stress in dynamic tissue, we assumed that the tissue goes through a succession of quasistatic states in an overdamped regime,¹⁵ consistent with a viscoelastic response of the cell junctions to the deformations created by the forces acting on them.^{16,17} Consequently, we incorporated a viscous term proportional to the velocity of the corresponding vertex in each junction’s equation. Importantly, these velocities are not unknown: ForSys estimates them using the spatial coordinates of the vertices tracked over time. In ForSys, we call this modality of stress inference “Dynamic”.

Dynamic inference depends on a dimensionless number, which we call the Scale Parameter ρ (see a detailed description in the “the dynamic inference case” section of STAR Methods). Thus, we fitted ρ and found its optimal value for each of the examples. Our results indicate that the best dynamic results are obtained with a scale parameter of about 0.1 (see Figure S9 and STAR Methods section “determination of the scale parameter *in silico*”).

Under our prescribed conditions (Figure 3A), ForSys in its static modality (Figure 3B) is outperformed by dynamic inference (Figure 3C), accurately reproducing stress and pressure distributions akin to the ground truth. Furthermore, our results indicate that dynamic modality improves static modality accuracy and precision, as indicated by lower MAPE ($p < 1e-09$; $p < 1e-9$; $p < 1e-7$, for the x-furrow, y-furrow, and circular furrow, respectively) (Figure S7A) and a higher saturated score function ($p < 1e-08$; $p < 1e-9$; $p < 1e-9$; $p = 0.03$, for the x-furrow, y-furrow, circular furrow, and random, respectively) (Figure 3D) (See the “statistical estimators” section in STAR Methods for details).

Interestingly, the accuracy and precision of stress inference in each ForSys modality are damped by the increases of TJ local movements, here reflected in the norm of the velocity vector ($|v|_2$) (Figures 3E and S7D). Notably, the dynamic modality outperforms the static one for all TJ velocities, as observed by the time evolution of MAPE (Figure S7D). This can be evidenced through the logarithmic ratio between dynamic and static scores (Figure 3E), where values greater than zero mean that the dynamic modality outperforms the static one. The outperformance

of the dynamical modality is clearer for higher TJ velocities (Figures 3E, S7D, and S8). Thus, ForSys, in its dynamic modality, can retain a better approximation due to its use of the vertices’ velocity, i.e., future positions, to estimate the stress.

In this section, we have shown through *in silico* validation that the dynamic modality of ForSys outperforms other methods in accurately inferring stresses in remodeling tissues.

ForSys validation *in vivo* using the mucociliary epithelium of *Xenopus* embryos

To test ForSys in a biological setting, we used published data from the mucociliary epithelium in *Xenopus* embryos (Figure 4A).¹⁸ We quantified myosin II intensity using a non-muscle myosin II A-specific intrabody (SF9-3xGFP, for simplicity referred to as myosin II), which has been previously used as a proxy for active myosin II.^{19,20} We segmented the microscopy images using EPySeg¹³ (see STAR Methods “SF9 myosin II sensor intensity measurements” for details) and compared myosin II measurements with the stress values inferred by ForSys.

As before, we qualitatively compared the derived stress distribution maps with the ground truths, here given by the normalized myosin II sensor intensity (Figure 4B). We observed a good qualitative agreement between inferred stress and myosin intensity, with regions of higher myosin fluorescence corresponding to higher inferred stress, most noticeable in Embryo 3 and Embryo 5 of Figure 4B. In contrast, in Embryo 4 of the same panel, ForSys can reproduce a more homogenous distribution along the tissue. On a quantitative level, we found that ForSys predictions are moderately correlated with the myosin measurements for each embryo ($R = 0.56 \pm 0.11$; mean $\pm$ std) (Figure 4C). In addition, ForSys stress predictions have a MAPE value of ($21 \pm 5\%$) (mean $\pm$ std). To further test the similarities between the myosin intensity and ForSys predictions, we pooled together all the membranes from the embryos shown in Figures 4B and 4C and compared their distributions. We found that myosin intensity and stress distributions are not significantly different (Figure 4D; $p = 0.76$, Mann-Whitney U test; $p = 0.97$, Wilcoxon signed-rank test; $N = 154$).

Altogether, ForSys in its static modality, recapitulates the stresses present in the mucociliary epithelium of the *Xenopus* embryo, as measured by the fluorescence of the myosin II sensor.

Dynamic stress inference of collective cell behavior in zebrafish

We sought to explore ForSys inferences in an *in vivo* model that mixes TJs with low and high motility. To this end, we turned to two morphogenetic processes that occur during the development and homeostasis of the zebrafish lateral line, a mechanosensory organ formed by a collection of discrete organs called neuromasts.

We first applied ForSys to an *in vivo* model of collective cell morphogenetic behavior leading to the formation of epithelial rosettes in the lateral line primordium of developing zebrafish. The primordium is a collection of just over 100 cells that move collectively from head-to-tail of the fish embryo (Figure 5A). During migration, groups of approximately 25

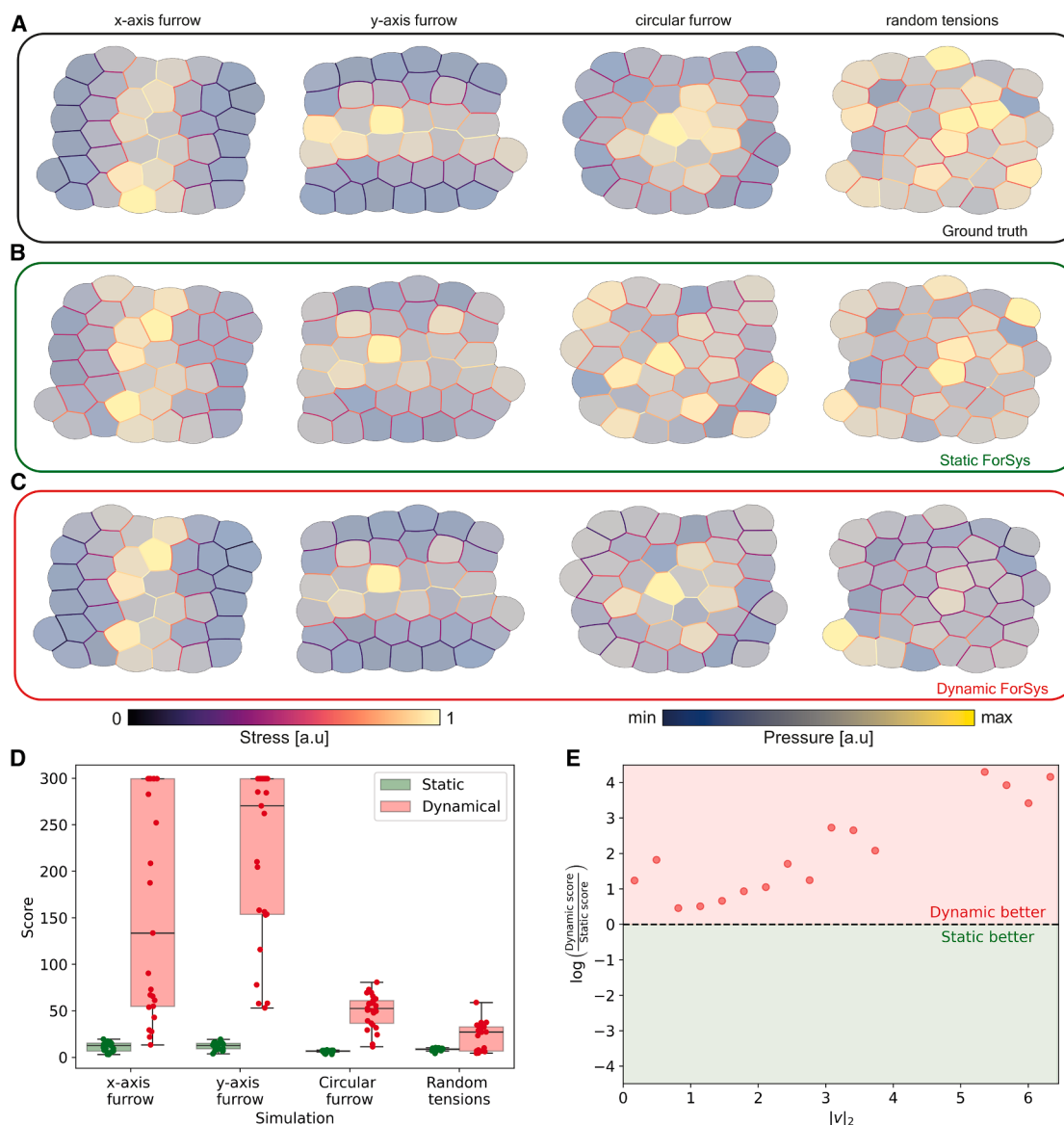


Figure 3. In silico validation of ForSys for tissues in dynamical equilibrium

(A–D) We generated four examples with seapipy to test dynamical equilibrium conditions. Each column shows a representative repetition per example. The first row (A) shows the ground truth values for the stress and the pressures, the static inference made by ForSys is in the second row (B), and the dynamical ForSys inference is in (C). We show each example at one time point after the system's tensions changed. The color bars below the last two panels show the colormap for both the stresses and pressures. The saturated score function values (D) for all simulations are represented in two boxplots, static and dynamical inference, paired by condition. Dots show the result for individual repetitions and each boxplot shows the median value as a horizontal bar and first to third interquartile ranges as boxes; the upper whisker is either $1.5 \times$ the interquartile range or the maximum value (whichever is the smallest) and the lower whisker is either $1.5 \times$ the interquartile range or the minimum value (whichever is the biggest).

(E) Dynamic to static score function ratio ($r = \log \left(\frac{\text{dynamic}}{\text{static}} \right)$) as a function of the $|v|_2$ bin. A ratio bigger than zero shows that the dynamic solutions performed better (Red zone), and a negative value (Green Zone) favors the static solution. The black dashed line at $y = 0$ separates both zones. All velocity bins favor the dynamic solution. Pressure colormaps have the following values, (minimum, maximum), for each of the panels, from left to right: For panel (A) they are (0.0609, 0.4209), (0.0488, 0.3777), (0.0285, 0.0822), and (0.0384, 0.0946); for panel (B) they are (−0.21, 0.49), (−0.33, 1), (−0.20, 0.0.20), and (−0.24, 0.25); for panel (C) they are (−0.21, 0.49), (−0.26, 0.8), (−0.19, 0.31), and (−0.23, 0.42). Also see [Figures S7–S9](#).

trailing cells periodically detach from the primordium, sequentially giving rise to individual neuromasts that are then deposited at a semi-regular pace.²¹ Although the lateral line primordium has been extensively characterized genetically,^{22,23} the

evolution of mechanical forces during migration and rosetto-genesis remains underexplored.

Therefore, we decided to use ForSys in its dynamic modality to analyze time lapse data of migrating primordia, whose cells'

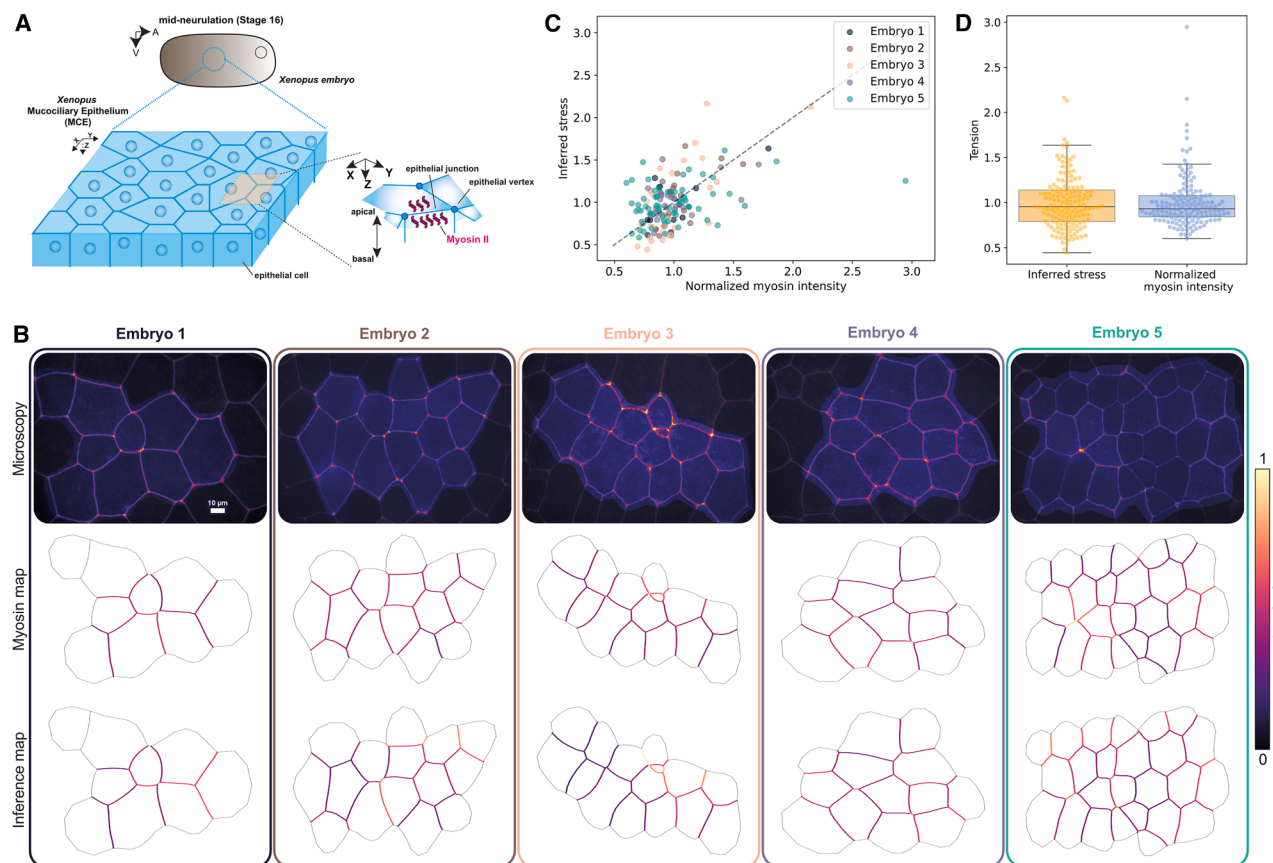


Figure 4. Comparison of ForSys-derived stress with myosin II measurements in the *Xenopus* embryo mucociliary epithelium

(A) Scheme of the *Xenopus* embryo and position of the mucociliary epithelium.

(B) Five examples of inference in *Xenopus* embryos. The 10 μm scale bar present in the Embryo 1 microscopy also applies to the other 4 images. The microscopy image is shown alongside the myosin intensity map and the ForSys inference result. The color code in the maps represents the myosin sensor intensity and the stress prediction. The scale was saturated at tension values of one, using the “absolute” normalization provided in the ForSys package. The highlighted region in the microscopies shows the area that was analyzed.

(C) Relationship between myosin sensor intensity and stress inferred for the five examples. Each scatter point shows the value for a particular membrane in that example. The dashed black line represents the $y = x$ line. Each color coincides with the rounded rectangle around the embryo and its font color in panel (B). The average Pearson correlation coefficient is $R = 0.56 \pm 0.11$; (mean $\pm$ std).

(D) Quantification of stresses and myosin sensor intensity for the five examples. The Inferred stresses and myosin intensities are not significantly different from each other ($p = 0.76$, Mann-Whitney U test; $p = 0.97$, Wilcoxon signed-rank test; $N = 154$). The boxplots show the median value as a horizontal bar and first to third interquartile ranges as boxes; the upper whisker is either $1.5 \times$ the interquartile range or the maximum value (whichever is the smallest), and the lower whisker is either $1.5 \times$ the interquartile range or the minimum value (whichever is the biggest).

plasma membranes were fluorescently labeled with EGFP. Migrating primordia were followed for 30 min with a temporal resolution of 2 min (Video S1) (Figures 5B and 5B') (see a detailed description in the “zebrafish primordium migration experiments” section of STAR Methods).

ForSys predicted the mitotic division of primordial cells by revealing high stress in the pre-dividing cellular membrane relative to the membrane of the non-dividing surrounding cells (Figure 5C). The stresses remain partially conserved after division, mainly in the cell membrane separating the resulting cell siblings (Figure 5C').

We then applied ForSys to predict stress tissue-wide. Apical constrictions of epithelializing cells are mechanistically associated with the formation of the rosettes that preempt neuromast

morphogenesis.²⁴ The apical constriction is readily detectable by morphology when looking at the apical plane of the primordium (Figures 5D and 5D').²⁵ The relationship between apical constrictions and forces in more basal planes of the cells and how they relate to rosetogenesis remains undefined.

To begin to address this possible relationship, we used time series data and aggregated the position of the cells along the anteroposterior axis of the primordium by kernel density estimation. We weighed each cell using the intracellular pressure inferred from ForSys, which results in a smoothed curve estimating intracellular pressure along the migration axis (Figure 5E). This analysis showed that the anteroposterior positions of the rosettes, manually annotated by looking at apical constriction (Asterisks in Figures 5D' and 5E), correlate with

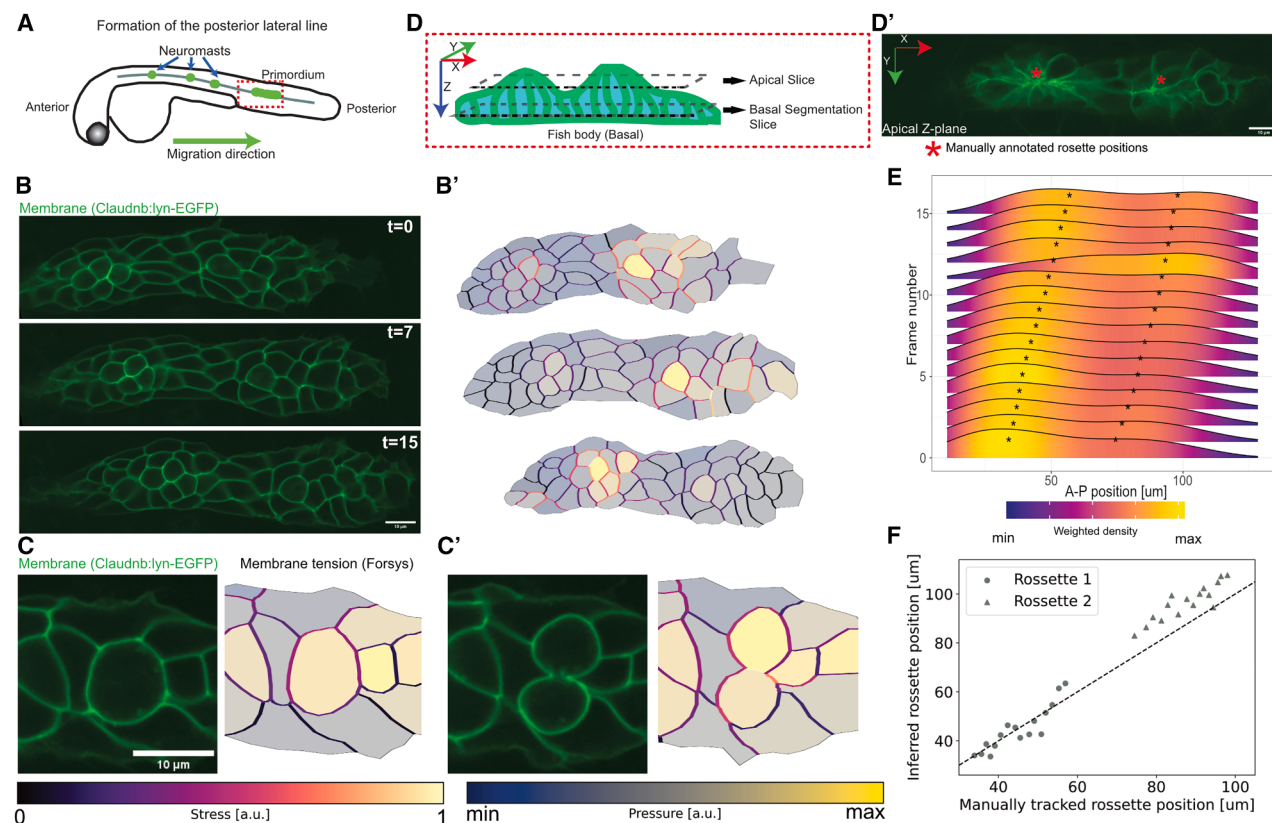


Figure 5. ForSys inference of a moving epithelium in the zebrafish lateral line at 2 dpf

(A) Schematic of the biological model. The neuromasts of the posterior lateral line are formed by detaching from a primordium that migrates from the anterior to the posterior of the fish.

(B) Frames 0, 7, and 15 of the primordium migration in which cell membranes are fluorescently marked with Claudnb:lyn-EGFP. The membrane signal is used for segmentation, which ForSys can use to predict cell membrane tension and intracellular pressure (B').

(C and C') Consecutive frames show cell division. The membrane tension in the cell just about to divide is considerably higher than the surrounding membranes. After division, the dividing membrane retains a high tension.

(D and D') Schematic of the primordium orientation and the position of the optical planes. Constriction of the cell membranes in rosettes is evident in the apical plane. The asterisks show the anteroposterior location of rosettes. The cell segmentation was done on a Z-plane at a more basal plane (E) Ridge line plots of Cell densities along the anteroposterior axis throughout 16 frames for a representative primordium. Time goes from bottom to top. The direction of primordium migration is to the right. The asterisks show the positions of the manually annotated rosettes.

(F) Anteroposterior position of the manually tracked rosette against the inferred position by taking the local maxima of the density of pressure values from (E). The diagonal line marks $y = x$ as a reference for comparing predicted and manually annotated values. Scale bars represent 10 μm . Tension and pressure scales are shown at the bottom of the figure. Pressure scale values (minimum value and maximum value) for panel (B') are (−0.63, 1.6), (−0.75, 1.24), (−0.76, 1.37) for $t = 0, 7$, and 15, respectively. Minimum and maximum values for the pressure scale in panel (C) is (−0.74, 1.08) and (−0.85, 1.15) for panel (C').

the predicted zones of high intracellular pressure in the basal plane inferred by ForSys. The closeness between the predicted pressure maxima and the manually annotated rosette formation indicates a high correlation between these two quantities during primordium migration ($R = 0.99$, $p < 1e-51$, $N = 61$; for rosettes 1 and 2 combined) (Figure 5F).

Encouraged by our previous results, we next analyzed mature neuromasts. The center of this organ is occupied by mechanosensory hair cells, which are surrounded by non-sensory supporting cells (Figure 6A).²⁶ We used a plasma membrane marker to define cells, which were segmented using ilastik²⁷ and EPySeg¹³ (Figure 6B) (see a detailed description in the “zebrafish neuromasts videomicroscopy and analysis” section of STAR Methods). Then, we used the dynamical mo-

dality of ForSys to estimate the stress at each membrane (Figure 6B') and found that membranes belonging to hair cells have higher stress on average. Homotypic interfaces between hair cells have the highest stress ($p < 1e-7$ vs. hair cell-supporting cell interfaces). On the other hand, homotypic contacts between supporting cells have the lowest stress ($p < 0.006$ vs. hair cell-supporting cell interfaces) (Figure 6C).

We then focused on a still-puzzling process called planar cell inversion (PCI).^{28,29} PCI occurs when supporting cells give rise to hair-cell progenitors, which divide once to generate a pair of hair cells. Approximately half of the resulting nascent hair-cell pairs undergo a 180° rotation around their geometric center^{28,29} (Figures 6D and 6E). The mechanical forces occurring during

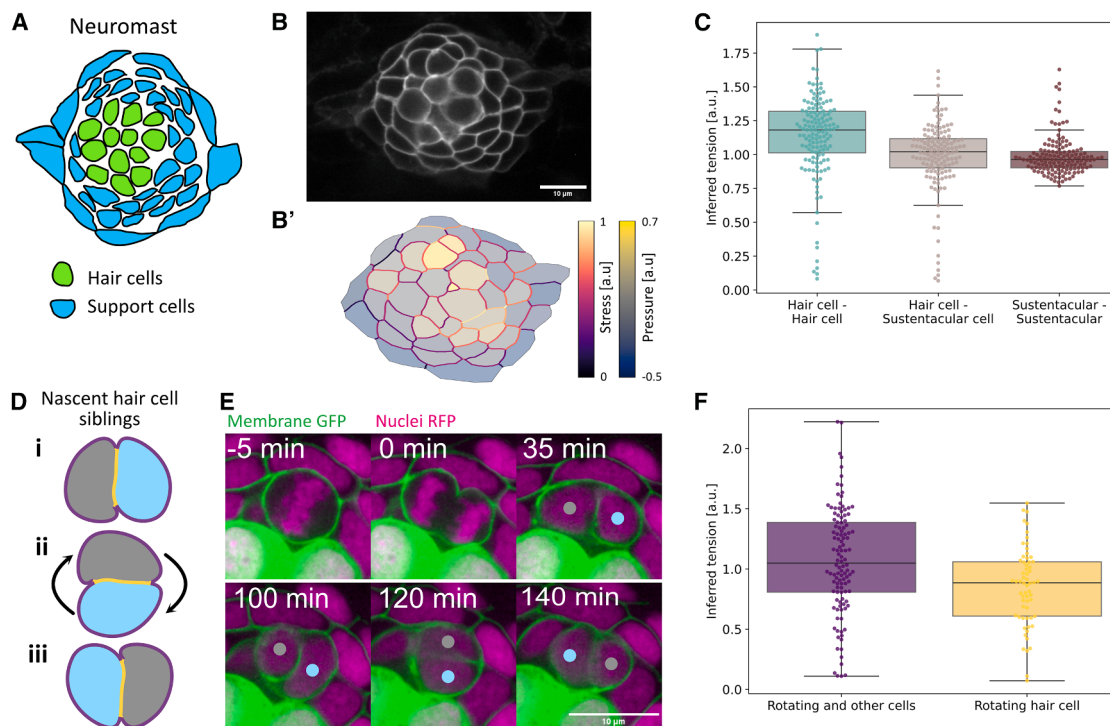


Figure 6. *In vivo* ForSys inference in an epithelium with rotating cells

(A) Schematic of cell composition in a zebrafish lateral line neuromast. Sensory hair cells are located in the center and are surrounded by support cells. (B) Image of a neuromast whose cells can be tracked by membrane-tethered EGFP. (B') ForSys tension and pressure inference after membrane segmentation. (C) The tension inferred for membranes is classified by the type of cell-cell contact. The homotypic contacts between hair cells show the highest predicted tension, while the homotypic contacts between support cells show the lowest on average. Each data point is the mean of the predicted tension values for each membrane type in one frame. The frames come from $N = 7$ time-lapse experiments. (D) Schematic of the planar cell inversions occurring in 50% of the nascent hair cell pairs: sibling hair cells perform a 180° rotation to exchange positions along the anterior-posterior axis. The coloring of membranes corresponds to the colors of the respective boxplot in panel (F). (E) Time-lapse frames show the *in vivo* rotation process: around 100 min after mitosis, the nascent hair cells exchange anteroposterior positions by rotating in the epithelial plane. The sibling cells remain attached during the rotation, while the surrounding cells do not actively participate in the movement. The TgBAC(cxcr4b: H2B-RFP) transgenic was used to label nuclei. (F) Homotypic tensions between the young rotating hair cells are significantly lower than their contacts with the surrounding cells. In (C) and (F), boxplots show the median value as a horizontal bar and first to third interquartile ranges as boxes; the upper whisker is either $1.5 \times$ the interquartile range or the maximum value (whichever is the smallest) and the lower whisker is either $1.5 \times$ the interquartile range or the minimum value (whichever is the biggest). Scale bars represent 10 μm .

cell-pair inversions are not known, but the physical modeling of the cell doublet suggests that differential surface tension plays a central role.^{30,31} Therefore, we focused our analysis on the homotypic junctions between the sibling hair cells and compared them to those with the surrounding supporting or hair cells. We found that the stress in the membranes juxtaposing the rotating hair-cell pair is significantly smaller than that between hair cells and the surrounding cells ($p < 0.0005$) (Figure 6F). Because tension and adhesion are generally inversely related, PCI could be characterized by a strong adhesion within the rotating cell pair and weaker adhesion with the surrounding cells. This result suggests a cell-type and cell-state-specific adhesion pattern that influences contact remodeling necessary for coordinated cell-pair rotations.

Taken together, these results show that ForSys's dynamical implementation predicts high stresses before cell division in a migrating tissue. They also revealed that rosette formation could be prefigured by mechanical rosetogenesis changes in the cells,

which allows the inference of apical constrictions during rosetogenesis using information from basal planes.

Scale parameter selection *in vivo*

The results presented in the previous section were obtained by setting the scale parameter to 0.1, the optimal value identified through *in silico* validation. To assess the validity of this default value experimentally, we performed *in vivo* laser ablation of cell membranes in zebrafish neuromasts and measured the time evolution of the distance between the pivot vertices delimiting the ablated membranes (Figures 7A–7C).

We modeled the system using a minimal physical description. For simplicity, we aligned the two pivot vertices along the x axis, with their positions defined as $x_1(t)$ and $x_2(t)$ (Figure 7C, inset). Prior to ablation, the membrane is assumed to be in mechanical equilibrium, consistent with the central assumption of ForSys. Thus, the magnitude of the net external force at each pivot vertex is balanced by the membrane stress that could be inferred by

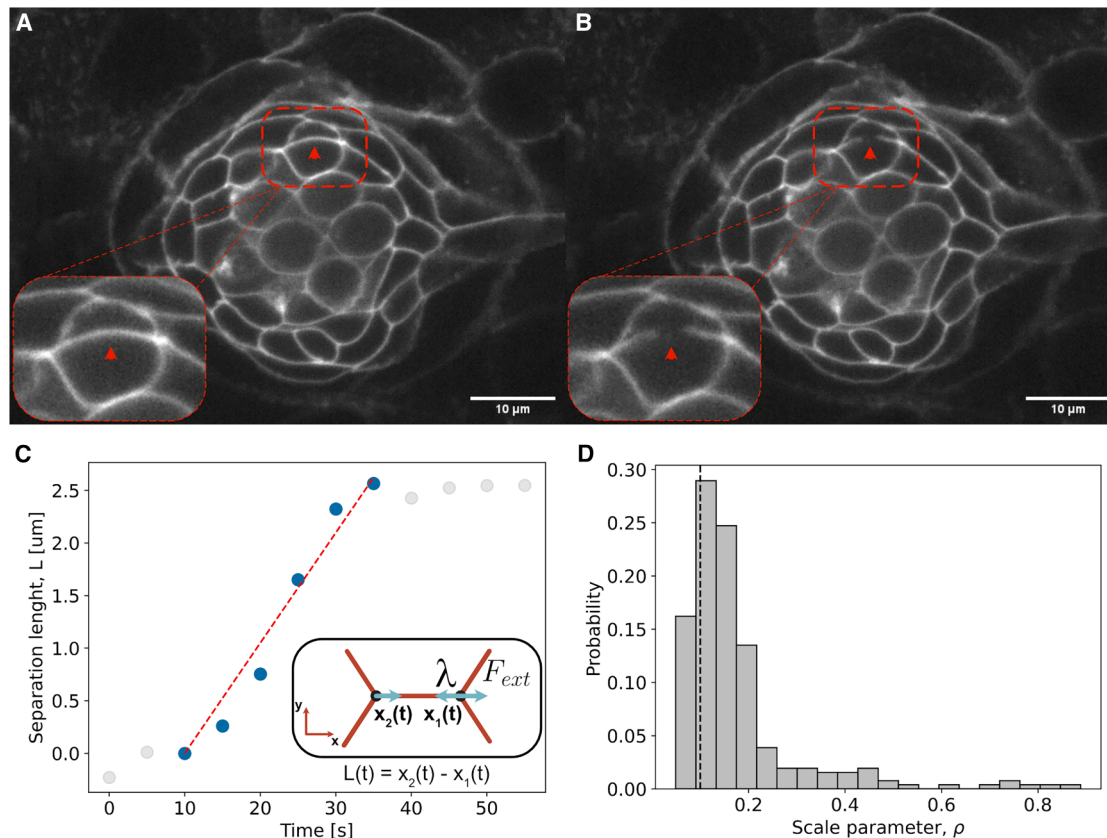


Figure 7. Scale parameter determination through laser ablation

The scale parameter was determined through laser ablation experiments, as described in the STAR Methods section “zebrafish laser ablation experiments.” Microscopy images are taken every second, including just before (A) and just after (B) ablation. The red arrow indicates the membrane where the ablation takes place, and the dashed red line is a guide to the eye. The inset in the lower right corner of both (A) and (B) panels zooms in on the ablation zone. Scale bars represent 10 μm .

(C) Example trajectory of the membrane after ablation. The y axis shows the elongation relative to the length in the frame just before ablation occurs. Blue scatter dots are used for the fitting, gray points are not. The inset in the lower right corner shows a sketch of the ablated membrane.

(D) Using the linear fits from thirty-seven ablation experiments, we determine the scale parameter by multiplying each experimental slope with the characteristic velocity (v) of each of the seven neuromasts that comprise Figure 6. In panel (D), the scale parameter resulting from the fits is represented as a histogram. The vertical dashed line represents the value used in our inferences ($\rho = 0.1$).

ForSys ($\|F_{ext}\| = \lambda$; see Equation 2 in STAR Methods section “static stress inference”). Immediately after ablation, each vertex recoils with velocity v , and a viscous damping force opposes this motion, balancing the membrane stress ($\lambda = \eta v$; see Equation 6 in the STAR Methods section “static stress inference”). As a result, the distance between the two vertices evolves over time as:

$$L(t) = L_0 + 2(\lambda / \eta)t \quad (\text{Equation 1})$$

where $L(t) = x_2(t) - x_1(t)$ is the separation length between the vertices at time t , and L_0 is their initial separation length.

By fitting this expression to the recoil dynamics measured in zebrafish neuromasts at short times, we extracted the ratio λ/η (Figure 7C). Using the characteristic velocity of neuromast vertices ($\bar{v}$, as defined in Equation 7 in the STAR Methods section “static stress inference” and determined from the inferences shown in Figure 6) and applying Equation 8 in the STAR Methods section

“static stress inference”, we estimated the corresponding value of the scale parameter. The experimentally determined value of the scale parameter was 0.17 ± 0.13 (mean $\pm$ std), which is consistent with the value used in our inferences (Figure 7D).

DISCUSSION

Here, we introduce ForSys, a software that statically and dynamically infers stresses without disrupting biological tissues. Traditional inference methods rely on geometrical information to calculate the relationship among the stresses acting on cell membranes in a static image. However, these methods generally lack the dynamical component present in time-series microscopy. ForSys extends the applicability of inference techniques by enabling dynamic stress inference in cell membranes when tissues are in motion.

We validated our software in its static and dynamic modalities with different *in silico* spatial patterns of tissue stresses using a

cell-based computational model implemented in Surface Evolver,³² which we integrated into a Python package called *seapipy*.³³ Our results show that ForSys can recover the ground truth in its static and dynamic modalities. Significantly, the dynamic modality improves the accuracy of the static modality in the tested conditions. Unlike static inference, which involves a single intrinsic scale embedded in the inferred stresses, dynamic inference introduces a viscous term proportional to the velocities of the nodes, thereby introducing a second physical scale. The interplay between these two scales is captured by the scale parameter, which modulates the relative contributions of viscous forces and other mechanical stresses in the inference process. In this sense, the scale parameter plays a role analogous to the Weissenberg number or the Capillary number.^{34–36} The optimal value found for this number indicates that elastic forces are an order of magnitude larger than viscous forces. Strikingly, dynamic inference outperforms static inference *in silico* even when elastic forces dominate over the viscous forces, pointing to a wide applicability of the dynamic modality.

We then tested ForSys in the *Xenopus* embryonic mucociliary epithelium. We found a positive correlation between the inferred stress and cortical stress that was indirectly measured using variations in the intensity of myosin II. As the development of the embryonic mucociliary epithelium progresses over several hours, continuous, direct probing of the mechanical forces is extremely laborious, interferes with tissue development, and is hardly compatible with single-cell resolution measurements. Therefore, using ForSys for the non-invasive mapping of mechanical forces at the scale of an entire tissue across time could pave the way for a more comprehensive understanding of the mechanical forces that drive tissue development.

We further demonstrated the power of ForSys by studying two aspects of organ development and homeostasis using the neuromasts of the lateral line in zebrafish embryos. Specifically, we addressed two processes that involve a complex collective cell behavior. First, we applied dynamical ForSys to the migrating lateral-line primordium. Although this process has been extensively dissected genetically, it is still unknown what forces play a role during migration and neuromast deposition.^{37,38} Therefore, this process of collective cell migration will benefit from an accessible and non-invasive method to estimate forces in a dynamical tissue. Two characteristics of this migratory primordium make it well-suited for applying ForSys: the tissue as a whole is migrating through the lateral line, and its membranes have a curved shape. We showed that ForSys can detect cell division and rosette formation. ForSys will be useful for testing various hypotheses about tissue mechanics in other dynamic cell systems, for instance, during tissue repair and organ regeneration.

We also applied ForSys to address the still mysterious process of planar cell inversion, during which sibling cells rotate around their centroid after the mitotic division of their progenitor.²⁸ We discovered that homotypic contacts between rotating cells have the most stress, whereas the contacts between the rotating pair have lower stress than the contacts of each hair cell with its neighbors. This strongly suggests that adhesion dynamics during rotation are associated with strong homotypic interactions between the sibling cells and weak heterotypic interactions

with the surrounding cells, enabling contact exchange during the inversion.²⁸

The development and improvement of stress inference tools presents a promising avenue to capture the spatiotemporal complexity of mechanical forces during tissue morphogenesis, which circumvents challenges related to sample fixation and limited temporal resolution.³⁹ An interesting possibility is to use fluorescence-based tension and strain mapping techniques, such as those employing molecular tension sensors^{40,41} or fluorescence intensity correlates^{42,43} that enable the direct visualization of mechanical forces in tissues in combination with ForSys. Future studies combining these complementary methods could provide a more comprehensive understanding of tissue mechanics *in vivo*.

ForSys provides a versatile and noninvasive tool for studying spatiotemporal patterns of mechanical stresses during tissue morphogenesis *in vivo*. This software makes stress predictions that can guide researchers in conducting further experiments, which can significantly contribute to understanding the mechanisms involved in development and regeneration. ForSys was built as open-source software in Python, thus allowing the community to participate in its development and maintenance. With its usability in mind, aside from the possibility of writing Python scripts to interact with ForSys, it has been further equipped with a Command Line Interface and a Graphical User Interface embedded in Fiji⁴⁴ to facilitate its adoption. In our eyes, an interesting future perspective will be to extend the software to tissues in non-equilibrium conditions and adapt the method to operate within a 3D geometry to generate 4D mechanical stress inference.

Limitations of the study

The applicability of the method described in this article, as well as the ForSys software, relies on two groups of assumptions: methodological and theoretical.

The methodological assumptions primarily concern limitations related to the quality of the input data supplied to the program. These include, but are not limited to, poor segmentation leading to the inaccurate reconstruction of cellular membranes, or the inability to track all agents contributing to the system's movement, such as interactions with structures located outside the region of interest or out of the imaging plane. In general, the higher the quality of the input data, the more accurate the predictions produced by ForSys.

The theoretical assumptions involve the validity of the underlying hypotheses in the tissue under study. For instance, if the acceleration of tricellular junctions is sufficiently large, mechanical equilibrium may not hold, and the tissue would no longer behave as an overdamped system. However, in most biological tissues, inertial effects containing acceleration terms are negligible compared to viscous damping, supporting the overdamped regime assumption.¹⁵ Another theoretical limitation of our model is the assumption that membrane stresses remain constant within the time window of each image frame. If stresses fluctuate on timescales faster than the image acquisition frequency, ForSys will not capture these rapid changes. In such cases, the inferred stresses should be interpreted as averaged or representative values over the time interval.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to Osvaldo Chara (osvaldo.chara@nottingham.ac.uk).

Materials availability

This study did not generate new materials.

Data and code availability

The seapipy³³ codebase is available on GitHub at <https://github.com/borgesaugusto/seapipy>. ForSys⁴⁵ is available on GitHub <https://github.com/borgesaugusto/forsys>. All data reported in this article will be shared by the lead contact upon request.

ACKNOWLEDGMENTS

The authors thank Luis Morelli, Fabian Rost, Nicolas Aldecoa, Alice Descoeurdes, and all the members of the Chara laboratory for valuable comments and suggestions. A.B. was funded by the BMBF 01GQ1904 grant. A.S.C. was funded by a Doctoral fellowship from CONICET, Argentina, and by a Biotechnology and Biological Sciences Research Council (BBSRC) grant [grant number BB/X014908/1]. J.S. acknowledges the support of the Novo Nordisk Foundation (grant number NNF22OC0076414, NNF19OC0056962) and LEO Foundation (grant number LF-OC-19-000219), the European Research Council Consolidator Grant (ERC CoG 101125803 MechanoFate). The Novo Nordisk Foundation Center for Stem Cell Medicine (reNEW) is supported by a Novo Nordisk Foundation grant number NNF21CC0073729. O.C. was funded by CONICET, Fondo para la Investigación Científica y Tecnológica (PICT-2019-03828) and the BBSRC grant BB/X014908/1 to O.C.

AUTHOR CONTRIBUTIONS

A.B. developed the ForSys code, simulated the *in silico* results, analyzed all the inference results, and wrote the article. A.S.C. contributed to the development of the ForSys code and analyzed all the inference results. G.V. generated the *Xenopus* images and edited the article. J.M.-R. acquired the zebrafish microscopy images and analyzed the corresponding results. J.S. supervised the *Xenopus* project and edited the article. H.L.S. supervised the Zebrafish project and edited the article. O.C. conceived the project, contributed to the ForSys code, analyzed all the inference results, wrote the article, and supervised A.B. and A.S.C.

DECLARATION OF INTERESTS

The authors declare no competing interests.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- **KEY RESOURCES TABLE**
- **EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS**
 - SF9 myosin II sensor intensity measurements
 - Zebrafish primordium migration experiments
 - Zebrafish neuromasts videomicroscopy and analysis
 - Zebrafish laser ablation experiments
- **METHOD DETAILS**
 - Applying ForSys to *in vivo* zebrafish microscopy
 - The conceptual model behind ForSys
 - seAPipy: Generation of *in silico* tissues to validate ForSys
 - Static stress inference
 - The dynamic inference case
 - Solving the system of equations
 - Determination of the scale parameter *in silico*
 - Interacting with the ForSys software

QUANTIFICATION AND STATISTICAL ANALYSIS

- Evaluating goodness of fit
- Statistical estimators
- Comparing ForSys with other computational methods

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1016/j.isci.2025.113685>.

Received: June 24, 2024

Revised: July 2, 2025

Accepted: September 29, 2025

Published: October 3, 2025

REFERENCES

1. Serwane, F., Mongera, A., Rowghanian, P., Kealhofer, D.A., Lucio, A.A., Hockenbery, Z.M., and Campàs, O. (2017). *In vivo* quantification of spatially varying mechanical properties in developing tissues. *Nat. Methods* 14, 181–186. <https://doi.org/10.1038/nmeth.4101>.
2. Mongera, A., Rowghanian, P., Gustafson, H.J., Shelton, E., Kealhofer, D. A., Carn, E.K., Serwane, F., Lucio, A.A., Giammona, J., and Campàs, O. (2018). A fluid-to-solid jamming transition underlies vertebrate body axis elongation. *Nature* 561, 401–405. <https://doi.org/10.1038/s41586-018-0479-2>.
3. Mongera, A., Pochitaloff, M., Gustafson, H.J., Stooke-Vaughan, G.A., Rowghanian, P., Kim, S., and Campàs, O. (2023). Mechanics of the cellular microenvironment as probed by cells *in vivo* during zebrafish presomitic mesoderm differentiation. *Nat. Mater.* 22, 135–143. <https://doi.org/10.1038/s41563-022-01433-9>.
4. Nishizawa, K., Lin, S.-Z., Chardès, C., Rupprecht, J.-F., and Lenne, P.-F. (2023). Two-point optical manipulation reveals mechanosensitive remodeling of cell-cell contacts *in vivo*. *Proc. Natl. Acad. Sci. USA* 120, e2212389120. <https://doi.org/10.1073/pnas.2212389120>.
5. Bonnet, I., Maroq, P., Bosveld, F., Fetter, L., Bellaïche, Y., and Graner, F. (2012). Mechanical state, material properties and continuous description of an epithelial tissue. *J. R. Soc. Interface* 9, 2614–2623. <https://doi.org/10.1098/rsif.2012.0263>.
6. Etoiray, R., Popović, M., Merkel, M., Nandi, A., Blasse, C., Aigouy, B., Brandl, H., Myers, G., Salbreux, G., Jülicher, F., and Eaton, S. (2015). Interplay of cell dynamics and epithelial tension during morphogenesis of the *Drosophila* pupal wing. *eLife* 4, e07090. <https://doi.org/10.7554/eLife.07090>.
7. Tetley, R.J., Staddon, M.F., Heller, D., Hoppe, A., Banerjee, S., and Mao, Y. (2019). Tissue fluidity promotes epithelial wound healing. *Nat. Phys.* 15, 1195–1203. <https://doi.org/10.1038/s41567-019-0618-1>.
8. 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. <https://doi.org/10.1016/j.devcel.2015.12.012>.
9. Bosveld, F., Wang, Z., and Bellaïche, Y. (2018). Tricellular junctions: a hot corner of epithelial biology. *Curr. Opin. Cell Biol.* 54, 80–88. <https://doi.org/10.1016/j.ceb.2018.05.002>.
10. Brodland, G.W., Veldhuis, J.H., Kim, S., Perrone, M., Mashburn, D., and Hutson, M.S. (2014). CellFIT: A Cellular Force-Inference Toolkit Using Curvilinear Cell Boundaries. *PLoS One* 9, e99116. <https://doi.org/10.1371/journal.pone.0099116>.
11. Roffay, C., Chan, C.J., Guirao, B., Hiiragi, T., and Graner, F. (2021). Inferring cell junction tension and pressure from cell geometry. *Development* 148, dev192773. <https://doi.org/10.1242/dev.192773>.
12. Vasan, R., Maleckar, M.M., Williams, C.D., and Rangamani, P. (2019). DLITE Uses Cell-Cell Interface Movement to Better Infer Cell-Cell

- Tensions. *Biophys. J.* 117, 1714–1727. <https://doi.org/10.1016/j.bpj.2019.09.034>.
13. Aigouy, B., Cortes, C., Liu, S., and Prud'Homme, B. (2020). EPySeg: a coding-free solution for automated segmentation of epithelia using deep learning. *Development* 147, dev194589. <https://doi.org/10.1242/dev.194589>.
14. Pachitariu, M., and Stringer, C. (2022). Cellpose 2.0: how to train your own model. *Nat. Methods* 19, 1634–1641. <https://doi.org/10.1038/s41592-022-01663-4>.
15. Purcell, E.M. (1977). Life at low Reynolds number. *Am. J. Phys.* 45, 3–11. <https://doi.org/10.1119/1.10903>.
16. Bambardekar, K., Clément, R., Blanc, O., Chardès, C., and Lenne, P.-F. (2015). Direct laser manipulation reveals the mechanics of cell contacts in vivo. *Proc. Natl. Acad. Sci. USA* 112, 1416–1421. <https://doi.org/10.1073/pnas.1418732112>.
17. Clément, R., Dehapiot, B., Collinet, C., Lecuit, T., and Lenne, P.-F. (2017). Viscoelastic Dissipation Stabilizes Cell Shape Changes during Tissue Morphogenesis. *Curr. Biol.* 27, 3132–3142.e4. <https://doi.org/10.1016/j.cub.2017.09.005>.
18. Ventura, G., Amiri, A., Thiagarajan, R., Tolonen, M., Doostmohammadi, A., and Sedzinski, J. (2022). Multiciliated cells use filopodia to probe tissue mechanics during epithelial integration in vivo. *Nat. Commun.* 13, 6423. <https://doi.org/10.1038/s41467-022-34165-0>.
19. Nizak, C., Martin-Lluesma, S., Moutel, S., Roux, A., Kreis, T.E., Goud, B., and Perez, F. (2003). Recombinant Antibodies Against Subcellular Fractions Used to Track Endogenous Golgi Protein Dynamics *in Vivo*: Recombinant Antibodies for Cell Biology. *Traffic* 4, 739–753. <https://doi.org/10.1034/j.1600-0854.2003.00132.x>.
20. Hashimoto, H., Robin, F.B., Sherrard, K.M., and Munro, E.M. (2015). Sequential Contraction and Exchange of Apical Junctions Drives Zippering and Neural Tube Closure in a Simple Chordate. *Dev. Cell* 32, 241–255. <https://doi.org/10.1016/j.devcel.2014.12.017>.
21. Ghysen, A., and Dambly-Chaudière, C. (2007). The lateral line microcosmos. *Genes Dev.* 21, 2118–2130. <https://doi.org/10.1101/gad.1568407>.
22. Olson, H.M., and Nechiporuk, A.V. (2018). Using Zebrafish to Study Collective Cell Migration in Development and Disease. *Front. Cell Dev. Biol.* 6, 83. <https://doi.org/10.3389/fcell.2018.00083>.
23. Scarpa, E., and Mayor, R. (2016). Collective cell migration in development. *J. Cell Biol.* 212, 143–155. <https://doi.org/10.1083/jcb.201508047>.
24. Hava, D., Forster, U., Matsuda, M., Cui, S., Link, B.A., Eichhorst, J., Wiesner, B., Chitnis, A., and Abdellah-Seyfried, S. (2009). Apical membrane maturation and cellular rosette formation during morphogenesis of the zebrafish lateral line. *J. Cell Sci.* 122, 687–695. <https://doi.org/10.1242/jcs.032102>.
25. Ernst, S., Liu, K., Agarwala, S., Moratscheck, N., Avci, M.E., Dalle Nogare, D., Chitnis, A.B., Ronneberger, O., and Lecaudey, V. (2012). Shroom3 is required downstream of FGF signalling to mediate proneuromast assembly in zebrafish. *Development* 139, 4571–4581. <https://doi.org/10.1242/dev.083253>.
26. Hernández, P.P., Olivari, F.A., Sarrazin, A.F., Sandoval, P.C., and Allende, M.L. (2007). Regeneration in zebrafish lateral line neuromasts: Expression of the neural progenitor cell marker *sox2* and proliferation-dependent and-independent mechanisms of hair cell renewal. *Dev. Neurobiol.* 67, 637–654. <https://doi.org/10.1002/dneu.20386>.
27. Berg, S., Kutra, D., Kroeger, T., Straehle, C.N., Kausler, B.X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., et al. (2019). ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods* 16, 1226–1232. <https://doi.org/10.1038/s41592-019-0582-9>.
28. Kozak, E.L., Miranda-Rodríguez, J.R., Borges, A., Dierkes, K., Mineo, A., Pinto-Teixeira, F., Viader-Llargués, O., Solon, J., Chara, O., and López-Schier, H. (2023). Quantitative videomicroscopy reveals latent control of cell-pair rotations *in vivo*. *Development* 150, dev200975. <https://doi.org/10.1242/dev.200975>.
29. Wibowo, I., Pinto-Teixeira, F., Satou, C., Higashijima, S., and López-Schier, H. (2011). Compartmentalized Notch signaling sustains epithelial mirror symmetry. *Development* 138, 1143–1152. <https://doi.org/10.1242/dev.060566>.
30. Jacobo, A., Dasgupta, A., Erzberger, A., Siletti, K., and Hudspeth, A.J. (2019). Notch-Mediated Determination of Hair-Bundle Polarity in Mechanosensory Hair Cells of the Zebrafish Lateral Line. *Curr. Biol.* 29, 3579–3587.e7. <https://doi.org/10.1016/j.cub.2019.08.060>.
31. Erzberger, A., Jacobo, A., Dasgupta, A., and Hudspeth, A.J. (2020). Mechanochemical symmetry breaking during morphogenesis of lateral-line sensory organs. *Nat. Phys.* 16, 949–957. <https://doi.org/10.1038/s41567-020-0894-9>.
32. Brakke, K.A. (1992). The Surface Evolver. *Exp. Math.* 1, 141–165. <https://doi.org/10.1080/10586458.1992.10504253>.
33. Borges, A. (2024). borgesaugusto/seapipy: v0.2.0-alpha. Version v0.2.0-alpha. <https://doi.org/10.5281/ZENODO.10809290>.
34. Dealy, J.M. (2010). Weissenberg and Deborah numbers—Their definition and use. *Rheology Bulletin* 79, 14–18.
35. Poole, R.J. (2012). The Deborah and Weissenberg numbers. *Rheology Bulletin* 53, 32–39.
36. Thompson, R.L., and Oishi, C.M. (2021). Reynolds and Weissenberg numbers in viscoelastic flows. *J. Nonnewton. Fluid Mech.* 292, 104550. <https://doi.org/10.1016/j.jnnfm.2021.104550>.
37. Hartmann, J., Wong, M., Gallo, E., and Gilmour, D. (2020). An image-based data-driven analysis of cellular architecture in a developing tissue. *eLife* 9, e55913. <https://doi.org/10.7554/eLife.55913>.
38. Yamaguchi, N., Zhang, Z., Schneider, T., Wang, B., Panozzo, D., and Knaut, H. (2022). Rear traction forces drive adherent tissue migration in vivo. *Nat. Cell Biol.* 24, 194–204. <https://doi.org/10.1038/s41556-022-00844-9>.
39. Borges, A., and Chara, O. (2024). Peeking into the future: inferring mechanics in dynamical tissues. *Biochem. Soc. Trans.* 52, 2579–2592. <https://doi.org/10.1042/BST20230225>.
40. Grashoff, C., Hoffman, B.D., Brenner, M.D., Zhou, R., Parsons, M., Yang, M.T., McLean, M.A., Sligar, S.G., Chen, C.S., Ha, T., and Schwartz, M.A. (2010). Measuring mechanical tension across vinculin reveals regulation of focal adhesion dynamics. *Nature* 466, 263–266. <https://doi.org/10.1038/nature09198>.
41. Borghi, N., Sorokina, M., Shcherbakova, O.G., Weis, W.I., Pruitt, B.L., Nelson, W.J., and Dunn, A.R. (2012). E-cadherin is under constitutive actomyosin-generated tension that is increased at cell-cell contacts upon externally applied stretch. *Proc. Natl. Acad. Sci. USA* 109, 12568–12573. <https://doi.org/10.1073/pnas.1204390109>.
42. Campàs, O., Mammoto, T., Hasso, S., Sperling, R.A., O'Connell, D., Bischof, A.G., Maas, R., Weitz, D.A., Mahadevan, L., and Ingber, D.E. (2014). Quantifying cell-generated mechanical forces within living embryonic tissues. *Nat. Methods* 11, 183–189. <https://doi.org/10.1038/nmeth.2761>.
43. LeGoff, L., Rouault, H., and Lecuit, T. (2013). A global pattern of mechanical stress polarizes cell divisions and cell shape in the growing *Drosophila* wing disc. *Development* 140, 4051–4059. <https://doi.org/10.1242/dev.090878>.
44. 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. <https://doi.org/10.1038/nmeth.2019>.
45. Borges, A., Ceccarelli, A.S., and Chara, O. (2024). ForSys software (Zenodo). <https://doi.org/10.5281/ZENODO.11282554> Version v1.1.0.
46. Haas, P., and Gilmour, D. (2006). Chemokine Signaling Mediates Self-Organizing Tissue Migration in the Zebrafish Lateral Line. *Dev. Cell* 10, 673–680. <https://doi.org/10.1016/j.devcel.2006.02.019>.
47. Aigouy, B., Umetsu, D., and Eaton, S. (2016). Segmentation and Quantitative Analysis of Epithelial Tissues. In *Drosophila Methods in Molecular*

- Biology, C. Dahmann, ed. (New York: Springer), pp. 227–239. https://doi.org/10.1007/978-1-4939-6371-3_13.
48. Kindt, K.S., Finch, G., and Nicolson, T. (2012). Kinocilia Mediate Mechano-sensitivity in Developing Zebrafish Hair Cells. *Dev. Cell* 23, 329–341. <https://doi.org/10.1016/j.devcel.2012.05.022>.
 49. Parslow, A., Cardona, A., and Bryson-Richardson, R.J. (2014). Sample drift correction following 4D confocal time-lapse imaging. *J. Vis. Exp.* 51086, 51086. <https://doi.org/10.3791/51086>.
 50. Kreshuk, A., and Zhang, C. (2019). Machine Learning: Advanced Image Segmentation Using ilastik. In *Computer Optimized Microscopy Methods in Molecular Biology*, E. Rebollo and M. Bosch, eds. (Springer New York), pp. 449–463. https://doi.org/10.1007/978-1-4939-9686-5_21.
 51. Xiao, Y., Faucherre, A., Pola-Morell, L., Heddlestone, J.M., Liu, T.-L., Chew, T.-L., Sato, F., Sehara-Fujisawa, A., Kawakami, K., and López-Schier, H. (2015). High-resolution live imaging reveals axon-glia interactions during peripheral nerve injury and repair in zebrafish. *Dis. Model. Mech.* 8, 553–564. <https://doi.org/10.1242/dmm.018184>.
 52. Viader-Llargués, O., Lupperger, V., Pola-Morell, L., Marr, C., and López-Schier, H. (2018). Live cell-lineage tracing and machine learning reveal patterns of organ regeneration. *eLife* 7, e30823. <https://doi.org/10.7554/eLife.30823>.
 53. Farhadifar, R., Röper, J.-C., Aigouy, B., Eaton, S., and Jülicher, F. (2007). The Influence of Cell Mechanics, Cell-Cell Interactions, and Proliferation on Epithelial Packing. *Curr. Biol.* 17, 2095–2104. <https://doi.org/10.1016/j.cub.2007.11.049>.
 54. Nagai, T., and Honda, H. (2001). A dynamic cell model for the formation of epithelial tissues. *Philos. Mag. B* 81, 699–719. <https://doi.org/10.1080/13642810108205772>.
 55. Spencer, M.A., Jabeen, Z., and Lubensky, D.K. (2017). Vertex stability and topological transitions in vertex models of foams and epithelia. *Eur. Phys. J. E* 40, 2. <https://doi.org/10.1140/epje/i2017-11489-4>.
 56. Lawson, C.L., and Hanson, R.J. (1995). Solving Least Squares Problems (Society for Industrial and Applied Mathematics). <https://doi.org/10.1137/1.9781611971217>.
 57. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>.
 58. Newville, M., Otten, R., Nelson, A., Stensitzki, T., Ingargiola, A., Allan, D., Fox, A., Carter, F., Michał, Osborn, R., et al. (2023). Imfit/Imfit-py: 1.2.2 (Zenodo). <https://doi.org/10.5281/ZENODO.598352> Version 1.2.2. .

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Experimental models: Organisms/strains		
Zebrafish Tg[myo6b:actb1-EGFP]	Kindt lab	https://doi.org/10.1016/j.devcel.2012.05.022
Zebrafish Tg[-8.0cldnb:Lyn-EGFP]	Gilmour lab	https://doi.org/10.1016/j.devcel.2006.02.019
Zebrafish TgBAC(cxcr4b:H2B-RFP)	Lecaudey Lab	https://doi.org/10.1038/s41467-018-06094-4
Xenopus laevis	Sedzinski Lab	N/A
Recombinant DNA		
SF9-3xGFP myosin II sensor	Sedzinski lab	https://doi.org/10.1038/s41467-022-34165-0
Software and algorithms		
seapipy v0.2.1	Repository	https://doi.org/10.5281/zenodo.10853567
ForSys v1.1.2	This paper	https://doi.org/10.5281/zenodo.15789474
Epyseg v0.1.52	Prud'Homme Lab	https://doi.org/10.1242/dev.194589
Tissue Analyzer	Eaton lab	https://doi.org/10.1007/978-1-4939-6371-3_13
Ilstik v1.3.3	Kreshuk lab	https://doi.org/10.1038/s41592-019-0582-9

EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS

SF9 myosin II sensor intensity measurements

Images of stage 16 to stage 20 *Xenopus* embryos expressing the SF9-3xGFP myosin II sensor were acquired in using a 3i spinning disk microscope with a Plan-Apochromat $\times 63$ oil objective (N.A. = 1.4) mounted on an inverted Zeiss Axio Observer Z1 microscope (Marianas Imaging Workstation [3i—Intelligent Imaging Innovations]), equipped with a CSU-X1 spinning disk confocal head (Yokogawa) and an iXon Ultra 888 EM-CCD camera (Andor Technology). From these images we obtained maximum intensity Z projections, which were then used to extract myosin intensity values of the epithelial junctions. After automatic segmentation with EPySeg, cell membranes were manually corrected, and the final shape was obtained by using a third-order Savitzky–Golay filter with a window of 5 pixels. For all vertices constituting a membrane in the segmentation, smoothed intensity values were first obtained by taking the median over first neighbors. The intensity value for each membrane is then defined as the mean of smoothed intensities at each of its constituting vertices. Then, to allow comparison with the inferred stresses, these values were normalized to a mean value of one for each embryo.

Zebrafish primordium migration experiments

Zebrafish carrying the Tg[-8.0cldnb:Lyn-EGFP]⁴⁶ were kept under standard conditions at 28.5°C. All zebrafish experiments were performed in accordance with protocols approved by the Ethical Committee of Animal Experimentation of the Helmholtz Zentrum München, the German Animal Welfare act Tierschutzgesetz §11, Abs. 1, Nr. 1, Haltungserlaubnis according to the European Union animal welfare, and under protocol number Gz.:55.2-1-54-2532-202-2014 and Gz.:55.2-2532.Vet_02-17-187 from the “Regierung von Oberbayern” (Germany).

At 40–48 hours post-fertilization, larvae were anesthetized with 1.5% MS222 in Danieau Buffer and mounted in 0.8% low-melting point agarose on a glass-bottom petri dish. Larvae were imaged in a custom-built Zeiss inverted spinning disk confocal microscope. 16 slices Z stacks of the migrating primordium II were acquired every two minutes with a 63X objective. Subsequently, one z-slice was manually selected from each frame, and the membrane image was segmented using Tissue Analyzer.⁴⁷ The image segmentations were used for ForSys predictions, and the cells’ centroids’ X and Y coordinates, the time point (frame number), and the cell pressures were exported. The probability density function of the cell position along the anteroposterior axis was estimated via a Gaussian kernel in the R statistical software. The value of cell pressure was used as a weight in the density estimation. From this density curve, local maxima were determined by an *ad-hoc* algorithm implemented in R, which compares the sign of the first and second derivatives.

Zebrafish neuromasts videomicroscopy and analysis

The videomicroscopy data previously published by Kozak and colleagues was acquired as described and re-analyzed.²⁸ Zebrafish larvae (*Danio rerio*) carrying the transgenics myo6b:β-actin-GFP⁴⁸ and *Tg*[−8.0*cldnb*:*Lyn*-EGFP]⁴⁶ were anesthetized in MS222 and mounted in 1% low-melting point agarose. Larvae were imaged simultaneously using a Zeiss custom-built inverted spinning-disc confocal microscope with a 63× water-immersion objective. Stacks of 16–20 z-slices 1 μm apart were acquired every 200 s. Hair cells can be identified by the myo6b:β-actin-GFP staining in the apical side, while *Tg*[−8.0*cldnb*:*Lyn*-EGFP] marks all neuromast membranes and allows cell segmentation. In total, 14 hair cell divisions were included in the analysis, and partitioned into the “inverting” or “non-inverting” group by whether the final position of the sibling hair cells is the opposite with respect to the anteroposterior axis.²⁸ All 4D movies were processed using FIJI software.⁴⁴ Stacks were centered by laying point regions of interest at time frames of significant drift and then running the Manual Drift Correction plugin and then further registering for z-slice drifts using the plugin Correct 3D drift.⁴⁹

After registration, one z-slice per time point was selected to segment membranes, using the Autocontext workflow from ilastik v1.3.3.⁵⁰ The resulting probabilities were loaded in the multicut segmentation workflow to get a skeletonized segmentation of cells. These automated segmentations were loaded into TissueAnalyzer⁴⁷ and manually corrected and semi automatically tracked. From the software Tissue analyzer, we exported two types of data: (1) cell tracking data containing x and y centroid position, cell area, perimeter (in pixels) and an ID identifying individual cells through time; (2) bond tracking data, indicating the identity of cells sharing a membrane segment, and the length of the membrane segment (in pixels).

The same segmentation images were then loaded into ForSys for stress inference and merged with the Tissue Analyzer data thereby getting an integrated dataset where it is known the identity of each cell, its pressure inference, and the membrane tension estimation for each membrane segment.

For Figure 6E, fish additionally carrying the transgenic TgBAC(cxc4b:H2B-RFP) were used to label nuclei.

Zebrafish laser ablation experiments

Zebrafish larvae *Tg*[−8.0*cldnb*:*Lyn*-EGFP] 5 days post fertilization were prepared identically and imaged in the same microscope as described in STAR Methods section “zebrafish neuromasts videomicroscopy and analysis”. The laser ablation was performed with a iLasPulse laser system (Roper Scientific SAS, Evry, France) mounted on a Zeiss Axio Observer inverted microscope equipped with a 63× water objective lens., as described in previous works.^{51,52} Thirty-seven neuromasts were ablated with only one ablation per fish. For each ablation, the microscope was centered in a L2 or L3 neuromast. Time Lapse movies were taken with a frequency of one image per second. Three laser pulses (355 nm, 400ps) were automatically triggered between the 3rd and 4th frame to acquire microscopy images before and after the ablation of the membrane in a small circular ROI in the mid part of the cellular membrane (Figures 7A and 7B). We manually tracked the position of the tricellular junctions belonging to the ablated membrane as a function of time. We used the membrane’s recoil as a function of time in an interval just after the ablation to calculate the scale parameter (Figures S7C and S7D). Details of this calculation are in the “scale parameter selection *in vivo*” of the main text.

METHOD DETAILS

Applying ForSys to *in vivo* zebrafish microscopy

A basal z-slice was selected for each frame of a time-lapse video microscopy. Each frame was segmented with EPySeg¹³ and then manually curated by loading the complete z-volume into tissue analyzer.⁴⁷ The resulting images are skeletonized segmentations of cell membranes, where pixels can only take the value of 1, for membranes, and 0 otherwise. The skeletons are 8-connected and each white pixel has exactly two neighbouring white pixels except for tricellular junction pixels, which have three. The skeleton images can be directly used as input to ForSys Skeleton() function. A preliminary run of ForSys on the segmented images was used to diagnose the accuracy of the tracking for tricell junctions from one frame to the next. Then a JSON file was created to correct any mistracking by labelling the connections between timepoints. As described in the STAR Methods section “determination of the scale parameter *in silico*”, we used the estimated value of $\rho = 0.1$ for the scale parameter in the dynamic modality. Finally, ForSys outputs for each time-lapse experiment the stress and identity of the membranes, as well as the pressure and identity of the cells in the system.

The conceptual model behind ForSys

ForSys uses microscopy images as input to estimate the mechanical state of the tissue. The software extracts vertices, edges, and cells from the segmentation. ForSys accepts skeletonized segmentation, such as those generated with EPySeg¹³ as well as labeled masks, generated for example with Cellpose.¹⁴ After ForSys reads the input data, the algorithm works identically regardless of the microscopies’ input format (Figure S4). Although most vertices separate two edges, a number of them connect three or more edges and are central for stress inference. We call these pivot vertices or junctions. ForSys calculates the mechanical stresses operating on each edge while assuming mechanical equilibrium in each vertex. Conveniently, this creates a system of equations representing the geometrical state of the tissue.^{10,12} One equation per coordinate is built from every pivot vertex using force balance at the junction.

In the dynamic modality, ForSys assumes that each vertex is in an overdamped regime, where a viscous damping force proportional to the velocity balances the mechanical stresses at that vertex. This creates a non-homogeneous system of equations where the inhomogeneity is proportional to the vertex velocity. In both static and dynamic modalities, the resulting system of equations is

solved through a Least Squares minimization with the constraint that the average tension equals one (see “[solving the system of equations](#)” in [STAR Methods](#) for more details).

Finally, ForSys uses the stress inferred as an input to estimate cellular pressure within the tissue. For this, a Young-Laplace equation is built at each cell-cell membrane, and the corresponding system is solved similarly to the stresses. However, this requires that the mean pressure of the system is equal to zero (see section [static stress inference](#) for details). ForSys renders intercellular stresses as a color code of the cellular outlines, specifically at the edges. Similarly, intracellular pressures are depicted in a color code within the cytoplasmic area of the cells. Moreover, the numerical values of the inference and other observables are easily exportable, facilitating further analysis of the mechanical state of the tissue.

seAPIpy: Generation of *in silico* tissues to validate ForSys

To validate the accuracy of ForSys, we compared the intercellular mechanical stresses inferred by the software with a ground truth distribution of stresses within the tissue. To establish the ground truth, we employed a cell-based computational model to simulate tissues with known intracellular pressures and intercellular stress patterns. Specifically, we employed the vertex model, which is particularly suitable for mechanically evolving epithelial tissues.^{53,54} For the implementation of the vertex model, we utilized Surface Evolver software.³² To facilitate the integration and streamline the simulation process, we developed a Python-based software called seapipy.³³ This open-source computational tool enables Python scripting to generate the desired initial tissue conditions and simulate them using a vertex model implemented in Surface Evolver. seAPIpy generates a Voronoi tessellation with a given geometry as a starting configuration and assigns initial stresses to the edges (Figure S3). Through seAPIpy functions, the user may add Surface Evolver commands to create the desired conditions for evolution and generate the Surface Evolver-compatible file.

By leveraging both ForSys and the capabilities of seAPIpy software, we implemented four conditions as examples that were later used to test ForSys stress inference in both its static and dynamic modalities. The first two conditions induce a furrow formation on vertical and horizontal strips, respectively. In the third condition, a central zone of elevated stress is introduced, which diminishes radially. Lastly, a fourth condition assigns five different random stresses to edges, following a uniform distribution, with a 50 % spread in stress values. Each condition underwent twenty-five repetitions. These simulations served as the ground truths for validating ForSys *in silico*, as shown in the following two sections (Figures 2A, 3A, and S3).

We generated four examples to validate our software *in silico*. In all four cases, tissues evolve until a time zero is defined. The stresses are modified according to a prescribed condition, and the tissue evolves for shorter periods while it relaxes.

We generated the initial condition in each example by creating a Voronoi tessellation from $N = 64$ points in a rectangular grid. Each point in the grid is moved with a Gaussian noise centered at zero. Initial cell target areas are randomly assigned as $A = 450 \pm 5$ (mean $\pm$ std) from a normal distribution. The initial stress of each edge is also taken from a normal distribution centered at 1 with a standard deviation of 0.1. From this state, the tissue evolved through several rounds of vertex averaging and T1 swaps with varying scales.

We defined time as the number of steps elapsed, multiplied by the scale ($t = n \Delta t$), and call it Surface Evolver Time (SET). The first time point is generated after 3875 SET, after which the tissue is evolved for an additional 125 SET. At this point, membrane stresses are changed according to each condition, and each simulation snapshot is saved every 0.25 SET.

In the conditions corresponding to the horizontal and vertical furrows, the new tensions are generated by summing the value corresponding to the position of the center of an edge in the probability density function (PDF) of a normal distribution to the initial randomized value. The normal distribution has its maximum at the centroid of the tissue and a standard deviation of ~ 2 cellular radii. Vertical furrows have the PDF on the y-axis, and horizontal furrows on the x-axis. Similarly, the circular furrow uses the distance of the edge's center to the tissue's centroid to calculate the new stress. Finally, in the condition corresponding to the “random examples”, tensions are assigned from normal distributions with a 50% spread around five possible values (1, 1.1, 1.2, 1.3, 1.5) chosen uniformly. Cellular pressures are calculated by Surface Evolvers as Lagrange multipliers, by taking into account the area constraints of each cell.

Therefore, seapipy facilitates testing multiple *in silico* examples and has an easy integration into the analysis pipeline. seapipy offers advantages over an existing package (python-evolver; <https://github.com/elmisback/python-evolver>) because it incorporates Surface Evolver syntax directly into the Python code, eliminating the need to write Surface Evolver commands manually into the input files. seapipy allows for a systematic and straightforward generation of *in silico* ground truths, enabling a better exploration of strengths and limitations in stress inference tools.

Static stress inference

We assume a 2D tissue with C cells, representing each cell as a polygon. The system consists of V vertices and E edges in total. Each edge is composed of two vertices. We define pivot vertices as those that correspond to junctions between three cells or are at the border of the tissue. This method can be applied to junctions shared by more than three cells but at the risk of losing stability in the underlying model.⁵⁵ All vertices between two pivots are regarded as virtual, and only pivot vertices are used to compute stresses. We then use Newton's second law and assume mechanical equilibrium to assert that the sum of forces at each pivot vertex equals zero. We can calculate the force acting on each vertex as a sum of the contributions of the forces along the edges connected to it. Mathematically, the force at each pivot vertex will have an equation in the form

$$\vec{F}_i = \sum_{\langle ij \rangle} \lambda_{ij} \vec{r}_{ij} \quad (\text{Equation 2})$$

Where i and j indicate the vertices i and j , $\vec{F}_i$ is the force on vertex i , λ_{ij} is the edge force modulus in that edge, and $\vec{r}_{ij}$ is the versor along the edge starting at vertex i . The sum is done over all j vertices connected to the vertex i . Note that $\lambda_{ij} = \lambda_{ji}$. The direction of the r_{ij} versor is obtained by fitting a circle to the corresponding membrane, following other authors.^{10,12}

Applying Equation 2 to all the vertices in the tissue will translate into a homogeneous set of linear equations that have to be solved simultaneously with the edge tensions (λ_{ij}) as unknowns. Hence, we write Equation 2 and equate it to zero for each system vertex to guarantee that all the forces are balanced. Each of these V equations will be written as

$$\lambda_{i1} \vec{r}_{i1} + \lambda_{i2} \vec{r}_{i2} + \dots + \lambda_{iN} \vec{r}_{iN} = 0 \quad (\text{Equation 3})$$

this equation corresponds to the i th vertex, and the edge tensions λ are the unknowns.

Similarly, it is possible to infer the pressures of each cell in the tissue by assembling a system of equations that connects the stress at each membrane with its curvature. The Young-Laplace equation relates these quantities with the pressure difference between two neighboring cells. Symbolically,

$$P_j - P_i = \lambda_{ij} \rho_{ij} \quad (\text{Equation 4})$$

Where P_i is the pressure of cell i , λ_{ij} is the stress of the membrane shared by cells i and j , and ρ_{ij} is the curvature of the shared membrane. This leads to a system of E equations, one per edge, and C unknowns.

The dynamic inference case

The static inference algorithm assumes that vertices do not move. To perform stress inference in a dynamic tissue where vertices are moving, we modified the static algorithm to include vertex movement. If the system has a low Reynolds number, viscous forces dominate the dynamics over inertial components; Equation 2 can be modified, assuming a constant viscosity throughout the tissue, to incorporate viscous forces as

$$\vec{F}_i = \sum_{\langle ij \rangle} \lambda_{ij} \vec{r}_{ij} - \eta \vec{v}_i \quad (\text{Equation 5})$$

where $\vec{v}_i$ is the velocity of vertex i and η is the viscous damping coefficient of the tissue. This would modify the coupled system of equations, which could be rearranged to get the i th vertex

$$\sum_{\langle ij \rangle} \lambda_{ij} \vec{r}_{ij} = \eta \vec{v}_i \quad (\text{Equation 6})$$

To determine the scales correctly, we proceeded to make Equation 6 nondimensional. For this, we redefine the stresses by using an unknown reference stress $\lambda'_j = \frac{\lambda_j}{\lambda}$. We take this reference stress as the average stress in the system. We used a reference velocity defined as the time average over all the frames of the mean junction velocity

$$\bar{v} = \frac{1}{N_{frames}} \sum_{t=t_1}^{t_f} \sum_i \frac{\|\vec{v}_i^t\|}{N_v^t} \quad (\text{Equation 7})$$

Combining these equations gives a nondimensional expression for the force balance at each junction

$$\sum_j \lambda'_j \vec{r}_{ij} = \left(\frac{\eta \bar{v}}{\lambda} \right) \frac{\vec{v}_i}{\bar{v}} \quad (\text{Equation 8})$$

Importantly, this led to the nondimensional parameter $\frac{\eta \bar{v}}{\lambda}$. Even though the right-hand side of Equation 5 is not a viscosity but rather a damping coefficient, we can interpret it as such in this context. Therefore, as the stress of each membrane represents the elastic forces in the system, this parameter can be interpreted as the relation between the elastic and the viscous forces acting on the system. We call this number the scale parameter, ρ ($\rho = \frac{\eta \bar{v}}{\lambda}$).

Solving the system of equations

In static and dynamic cases, it is necessary to solve a system of linear equations with homogeneous and inhomogeneous conditions, respectively. In both cases, we will turn the system into its matrix form, add a constraint to the unknowns through a Lagrange multiplier, and convert it into a least squares problem. Finally, we will attempt to invert the resulting matrix, and if that is not possible, we will use a numerical algorithm to find the best solution.

Given a two-dimensional tissue with V vertices and E edges, the system would have $2V$ equations, as each vertex has one equation per dimension and E unknowns, one for each edge. Following the method proposed by Brodland et al.,¹⁰ the set of equations in Equation 2 is then translated into matrix form as

$$M_{\lambda} X = B \quad (\text{Equation 9})$$

Where M_{λ} is a $2V \times E$ matrix with versor coefficients, X is the unknowns column matrix of $E \times 1$, and B is a $2V \times 1$ column matrix with either all zeros under static conditions or the velocity components for each vertex in the dynamical case. To avoid the null solution in the static case, one further condition is added: The mean value for the unknowns, i.e., λ_s , is set equal to one, using the equation

$$\sum_{k=1}^E \lambda_k = E \quad (\text{Equation 10})$$

where E is the number of edges and λ_k is the tension corresponding to the k th edge. In the matrix representation, this entails adding a Lagrange multiplier to the unknowns, a row and column of ones for the tension constraint, and a new row in the B matrix. Hence, the equations to be solved have $2V + 1$ equations and $E + 1$ unknowns. Therefore, the typical equation for the one of the components of vertex i will be of the form

$$\lambda_{ij} r_{ij}^x + \lambda_{ik} r_{ik}^x + \lambda_{il} r_{il}^x + L = \left(\frac{\eta \bar{V}}{\lambda} \right) \frac{v_i}{V} \quad (\text{Equation 11})$$

where j , k and l are vertices that connect to vertex i , and L is the Lagrange multiplier added by the constraint on the average value of stresses.

As this system might not always guarantee a solution, we transformed it using least squares. To this end, we apply the transpose matrix M_{λ}^T to the equation, giving a new system

$$M_{\lambda}^T M_{\lambda} X = M_{\lambda}^T B \quad (\text{Equation 12})$$

Symbolically,

$$M'_{\lambda} X' = B' \quad (\text{Equation 13})$$

On the other hand, the B matrix in Equations 6 and 7 for the dynamic case has the corresponding nondimensional velocity component multiplied by the scale parameter (ρ) described in Equation 8 in each row. Its final element has the number of edges E to enforce the constraint. To quantify the movement present in the tissue, we calculate the 2-norm of the B matrix, removing the last row, this vector is referred to as $|v|_2$. Each vertex is tracked through time to obtain the vertex velocity, and the forward velocity is calculated in all but the last step, where the backward expression is used. If a vertex cannot be followed in a frame, i.e., due to significant changes in the tissue shape, it is assigned a null velocity for the frames where it cannot be tracked.

Hence, to elucidate the acting forces within the tissue, the software attempts to solve it by inverting the M_{λ}' matrix, thus having a solution

$$X = M'^{-1} B' \quad (\text{Equation 14})$$

If the system is not invertible, i.e., M' is singular, or if any of the edge tensions found are negative, a Least Squares algorithm can be used to find the stress values, such as a Non-Negative Least Squares, SciPy's package or Imfit.^{56–58}

After solving the system, the calculated stresses can be used to infer the pressures of the cells. As seen from the Young-Laplace equation (Equation 4), pressures are expressed through an inhomogeneous system of linear equations. The left-hand side is a matrix with one column per cell and one row per membrane. Each row has two entries different from zero, one +1 and one -1, representing the difference in pressure at that membrane. The right-hand side consists of a column matrix with the product of each membrane's stresses and curvatures ($\lambda_{ij} \rho_{ij}$). Then, the equations are solved analogously to the stress case using the Least Squares with the constraint that the average pressure must be zero.

Determination of the scale parameter *in silico*

We performed a parameter sweep to determine the optimal value of the Scale Parameter (ρ) from *in silico* simulations, testing values ranging from 0 to 0.5. For each of the four simulation scenarios, and across five repetitions per scenario, we calculated a performance score at each time point (Figures S9A–S9D). The optimal value of ρ was selected as the median score across repetitions, which in each case also coincided with the mode (Figures S9E and S9F). The resulting values were: $\rho = 0.08$ for the x-axis furrow, $\rho = 0.07$ for the y-axis furrow, $\rho = 0.13$ for the circular furrow, and $\rho = 0.18$ for the random tension configuration. Based on these results, we set the default scale parameter to 0.1 in the dynamic inference mode of our software.

Interacting with the ForSys software

ForSys requires a Python (>3.8) environment to run. After installation, there are three ways to use the ForSys software. First, writing a Python script is the most comprehensive method, giving full access to all the information that ForSys processes. Example scripts can

be found in the documentation, and the packages repository, and in [Figure S1](#). Additionally, to improve the usability of the ForSys tool, we created a Command Line Interface (CLI), and a Graphical User Interface (GUI). The CLI allows the analysis of tissues, using dynamical or static inference directly from the command line, without needing to write any Python Scripts ([Figures S2A and S2C](#)). Possible commands to use are detailed in the package's documentation. Finally, a GUI was created by integrating ForSys to Fiji to allow a high level interaction with the software ([Figures S2B and S2C](#)).

QUANTIFICATION AND STATISTICAL ANALYSIS

Evaluating goodness of fit

We evaluated the goodness of fit of the inferred data to the ground truth using a tailored saturated score function. This score combines the Pearson correlation coefficient (p), the Mean Absolute Percentage Error (M), and the coefficient of determination (r) as

$$s(M, p, r) = \frac{\alpha}{M} + \frac{\beta}{2} \frac{1+p}{1-p} + \frac{\gamma}{1-r} \quad (\text{Equation 15})$$

where α , β and γ are free parameters set to one. As this function is unbounded from above, we saturate the score at $s = 299.5$ for representation purposes in [Figures 2, 3](#), and [S9](#). This value comes from an error of 1 %, i.e. $s(0.01, 0.99, 0.99) = 299.5$.

The three components of the score are displayed separately for the *in silico* static validation of ForSys in [Figures 2D](#), [S6A](#), and [S6B](#), and for the dynamic validation in [Figures S7A–S7C](#).

Statistical estimators

To compare distributions, the Mann-Whitney U test was used with different alternative hypotheses, depending on whether we tested for stochastic ordering or whether distributions are different. In all *in silico* cases, the number of samples is twenty-five, which is the number of repetitions per condition. The Pearson correlation coefficient (R) was used when we evaluated correlations. The number of samples in each case is indicated when reporting the p -value.

Comparing ForSys with other computational methods

We tested the similarity of the static implementation of ForSys with two other established software: CellFIT¹⁰ and DLITE.¹² To this end, we applied the DLITE python package to solve the four *in silico* examples used throughout this work, taking advantage of its CellFIT modality. We found that inferred stress for the last frame of each of the examples are similar ([Figure S5A](#)) and that the stress distributions emerging from the solution are roughly identical ([Figure S5B](#)). Moreover, for the accumulated data of all repetitions for each example at the last simulated frame, we found that ForSys static performs better than DLITE, except in the random example ([Figure S5C](#)).

Moreover, we generated an artificial normal distribution to measure the relative differences with a first moment of 1 and a second moment equal to 0.2. We calculated the Wasserstein Distance between the *in silico* distributions and the normal generated randomly. Given two distributions, X and Y , the Wasserstein Distance is zero if and only if the two distributions are equal. The distance between two distributions can be arbitrarily large for increasingly different shapes.

The Wasserstein Distance is almost zero in all cases, indicating that the distributions gathered from the three inference methods are similar. To compare its similarity, we used an artificially generated normal distribution. Using this metric, we found that the methods among themselves are ~ 30 times closer in the x-furrow and y-furrow, 10 times closer for the circular case, and ~ 5 times closer in the random densities example than to the normal distribution.