pyTFM: A tool for Traction Force and Monolayer Stress Microscopy

Andreas Bauer[1*], Magdalena Prechová[2*], Martin Gregor[2*], Ben Fabry[1]

1 Department of Physics, Friedrich-Alexander University Erlangen-Nürnberg, Erlangen, Germany

2 Laboratory of Integrative Biology, Institute of Molecular Genetics of the Czech Academy of Sciences, Prague, Czech Republic

* andreas.b.bauer@fau.de

Abstract

Cellular force generation and force transduction are of fundamental importance for numerous biological processes and can be studied with the methods of Traction Force Microscopy (TFM) and Monolayer Stress Microscopy. Traction Force Microscopy and Monolayer Stress Microscopy solve the inverse problem of reconstructing cell-matrix tractions and inter- and intra-cellular stresses from the measured cell force-induced deformations of an adhesive substrate with known elasticity. Although several laboratories have developed software for Traction Force Microscopy and Monolayer Stress Microscopy computations, there is currently no software package available that allows non-expert users to perform a full evaluation of such experiments. Here we present pyTFM, a tool to perform Traction Force Microscopy and Monolayer Stress Microscopy on single cells, cell patches and cell layers grown in a 2-dimensional environment. pyTFM was optimized for ease-of-use; it is open-source and well documented (hosted at `https://pytfm.readthedocs.io/`) including usage examples and explanations of the theoretical background. pyTFM can be used as a standalone Python package or as an add-on to the image annotation tool *ClickPoints*. In combination with the *ClickPoints* environment, pyTFM allows the user to set all

necessary analysis parameters, select regions of interest, examine the input data and intermediary results, and calculate a wide range of parameters describing forces, stresses, and their distribution. The Monolayer Stress Microscopy implementation in pyTFM allows for the analysis of small cell patches and single cells; we analyze the accuracy and performance of Traction Force Microscopy and Monolayer Stress Microscopy algorithms using synthetic and experimental data from epithelial cell patches.

1 Introduction

The generation of active forces gives cells the ability to sense the mechanical properties of their surroundings [1], which in turn can determine the cell fate during differentiation processes [2], the migratory behavior of cells [3] or the response to drugs [4].

Measuring cellular force generation is important for understanding fundamental biological processes including wound healing [5], tissue development [6], metastasis formation [7, 8] and cell migration [3].

Cellular forces can be divided into three categories: Forces that are transmitted between a cell and its surrounding matrix (also referred to as traction forces), forces that are transmitted between cells, and forces that are transmitted inside cells.

Traction forces can be measured with Traction Force Microscopy (TFM), which is most easily applied to cells grown in a 2-dimensional environment: Cells are seeded on a planar elastic substrate on which they adhere, spread, and exert forces. The substrate contains fiducial markers such as fluorescent beads for tracking cell force-induced deformations of the substrate. Typically, the substrate is imaged in a tensed and a relaxed (force-free) state, whereby force relaxation is achieved by detaching the cells from the substrate. These two images are then compared to quantify substrate deformations, either by tracking each individual marker bead, or more commonly, by cross-correlation based Particle Image Velocimetry (PIV) [9].

The deformation field of the substrate is subsequently analyzed to calculate the cell-generated tractions in x- and y-directions. (Note that if the substrate deformations in z-direction are also measured, which requires at least one additional image taken at a different focal plane, it is possible to compute the tractions in z-direction [10]. In what follows, however, we ignore deformations and tractions in z-direction.) The calculation

of the traction field from the deformation field is an inverse problem for which a number of algorithms have been developed, including numerical methods [11, 12], Fourier-based deconvolution [13], and Finite Element (FE) computations [14], all of which have specific advantages and disadvantages (see Sabass et al. 2007 [15] for a detailed discussion). pyTFM uses the Fourier Transform Traction Cytometry (FTTC) algorithm [13], as it is computationally fast and does not require the location of the cell boundary as an additional input.

Tractions must be balanced by forces transmitted within or between cells. These forces are usually described by stress tensors. The stress tensor field for cells grown in a 2-dimensional environment can be calculated using the Monolayer Stress Microscopy method [16, 17], whereby the cell or cell patch is modeled as an elastically stretched 2-dimensional sheet with point-like contacts to the matrix so that the tractions are balanced by the internal stress of the elastic sheet.

In pyTFM, the cell or cell patch is modelled as a linear elastic sheet represented by a network of nodes and vertices so that the stresses can be calculated by a standard two-dimensional Finite Element Method (FEM). First, forces with the same magnitude but opposing direction to the local tractions are applied to each node. Then, internal strains and consequently stresses are calculated based on the network geometry and elastic properties.

pyTFM uses the Monolayer Stress Microscopy algorithm developed by Tambe et al. 2013 [17]. In this implementation, the calculated network strain has no physical meaning, as the matrix strain and the cell strain are not required to match [18]. Consequently, the Young's modulus of the elastic sheet has no influence on the stress estimation, and the Poisson's ratio has only a negligible influence. Both parameters can therefore be freely chosen [17]. Note that there are different implementations of Monolayer Stress Microscopy in which cell and matrix deformations are coupled and the network elasticity corresponds to the effective cell elasticity, which must be known to obtain correct results [19]. A comparison about these two approaches can be found in [18].

pyTFM uses a modified Monolayer Stress Microscopy algorithm for small cell patches. Stress microscopy for single cells and small cell patches suffers from the low spatial resolution of the TFM algorithm. A significant part of the tractions can seem to

originate from outside the cell area, and when only tractions beneath the cell area are considered, the stress field is underestimated. This problem cannot be remedied by constraining the tractions to be zero outside the cell area (constrained TFM) as this tends to produce large spurious tractions at the cell perimeter [13] and hence unphysically high stresses in the cell monolayer. Ng et al. 2014 [20] addressed this issue by expanding the FEM-grid to cover all tractions generated by the cell patch and by exponentially decreasing the stiffness of the FEM-grid with increasing distance to the cell patch edge. In our implementation, the FEM-grid is also expanded to cover all cell-generated tractions, however, we found it unnecessary to introduce a stiffness gradient in the FEM-grid. Moreover, zero-translation and zero-rotation constraints are explicitly added to the FEM-algorithm in pyTFM.

Finally, pyTFM adds a number user-friendly features to easily set parameters, select regions of interest and quickly evaluate results. For this, pyTFM can be optionally used as an add-on to the image annotation tool *ClickPoints* [21]. This makes the analysis of large data sets particularly easy by sorting input and output data in a database and allowing the user to browse through it.

pyTFM is well documented, including detailed usage examples, information on the theory of TFM and Monolayer Stress Microscopy, and explanations about the calculated parameters. The documentation is hosted at `https://pytfm.readthedocs.io`.

2 Design and implementation

pyTFM is a Python package implemented in Python 3.6. It is mainly intended to be used as an add-on for the image display and annotation tool *ClickPoints*, but can also be used as a stand-alone Python library.

pyTFM performs TFM and Monolayer Stress Microscopy following the workflow shown in Fig. 1A. The main steps of the workflow are the calculation of the deformation field from images of the cell substrate in a tensed and relaxed state, the calculation of the traction field, and the calculation of the monolayer stress field. The mathematical details of these steps are discussed in Section 2.2. Deformation, traction and stress fields are further analyzed to extract scalar measures of cellular stress, force generation, and force transduction between cells.

Fig 1. Workflow of pyTFM and image database organization. A: Workflow of TFM and Monolayer Stress Microscopy analysis with pyTFM. B: Organization of the pyTFM *ClickPoints* database. Input images are colored in orange, intermediary results in yellow, and the final output in the form of scalar measures in green. The mask that defines the cell boundaries and the area over which strain energy, contractility and monolayer stresses are computed is colored light blue.

Cellular force generation is quantified by the total force generation and centripetal contractility. Total force generation in turn is described by the strain energy that is elastically stored in the substrate, and centripetal contractility is described by the sum of all cell-generated forces projected towards a single force epicenter. Stresses are quantified by average normal and shear stresses and their coefficient of variation, which is a measure for stress fluctuations. Cell-cell force transduction is quantified by the line tension, which is the force per unit length acting on a segment of a cell-cell boundary. Specifically, pyTFM calculates the average magnitude of the line tension as well as the average normal and shear component of the line tension. Additionally, pyTFM calculates the area and number of cells of each cell patch, which can be used to normalize the quantities above. We provide more details on how these quantities are defined and how to interpret them in the Supplementary S1 File.

The user is required to select an area of the traction field over which the strain energy, contractility and monolayer stresses are computed. This area should cover all cell-generated tractions and is thus typically larger than the cell area. However, a significant further extension of the user-selected area beyond the cell edge will lead to an underestimation of monolayer stresses, as will be further discussed in Section 2.2.2. Optionally, the outline of the cell or cell patch can be selected, defining the area over which average stresses and stress fluctuations are computed. Also optionally, the outline of cell-cell boundaries can be selected to calculate force transduction between cells.

pyTFM generates several output files. All fields (deformations, tractions, stresses) are saved in the form of NumPy arrays as binary .npy files and are plotted as vector fields or heat maps. The cell-cell force transduction and the strain energy density can also be plotted (see Fig. 5 for an example). The user has full control over which plots are produced. All calculated scalar results are saved in a tab-separated text file. pyTFM includes Python functions to read, compare and statistically analyze the result text files of several experiments. Alternatively, the result text files can be opened with

standard text editors or data analysis tools such as Excel.

2.1 Integration of pyTFM with *ClickPoints* databases

When using the pyTFM add-on in *ClickPoints*, input and output images are organized in a database (Fig. 1B), which allows users to efficiently navigate large data sets. The database is organized in frames and layers: Each frame represents one field of view. Initially, three layers are assigned to each frame. These layers contain images of the substrate in the tensed and relaxed state, and an image of the cells. Output plots such as the deformation field or the traction field are added as new layers in each analysis step. Additionally, each frame is associated with a mask object in the form of an integer array representing the user selected areas and cell outlines. This mask object can be drawn directly in *ClickPoints* and can be displayed in each layer of a frame.

pyTFM provides a graphical user interface for the *ClickPoints* environment, which allows the user to select input images, to set all relevant analysis parameters (e.g. the elasticity of the substrate), and to select whether the analysis should be performed on all frames or just the currently viewed frame (Fig. 2). A number of tools are provided by *ClickPoints*, e.g. to draw masks, to adjust contrast and brightness of the displayed images, to measure distances and object sizes, and to export images and video sequences.

Fig 2. User interface of pyTFM. 1: Check boxes to select specific analysis steps. 2: Selection of input images, drift correction and semi automatic segmentation of cell borders. 3: Drop-down menu to select between analysing all frames in a database or analysing only the currently viewed frame. 4: Parameters for PIV and TFM. 5: User-selected region (red outline) and cell boundaries (green) for computing tractions, stresses, contractility, strain energy and line tensions. 6: *ClickPoints* tools to select the region and the cell boundaries by drawing masks. 7: *ClickPoints* navigation bar through frames. Layers are navigate with the Page Up and Page Down keys, and frames are navigated with the left and right arrow keys. 8: *ClickPoints* panel to adjust contrast and brightness of the image display. This is helpful for manually segmenting cell borders.

2.2 Implementation of TFM and Monolayer Stress Microscopy

2.2.1 Deformation fields and TFM

Deformation fields are calculated from the images of the substrate in a tensed and relaxed state using the cross correlation-based Particle Image Velocimetry (PIV) algorithm implemented in the openPIV Python package [9]. PIV is performed by selecting for example a 50x50 pixel tile around a given pixel from the tensed image and shifting the tile by pixel increments in all directions across the corresponding tile in the relaxed image. This yields a correlation matrix of in this case 99x99 pixels. The deformation vector is then obtained by calculating the vector between the position of the highest correlation and the center of the matrix. The initial deformation vector is further refined to sub-pixel accurate values by fitting a 2D Gauss curve to the directly neighbouring correlation values. To reduce noise, deformation vectors with a signal-to-noise ratio smaller than 1.03 are exclude and replaced by the local mean of the surrounding deformations at distances <= 2 pixel. The signal-to-noise ratio of each deformation vector is defined as the ratio of the correlation of the highest peak and the correlation of the second-highest peak outside of a neighborhood of 2 pixels around the highest peak. The user may also correct a drift between the two input images: The drift is identified by cross-correlating the entire images and then corrected by cropping both images to the common field of view.

Tractions are calculated with the Fourier Transform Traction Cytometry (FTTC) method [13]. Deformations ($\vec{u}$) and tractions ($\vec{t}$) are related by the convolution of the traction vector field and a Greens tensor K:

$$\vec{u} = K \otimes \vec{t} \tag{1}$$

In the case of a linearly elastic semi-infinite substrate, K is given by the Boussinesq equations [22]. Inverting Eq. 1 and solving for the tractions is difficult in real space. However, by exploiting the convolution theorem, the equation simplifies to a multiplication in Fourier space:

$$\widetilde{\vec{u}}(\vec{k}) = \widetilde{K}(\vec{k})\widetilde{\vec{T}}(\vec{k}) \tag{2}$$

where $\widetilde{\vec{u}}(\vec{k})$, $\widetilde{\vec{T}}(\vec{k})$ and $\widetilde{K}(\vec{k})$ are the Fourier transforms of the deformation field, the traction field and the Greens tensor. The latter can be found in [13].

Eq. 2 can be analytically solved and thus allows for the direct calculation of tractions in Fourier space. Tractions in real space are then obtained by applying the inverse Fourier transform and an additional Gaussian filter with a sigma of typically 1-3 µm.

The original TFM algorithm assumes that the underlying substrate is infinitely thick, which is justified in the case of single cells with dimensions that are smaller than the thickness of the elastic substrate. In the case of cell patches, however, this assumption is inadequate. We have therefore included a correction term for finite substrate thickness [23].

2.2.2 Monolayer Stress Microscopy

Stresses in a cell sheet are calculated with an implementation of Monolayer Stress Microscopy as described in [16, 17]. For computing stresses in small cell patches or single cells, we implemented a method that corrects for the limited spatial resolution of unconstrained TFM, which otherwise would lead to a substantial underestimation of stresses [20]. Details of this correction are described below.

In the absence of inertial forces, tractions and stresses are balanced according to the relation:

$$\begin{aligned} -t_x &= \frac{\delta \sigma_{xx}}{\delta x} + \frac{\delta \sigma_{yx}}{\delta y} \\ -t_y &= \frac{\delta \sigma_{yx}}{\delta x} + \frac{\delta \sigma_{yy}}{\delta y} \end{aligned} \tag{3}$$

where σ_{xx}, σ_{yy} are the normal stresses in x- and y- direction, σ_{yx} is the shear stress, and t_x and t_y are the x- and y-components of the traction vector. This differential equation is solved using a Finite Element method (FEM) where the cell patch is modeled as a 2-dimensional network of nodes arranged in a grid of quadrilateral elements. Each node in the FEM-grid is loaded with a force of the same magnitude but opposing direction as the local tractions. In the standard FE method, the nodal displacements $\vec{d}$ of the cell patch are calculated by solving the equation

$$\vec{d} = K^{-1}\vec{f} \tag{4}$$

where $\vec{f}$ are the vector of nodal forces, and K^{-1} is the inverse stiffness matrix. The nodal displacements are converted to strains by taking the derivative in x- and y-direction. Then, the strain is used to calculate the stress from the stress-strain relationship of a linearly elastic 2-dimensional material:

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{pmatrix} = \frac{E}{1-v^2} \begin{pmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & 1-v \end{pmatrix} \begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{12} \end{pmatrix} \tag{5}$$

where E and v are Young's modulus and Poisson's ratio of the material, and ϵ_{11}, ϵ_{22} and ϵ_{12} are the components of the strain tensor. Most of the FEM calculation is performed using the *solidspy* Python package [24].

The stiffness matrix K in Eq. 4 depends on the Young's modulus in such a way that the Young's modulus in Eq. 5 cancels out. The traction-stress relation is therefore independent of the Young's modulus of the cell patch [17]. Furthermore, the Poisson's ratio has only a negligible influence on the stress prediction [17]. In the pyTFM algorithm, the Young's modulus is set to 1 Pa, and the Poisson's ratio is set to 0.5.

Eq. 4 is only uniquely solvable if the displacements of at least two nodes of the FEM-grid are assigned (which constrains the solution regarding translation and rotation). In the original Monolayer Stress Microscopy algorithm [17], nodes at the edge of the field of view are constrained to zero displacements in the direction perpendicular to the edge of the field of view. This results in erroneous stresses within a margin of approximately 150 µm to the image edge, which must be excluded from further analysis [17]. This is impractical in the case of small cell patches or single cells.

pyTFM addresses this problem by modifying Eq. 4 so that it can be solved without assigning the displacements of boundary nodes. This requires two steps. First, to ensure that all forces and torques of the cell or cell patch are balanced, the forces applied to the FEM-grid are corrected by subtracting the net force and rotating all force vectors to enforce zero torque. Second, equation 4 is constrained to zero force and torque by adding the equations:

$$\begin{aligned} \sum(f_x) &= 0 \\ \sum(f_y) &= 0 \\ \sum(f_x r_y - f_y r_x) &= 0 \end{aligned} \tag{6}$$

r_x and r_y are the components of the distance vector of the corresponding node to the center of the FEM-grid. Eqs. 6 are equivalent to imposing zero translation and zero rotation constraints. The combined system of Eqs. 6 and Eq. 4 is solved numerically using a standard least-squares minimization.

The analysis of stresses in small cell patches poses a second challenge: The FEM-grid should be of the same size and shape as the cell patch, as outside nodes add additional stiffness, leading to an underestimation of the stress field. However, the limited spatial resolution of both PIV and TFM implies that some forces generated close to the edge of the cell patch are predicted to originate from outside the cell patch (Fig. 4). Neglecting these forces would lead to an underestimation of the stress field. This can be avoided by extending the FEM-grid by a small margin so that all cell-generated forces are included in the analysis. In practice, the user outlines the area with clearly visible tractions (red outline in Fig. 2), over which pyTFM then spans the FEM-grid. We explain further details of this approach in Section 3.1.1.

2.2.3 Limits of applicability of Monolayer Stress Microscopy and TFM

The TFM and Monolayer Stress Microscopy algorithms can only be applied if a number of conditions are met. 2-dimensional TFM relies on the assumption that tractions in z-direction generate only small deformations in the x- and y-plane. This is valid if z-tractions are small, or if the substrate is almost incompressible (Poison's ratio close to 0.5) [11] Additionally, TFM assumes that the matrix is a linearly elastic material. Both assumptions are valid for polyacrylamide and PDMS, two popular substrates for TFM [25–28].

For Monolayer Stress Microscopy, cells are modeled as a linearly elastic material with uniform elastic properties. As local stiffness inhomogeneities introduce only negligible errors in the stress prediction, it is generally not necessary to consider non-linear elastic effects of the cells [17]. Furthermore, Monolayer Stress Microscopy assumes that the cell

dimensions in the x- and y-plane (length l) is larger the cell height (h). Increased cell height introduces an error in the stress prediction on the order of $(l/h)^2$ [17].

3 Results

3.1 Accuracy of TFM and MSM algorithms

To evaluate the accuracy of the calculated tractions and stresses, we designed a simple test system with a predefined stress field for which tractions and deformations can be analytically computed. We then compare the analytical solution to the solution provided by pyTFM.

The workflow of this test is illustrated in Fig. 3A: First, we define a square-shaped area of 150 µm width representing a cell patch. This area carries a uniform normal stress in x- and y-direction of 1 N/µm magnitude and zero shear stress. Stresses outside the cell patch are set to zero. Next, we calculate the corresponding traction field by taking the spatial derivatives of the stress field and applying Eq. 3.

Fig 3. Accuracy of stress and traction force calculation. A: We model a cell colony as a uniformly distributed square-shaped stress field for which we analytically compute a traction field and subsequently a deformation field. We use the deformation field as the input for Traction Force Microscopy and Monolayer Stress Microscopy to recover the traction and the stress fields. B: Input and reconstructed traction field. C: Input and reconstructed stress field. The yellow dashed line shows the extent of the original stress field. D: Contractility and average normal and shear stress and CV for the mean normal stress in the input and reconstructed traction and stress fields. The contractility is computed over an area that is 12 µm larger than the original stress field. Average normal and shear stresses and the CV of the mean normal stress are computed over the area of the original stress field.

From the traction field, we obtain the deformation field by first calculating the Fourier transform of the traction field. Then we use Eq. 2 to obtain the deformation field in Fourier space and, after applying the inverse Fourier transform, in real space. We use a modified Greens Tensor K to account for a finite substrate thickness [23]. The substrate thickness is set to 100 µm.

The deformation field is then used as the input for the TFM and Monolayer Stress Microscopy algorithms. We use an FEM-grid area that is 5 µm larger than the original stress field area since this resulted in the best stress recovery (Fig. 4A).

The computed mean of the normal and shear stresses and the standard deviation of

the normal stresses are finally compared with the known input stress (uniform normal stress in x- and y-direction of 1 N/µm magnitude, and zero shear stress). To compare the reconstructed traction field with the analytical solution, we also compute the total contractility (sum of all cell-generated forces projected towards a single force epicenter) over the FEM-grid area.

We find that the pyTFM algorithm accurately reconstructs the stress field (Fig. 3B). By contrast, the reconstructed traction field is blurred in comparison to the input traction field (Fig. 3C). This is the effect of a Gaussian smoothing filter with a sigma of 3 µm that is applied to the tractions computed by the FTTC algorithm. This filter helps to prevent unphysiological isolated and locally diverging tractions in the case of a noisy input deformation field. In our test case, we do not model the influence of noise and could therefore omit the filter; in practical applications, we find a sigma of 3 µm to give the best compromise between resolution and noise.

The computed average normal stress is slightly (7%) smaller than the input stress, but the error increases rapidly when the margin for extending the FEM-grid is decreased below 5 µm (Fig. 4A). Total contractility and the coefficient of variation for the normal stress are recovered accurately (Fig. 3D).

3.1.1 Effect of FEM-grid size on the stress recovery

pyTFM requires the user to select an area of the traction field over which pyTFM then computes contractility and strain energy and draws the FEM-grid for computing monolayer stresses. The size of this area influences the accuracy of the stress and force measurements. Selecting an area that is too small leads to an underestimation of stresses and contractility. Selecting an area that is too large also leads to an underestimation of stresses. To systematically analyze which effect the size of the user-selected area has on the traction and stress reconstruction, we expand the traction area and analyze the average normal stress and the contractility for the synthetic test data described above (Fig. 4A) and for a MDCK cell patch grown on a polyacrylamide substrate (Young's modulus 49 kPa, Fig. 4B). In the case of the synthetic data, we normalize the computed average normal stress and contractility to the known input stress (1 N/m) and to the known contractility of the input traction field (600 N), respectively. In the case of the experimental data, we normalize the computed average normal stress and contractility

to their respective maximum values as the true stress and contractility is unknown.

Fig 4. Effect of increasing the traction area on stress and contractility recovery. The predicted traction fields of an artificial test system (A) and a real MDCK cell patch (B). The outlines of 3 representative FEM-grids are shown on the left. The relationship between average normal stress and FEM-grid size is shown on the right.

We find that the normalized stress rapidly increases (by approximately 40%) with increasing area until it reaches a maximum, after which it declines at a slower rate. The contractility displays a similar initial increase but then remains approximately constant. The maximum of the normalized stress occurs when the traction area just covers all cell-generated tractions, including those that appear outside the cell patch. In the cases of the synthetic data, the maximum is reached at a traction area expansion distance of 5 µm beyond the cell patch outline, whereas in the case of the MDCK cell patch, it is reached at at an expansion distance of 20 µm. The reason for this larger distance in the MDCK data is the additional blurring of tractions introduced by the PIV algorithm (whereas no PIV was needed for analyzing the synthetic data). The traction area corresponding to the maximal normal stress can be regarded as the optimum, as approximately 93% of the input stress is recovered. Expanding the traction area and thus the FEM-grid beyond the optimum distance adds elastic material to the monolayer and thereby reduces the average stress. This stress reduction, however, occurs only gradually (Fig. 4B), which implies that in practice it is best to choose the traction area rather generously to include all cell-generated tractions. The contractility reaches its maximum values at almost the same expansion distance as the stress. Thus it is possible to use the same area to accurately compute both stress and contractility.

3.2 Analysis of a MDCK cell-colony with pyTFM

In the following, we illustrate the workflow of pyTFM (Fig. 1) using a MDCK cell colony as a representative example. Experimental details for this example are provided in Supplementary S2 File. Two images of fluorescent beads serve as the essential input, one image taken before and one image after cell removal by trypsinization of the cells (Fig. 5A). pyTFM calculates the deformation field (Fig. 5B) and the traction field (Fig. 5C). The user then selects the area (red outline in Fig. 5) over which pyTFM draws the FEM-grid and computes the contractility and strain energy (both are scalar values),

and the monolayer stress field (represented as a map of normal stresses (Fig. 5E)). If the user optionally selects the outline of the cell patch and the boundaries of the individual cells within the patch (green outlines in Fig. 5E), pyTFM also computes the line tension between the cells (Fig. 5F). The program also computes a number of scalar values for quantifying cellular force generation and stress distribution (Table 1).

Fig 5. Analysis of stress and force generation of a MDCK cell colony. A: Images of substrate-embedded fluorescent beads before and after the cells are detached by trypsinization. B: Substrate deformation field. C: Traction field. The user selects the area (red outline) over which contractility, strain energy and cell stresses are subsequently calculated. D: Image of the cell colony; fluorescent membrane staining with tdTomato-Farnesyl. E: Absolute value of the Mean normal stress in the cell colony. F: Line tension along cell-cell borders. The orange dashed line marks the outer edge of the cell colony.

The cell colony in this example displays several typical features: First, stresses and traction forces are unevenly distributed across the cell colony, as indicated for example by a high coefficient of variation of 0.38 for the normal component of the stress field (Table 1). Second, the average line tension is higher than the average normal or maximum shear stress. This indicates that, on the average, interfacial stresses between cells exceed intracellular stresses. Third, normal and tensile components of the stress field dominate over shear stress components, indicating that tractions are locally aligned. In addition, the shear component of the line tension is considerably smaller than its normal component, implying that cells in this colony pull on each other but do not exert appreciable forces parallel to their boundaries.

Table 1. Scalar values computed by pyTFM quantifying cellular force generation and stress distribution.

Scalar Quantity	Result
Contractility	0.64 µN
Strain energy	0.11 pJ
Avg. max. normal stress	2.62 mN/m
Avg. max. shear stress	0.78 mN/m
CV normal stress	0.38
Avg. line tension	2.04 mN/m
Avg. normal line tension	1.94 mN/m
Avg. shear line tension	0.56 mN/m

4 Availability and future directions

Currently, pyTFM exclusively uses the Fourier Transform Traction Cytometry algorithm [13]. This algorithm is simple, robust and well established but has a number of limitations (see Section 2.2.3). However, due to the structure of pyTFM, it is possible to implement alternative algorithms that address these issues with minimal changes to other parts of the software. An example is the Boundary Elements Method [11] that solves the inverse problem numerically in real space and allows users to set spatial constraints on the tractions. This avoids the occurrence of arguably unphysiological tractions outside the cell area. Another example is 2.5-dimensional Traction Force Microscopy that allows for the calculation of tractions in z-directions [10]. This algorithm is also necessary when cells are grown on compressible substrates and generate significant z-tractions. Finally, FEM-based Traction Force Microscopy algorithms allow for the analysis of cells grown on non-linear elastic substrates such as collagen [29].

pyTFM can be downloaded and installed from `https://github.com/fabrylab/pyTFM` under the GNU General Public License v3.0. Detailed instructions on the installation and usage are provided at `https://pytfm.readthedocs.io/`.

Conflict of interest

The authors declare not conflict of interest.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (SFB-TRR 225, project number 326998133, subproject A01, and FA 336/11-1), the Ministry of Health of the Czech Republic (grant 17-31538A) and The European Cooperation in Science and Technology (COST) grant CA15214-EuroCellNet (MEYS CR LTC17063).

5 Supporting information

S1 File. Scalar quantities used to describe cellular stresses and force generation. We discuss the definition and interpretation of the quantities that pyTFM uses to describe cellular stresses, force generation and cell-cell force transfer.

S2 File. Experimental details for analyzing the MDCK cell colony. We provide basic information on our protocols for polyacrylamide gel preparation and cell culture for the TFM analysis of the MDCK cell colony.

S3 File. pyTFM source code and documentation. This archive contains the pyTFM source code and documentation which includes installation and usage instructions and links to further example data sets.

References

1. Pelham RJ, l Wang Y. Cell locomotion and focal adhesions are regulated by substrate flexibility. Proceedings of the National Academy of Sciences. 1997;94(25):13661–13665. doi:10.1073/pnas.94.25.13661.

2. Engler AJ, Sen S, Sweeney HL, Discher DE. Matrix Elasticity Directs Stem Cell Lineage Specification. Cell. 2006;126(4):677–689. doi:10.1016/j.cell.2006.06.044.

3. Steinwachs J, Metzner C, Skodzek K, Lang N, Thievessen I, Mark C, et al. Three-dimensional force microscopy of cells in biopolymer networks. Nature Methods. 2015;13(2):171–176. doi:10.1038/nmeth.3685.

4. REHFELDT F, ENGLER A, ECKHARDT A, AHMED F, DISCHER D. Cell responses to the mechanochemical microenvironment Implications for regenerative medicine and drug delivery. Advanced Drug Delivery Reviews. 2007;59(13):1329–1339. doi:10.1016/j.addr.2007.08.007.

5. Pascalis CD, Pérez-González C, Seetharaman S, Boëda B, Vianay B, Burute M, et al. Intermediate filaments control collective migration by restricting traction forces and sustaining cell–cell contacts. Journal of Cell Biology. 2018;217(9):3031–3044. doi:10.1083/jcb.201801162.

6. Mammoto T, Ingber DE. Mechanical control of tissue and organ development. Development. 2010;137(9):1407–1420. doi:10.1242/dev.024166.

7. Mark C, Grundy TJ, Strissel PL, Böhringer D, Grummel N, Gerum R, et al. Collective forces of tumor spheroids in three-dimensional biopolymer networks. eLife. 2020;9. doi:10.7554/elife.51912.

8. Peschetola V, Laurent VM, Duperray A, Michel R, Ambrosi D, Preziosi L, et al. Time-dependent traction force microscopy for cancer cells as a measure of invasiveness. Cytoskeleton. 2013;70(4):201–214. doi:10.1002/cm.21100.

9. Liberzon A, Lasagna D, Aubert M, Bachant P, Jakirkham, Ranleu, et al.. OpenPIV/openpiv-python: fixed windows conda-forge failure with encoding; 2019. Available from: `https://zenodo.org/record/3566451`.

10. Maskarinec SA, Franck C, Tirrell DA, Ravichandran G. Quantifying cellular traction forces in three dimensions. Proceedings of the National Academy of Sciences. 2009;106(52):22108–22113. doi:10.1073/pnas.0904565106.

11. Dembo M, Wang YL. Stresses at the Cell-to-Substrate Interface during Locomotion of Fibroblasts. Biophysical Journal. 1999;76(4):2307–2316. doi:10.1016/s0006-3495(99)77386-8.

12. Huang Y, Schell C, Huber TB, Şimşek AN, Hersch N, Merkel R, et al. Traction force microscopy with optimized regularization and automated Bayesian parameter selection for comparing cells. Scientific Reports. 2019;9(1). doi:10.1038/s41598-018-36896-x.

13. Butler JP, Tolić-Nørrelykke IM, Fabry B, Fredberg JJ. Traction fields, moments, and strain energy that cells exert on their surroundings. American Journal of Physiology-Cell Physiology. 2002;282(3):C595–C605. doi:10.1152/ajpcell.00270.2001.

14. Yang Z, Lin JS, Chen J, Wang JHC. Determining substrate displacement and cell traction fields—a new approach. Journal of Theoretical Biology. 2006;242(3):607–616. doi:10.1016/j.jtbi.2006.05.005.

15. Sabass B, Gardel ML, Waterman CM, Schwarz US. High Resolution Traction Force Microscopy Based on Experimental and Computational Advances. Biophysical Journal. 2008;94(1):207–220. doi:10.1529/biophysj.107.113670.

16. Tambe DT, Hardin CC, Angelini TE, Rajendran K, Park CY, Serra-Picamal X, et al. Collective cell guidance by cooperative intercellular forces. Nat Mater. 2011;10(6):469–475. doi:10.1038/nmat3025.

17. Tambe DT, Croutelle U, Trepat X, Park CY, Kim JH, Millet E, et al. Monolayer Stress Microscopy: Limitations, Artifacts, and Accuracy of Recovered Intercellular Stresses. PLoS ONE. 2013;8(2):e55172. doi:10.1371/journal.pone.0055172.

18. Tambe DT, Butler JP, Fredberg JJ. Comment on "Intracellular stresses in patterned cell assemblies" by M. Moussus et al., Soft Matter, 2014, 10, 2414. Soft Matter. 2014;10(39):7681–7682. doi:10.1039/c4sm00597j.

19. Moussus M, der Loughian C, Fuard D, Courçon M, Gulino-Debrac D, Delanoë-Ayari H, et al. Intracellular stresses in patterned cell assemblies. Soft Matter. 2013;10(14):2414–2423. doi:10.1039/c3sm52318g.

20. Ng MR, Besser A, Brugge JS, Danuser G. Mapping the dynamics of force transduction at cell–cell junctions of epithelial clusters. eLife. 2014;3. doi:10.7554/elife.03282.

21. Gerum RC, Richter S, Fabry B, Zitterbart DP. ClickPoints : an expandable toolbox for scientific image annotation and analysis. Methods in Ecology and Evolution. 2016;8(6):750–756. doi:10.1111/2041-210x.12702.

22. Landau LD. Theory of elasticity. Oxford England Burlington, MA: Butterworth-Heinemann; 1986.

23. Trepat X, Wasserman MR, Angelini TE, Millet E, Weitz DA, Butler JP, et al. Physical forces during collective cell migration. Nature Physics. 2009;5(6):426–430. doi:10.1038/nphys1269.

24. Gómez J, Guarín-Zapata N. SolidsPy: 2D-Finite Element Analysis with Python; 2018. Available from: https://github.com/AppliedMechanics-EAFIT/SolidsPy.

25. Lee D, Zhang H, Ryu S. In: Mondal MIH, editor. Elastic Modulus Measurement of Hydrogels. Cham: Springer International Publishing; 2018. p. 1–21. Available from: https://doi.org/10.1007/978-3-319-76573-0_60-1.

26. Pritchard RH, Lava P, Debruyne D, Terentjev EM. Precise determination of the Poisson ratio in soft materials with 2D digital image correlation. Soft Matter. 2013;9(26):6037. doi:10.1039/c3sm50901j.

27. Kim TK, Kim JK, Jeong OC. Measurement of nonlinear mechanical properties of PDMS elastomer. Microelectronic Engineering. 2011;88(8):1982–1985. doi:10.1016/j.mee.2010.12.108.

28. Boudou T, Ohayon J, Picart C, Pettigrew RI, Tracqui P. Nonlinear elastic properties of polyacrylamide gels: Implications for quantification of cellular forces. Biorheology. 2009;46(3):191–205. doi:10.3233/BIR-2009-0540.

29. Zielinski R, Mihai C, Kniss D, Ghadiali SN. Finite Element Analysis of Traction Force Microscopy: Influence of Cell Mechanics, Adhesion, and Morphology. Journal of Biomechanical Engineering. 2013;135(7). doi:10.1115/1.4024467.

A

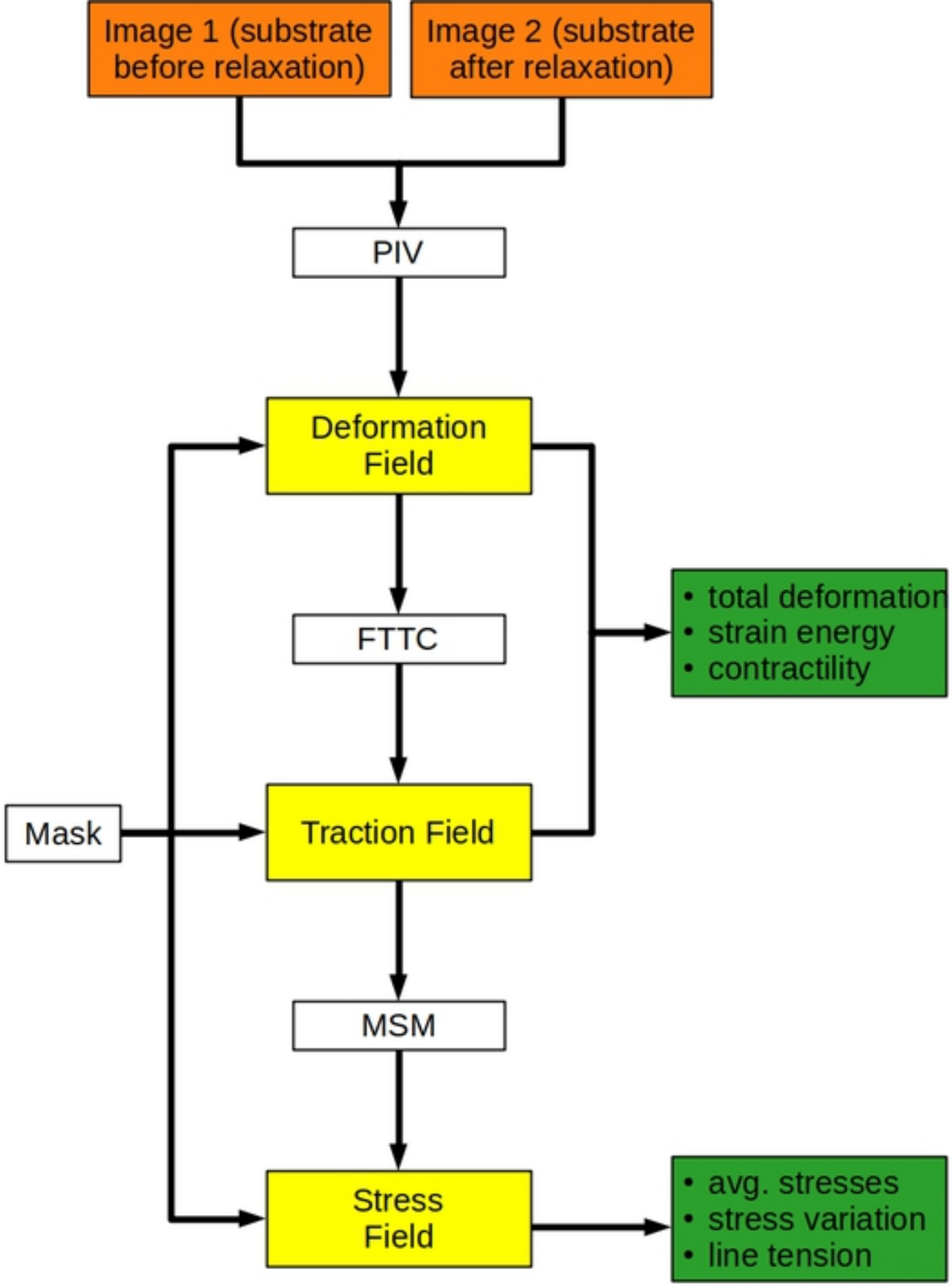


Image 1 (substrate before relaxation)
Image 2 (substrate after relaxation)
PIV
Deformation Field
FTTC
Traction Field
MSM
Stress Field
Mask
• total deformation
• strain energy
• contractility
• avg. stresses
• stress variation
• line tension

B

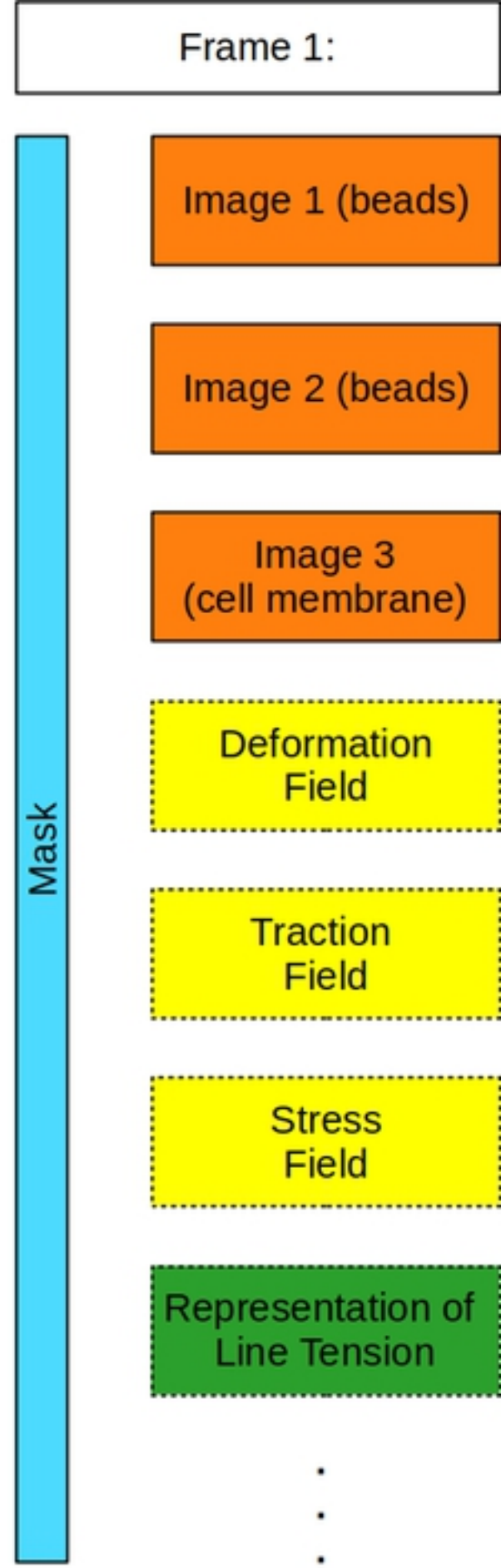

Frame 1:
Image 1 (beads)
Image 2 (beads)
Image 3 (cell membrane)
Deformation Field
Traction Field
Stress Field
Representation of Line Tension
Mask

Fig1

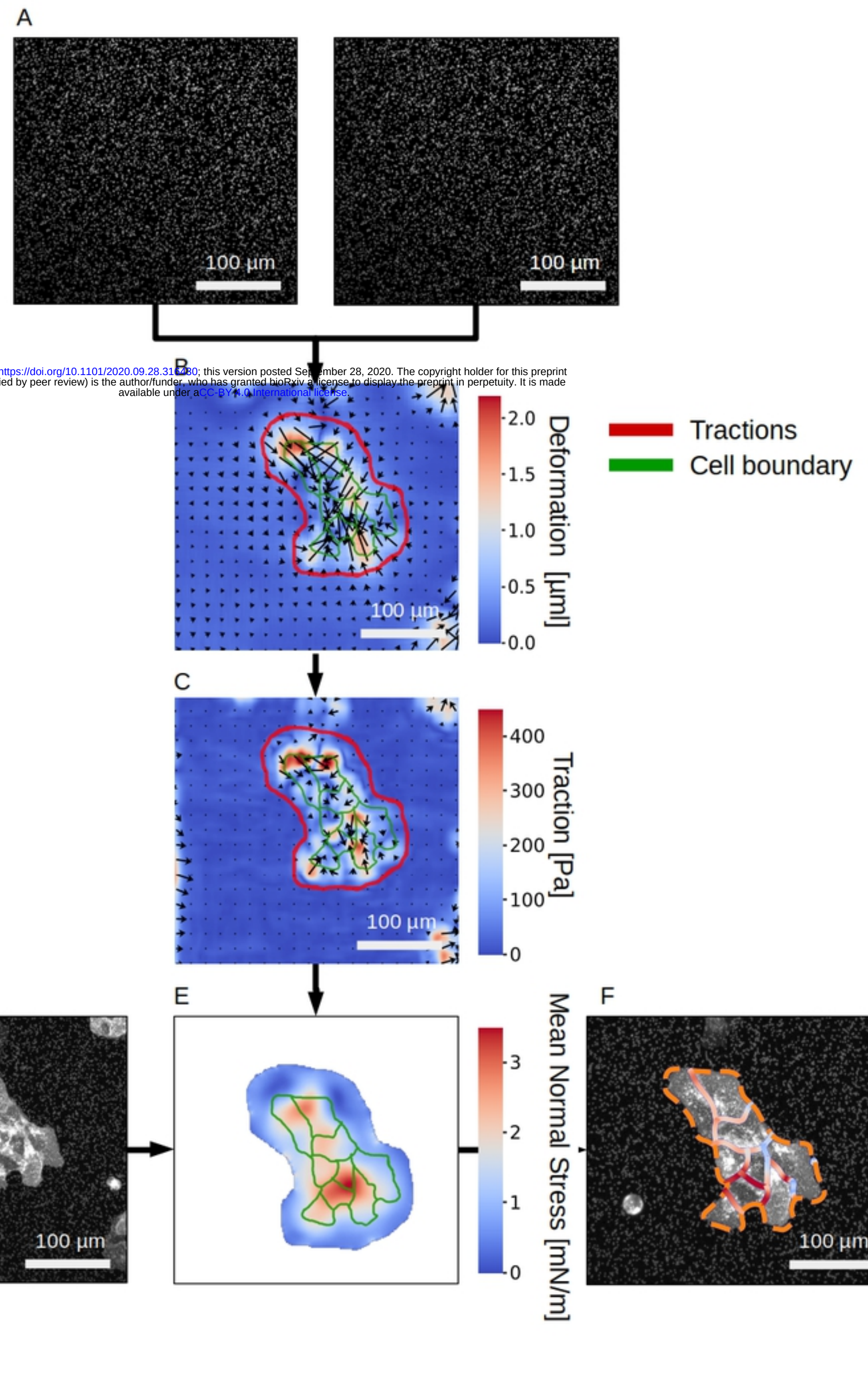


Fig5

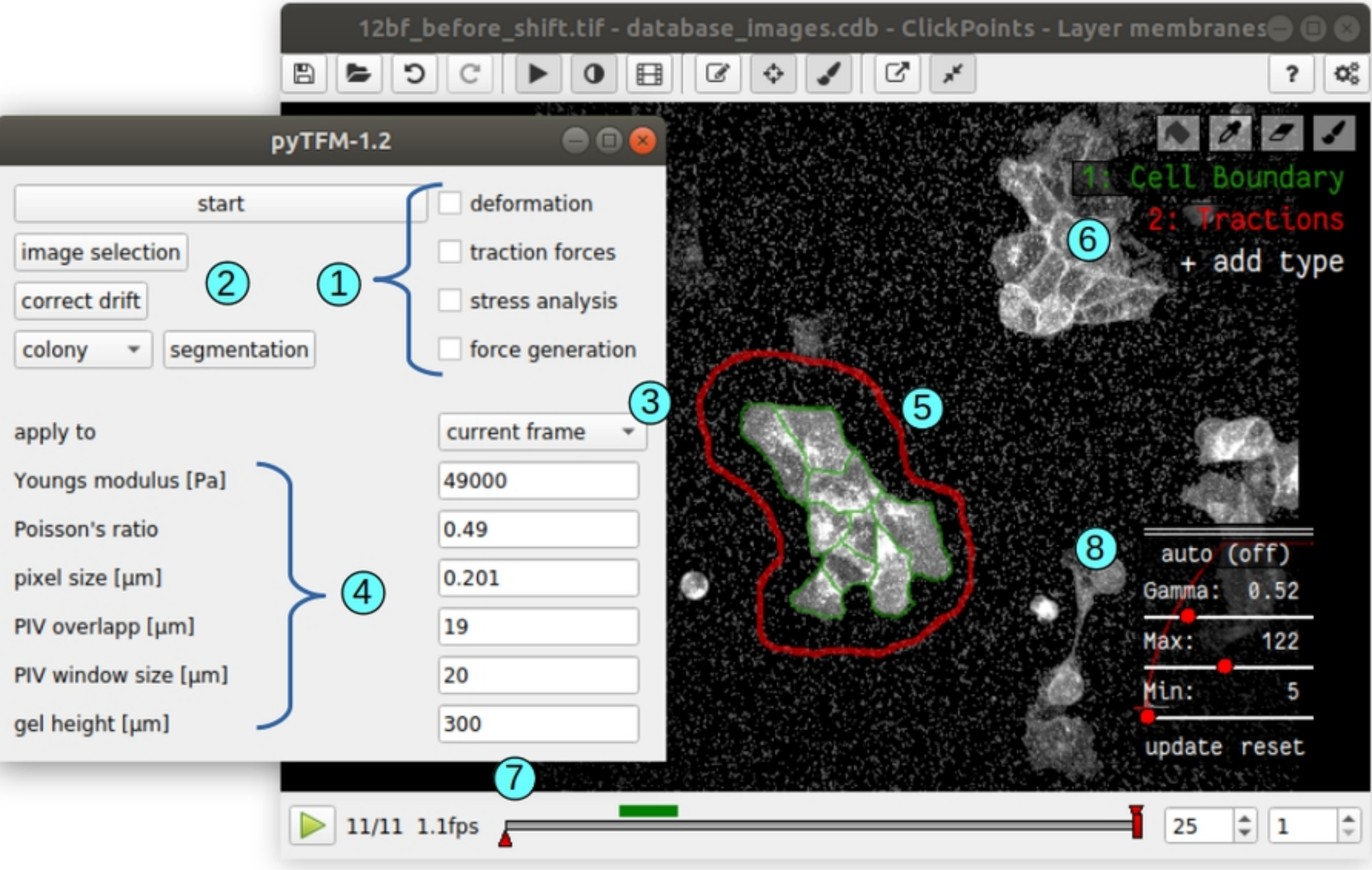

12bf_before_shift.tif - database_images.cdb - ClickPoints - Layer membranes
pyTFM-1.2
start
image selection
correct drift
colony
segmentation
deformation
traction forces
stress analysis
force generation
apply to
current frame
Youngs modulus [Pa]
49000
Poisson's ratio
0.49
pixel size [µm]
0.201
PIV overlapp [µm]
19
PIV window size [µm]
20
gel height [µm]
300
1
2
3
4
5
6
7
8
1: Cell Boundary
2: Tractions
+ add type
auto (off)
Gamma: 0.52
Max: 122
Min: 5
update reset
11/11 1.1fps
25
1

Fig2

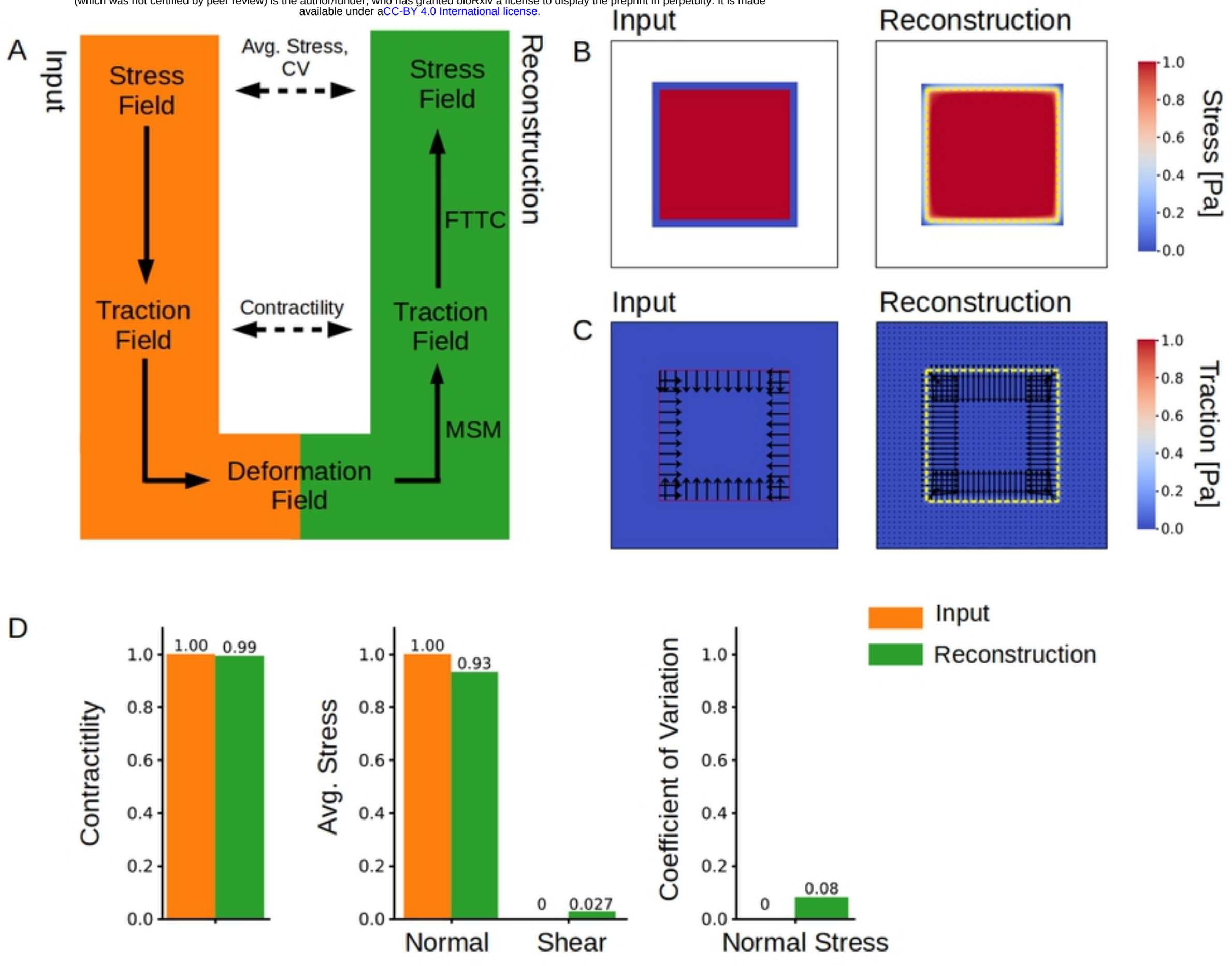
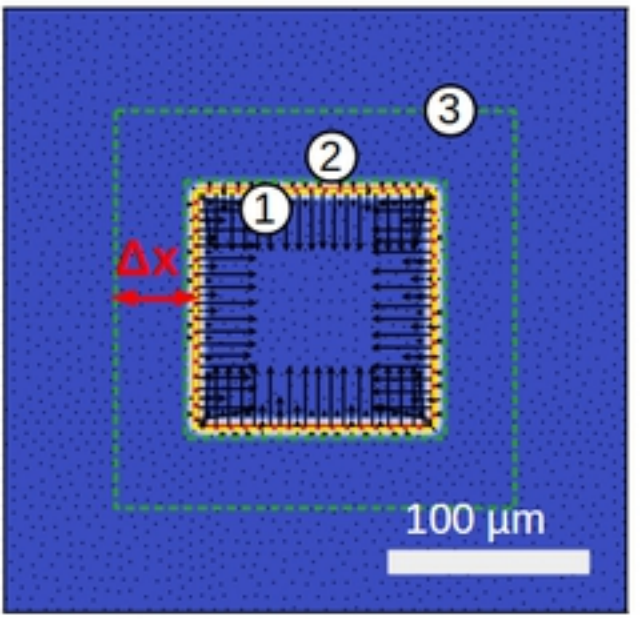
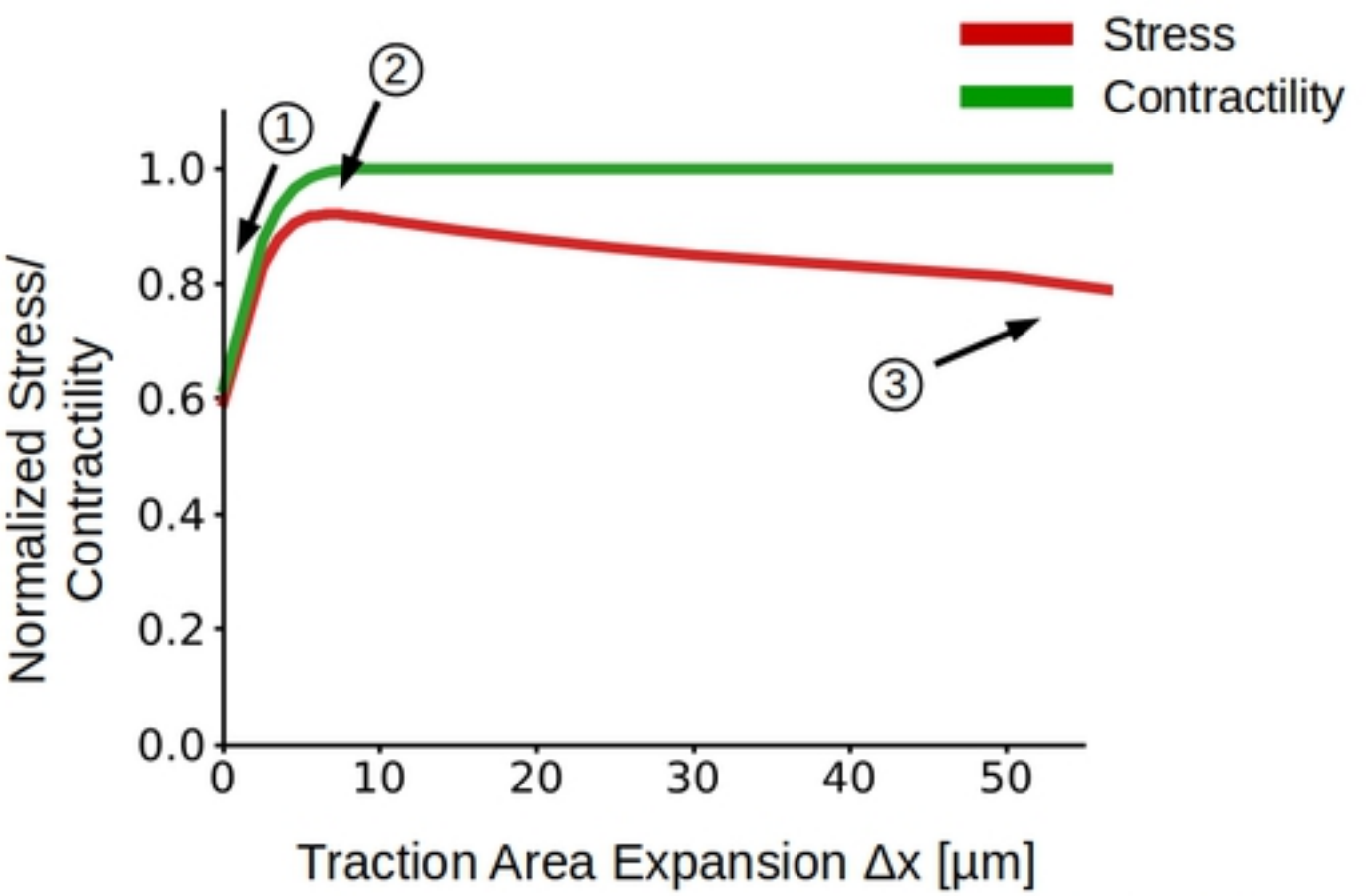


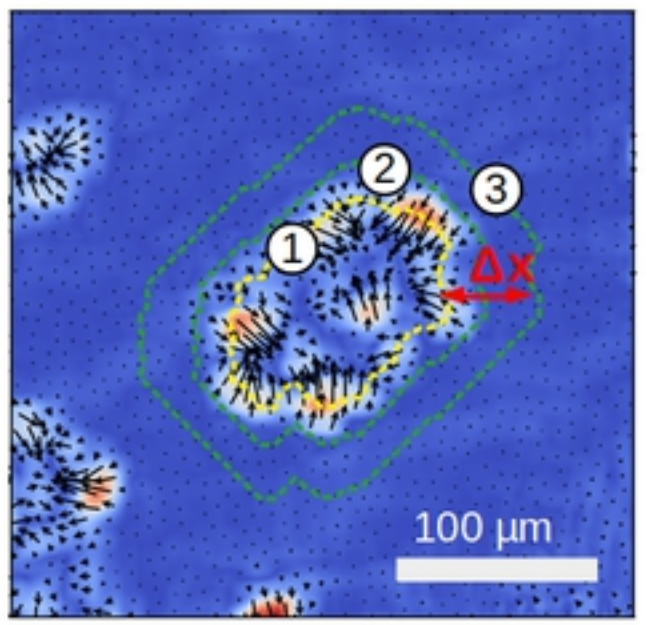
Fig1

A


1
2
3
Δx
100 μm


Stress
Contractility
1
2
3
Normalized Stress/
Contractility
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
Traction Area Expansion Δx [μm]

B


1
2
3
Δx
100 μm

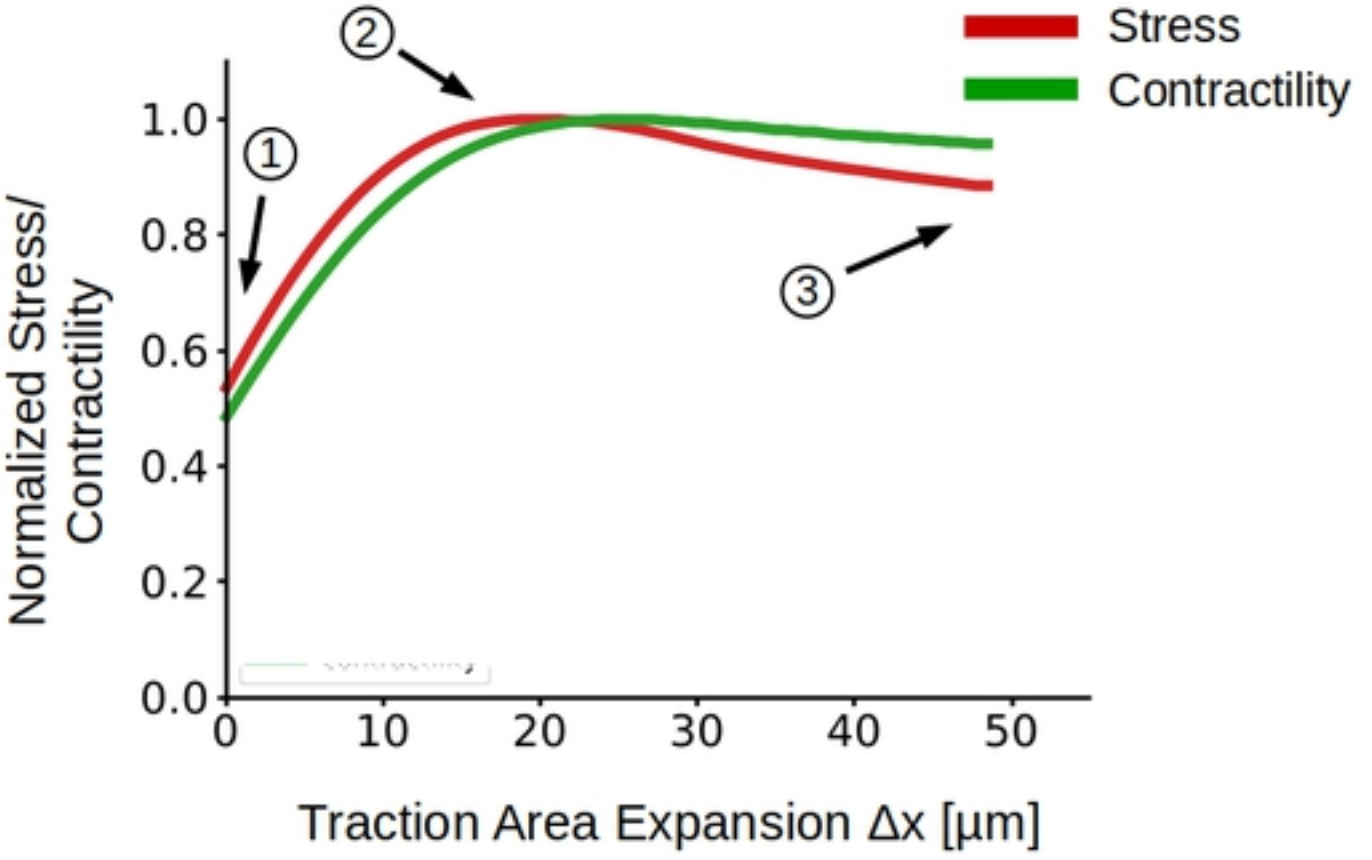

Stress
Contractility
1
2
3
Normalized Stress/
Contractility
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
Traction Area Expansion Δx [μm]

Fig4