Original software publication

TFMLAB: A MATLAB toolbox for 4D traction force microscopy

Jorge Barrasa-Fano [a], Apeksha Shapeti [a], Álvaro Jorge-Peñas [a], Mojtaba Barzegari [a], José Antonio Sanz-Herrera [b], Hans Van Oosterwyck [a,c,*]

[a] *Department of Mechanical Engineering, Biomechanics section, KU Leuven, Belgium*
[b] *Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Spain*
[c] *Prometheus, Division of Skeletal Tissue Engineering, KU Leuven, Leuven, Belgium*

ARTICLE INFO

ABSTRACT

We present TFMLAB, a MATLAB software package for 4D (x;y;z;t) Traction Force Microscopy (TFM). While various TFM computational workflows are available in the literature, open-source programs that are easy to use by researchers with limited technical experience and that can analyze 4D in vitro systems do not exist. TFMLAB integrates all the computational steps to compute active cellular forces from confocal microscopy images, including image processing, cell segmentation, image alignment, matrix displacement measurement and force recovery. Moreover, TFMLAB eases usability by means of interactive graphical user interfaces. This work describes the package's functionalities and analyzes its performance on a real TFM case.

Code metadata

Current code version	*v1.0*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-20-00104
Legal Code License	*LGPL v3.0*
Code versioning system used	*Gitlab*
Software code languages, tools, and services used	*MATLAB*
Compilation requirements, operating environments & dependencies	*Image processing toolbox, signal processing toolbox (MATLAB), DIPImage library, Elastix*
If available Link to developer documentation/manual	https://gitlab.kuleuven.be/MAtrix/Jorge/tfmlab_public/-/blob/public_tfmlab/manual.pdf
Support email for questions	jorge.barrasafano@kuleuven.be

1. Motivation and significance

Mechanobiology, the study of interactions between mechanical signals and biological responses, has substantially grown over the last 20 years. The importance of mechanical forces in driving cell behavior is now widely recognized in the literature [1–4]. Meanwhile, Traction Force Microscopy (TFM) has become the preferred methodology to quantify forces at the cell–matrix interface. Typically, synthetic or natural hydrogels mixed with cells and fiducial markers are imaged by means of optical microscopy before (stressed state) and after (relaxed state) cell relaxation. Image processing algorithms measure cellular force-induced deformations on the extracellular matrix (ECM) by tracking the movement of fiducial markers between these two states. Finally, cell forces are retrieved by applying mechanical models that relate forces to deformations. The reader is referred to excellent reviews in the field for more details [3,5,6].

TFM has traditionally been applied to study cell mechanics in 2D in vitro cultures where cells are seeded on top of a substrate [7–9]. Several open-source MATLAB software packages used in 2D (i.e. 2D displacement and force calculation for 2D cell culture) and 2.5D TFM (i.e. 3D displacement and force calculation for 2D cell culture) are available: displacement measurement algorithms such as Particle Image Velocimetry (PIV) [10] and Particle Tracking (PT) [11], or regularization techniques for force recovery [12]. Moreover, "all-in-one" software packages [13–16], and ImageJ plugins [17] have brought complex TFM computations closer to experimentalists. Current challenges lie in making TFM available

* Corresponding author at: Department of Mechanical Engineering, Biomechanics section, KU Leuven, Belgium.

E-mail addresses: jorge.barrasafano@kuleuven.be (Jorge Barrasa-Fano), apeksha.shapeti@kuleuven.be (Apeksha Shapeti), alvaro.jorgepenas@soundtalks.com (Álvaro Jorge-Peñas), mojtaba.barzegari@kuleuven.be (Mojtaba Barzegari), jsanz@us.es (José Antonio Sanz-Herrera), hans.vanoosterwyck@kuleuven.be (Hans Van Oosterwyck).

Jorge Barrasa-Fano, Apeksha Shapeti, Álvaro Jorge-Peñas et al.

SoftwareX 15 (2021) 100723

for physiologically more relevant experiments in 3D (cells fully embedded in a hydrogel) and incorporating time dependency (4D). Unlike for 2D TFM, the availability of 3D TFM open-source tools is scarce. Some groups that are currently publishing 3D TFM studies do not provide source codes [18–21]. Others, provide source codes but lack force calculation functionalities [22]. An exception are the open source contributions of Fabry's group. They developed *SAENO* [23], the first open source 3D TFM solver for nonlinear, fibrillary materials, like collagen and fibrin. While it remains a unique contribution to the field, we hereby address two limitations regarding its usability by researchers with limited technical experience. First, the software does not provide feedback of intermediate steps involved in the calculations (such as image filtering or shift correction), which requires higher technical expertise from users to select the optimal input parameters for each sample [24]. Second, while the signal to noise ratio of the images heavily depend, among others, on the type of microscope or the hydrogel used, *SAENO* does not allow to adjust image processing operations to new data as they are hidden from the user. More recently, they published *jointforces,* an open source Python package [25]. While they applied it for data from a 3D in vitro model of a tumor spheroid, the software only calculates 2D forces at the equatorial plane of the spheroid.

With TFMLAB, we provide an "all-in-one" open source 4D (3D + time) toolbox that includes all the necessary computational steps for TFM. Experimentalists can input the raw microscopy images and follow every computational step through interactive Graphical User Interfaces (GUIs) for parameter tuning. Users can choose different image filters to process various types of data and use the Free Form Deformation (FFD) algorithm for displacement measurement. In particular, FFD has proven to be superior to PIV and has been used in various experimental studies involving 2D, 3D and 4D (time lapse) measurements in single cell and multi-cellular systems [26–30]. Finally, accurate and computationally efficient force calculation is provided by our recently published physically-based nonlinear inverse method (PBNIM) [31,32].

TFMLAB combines complex state of the art methods into a closed and user-friendly software that does not require code manipulations and allows for applying 4D TFM in biological studies which holds strong promise for mechanobiology.

2. Software description

2.1. Software architecture

We developed TFMLAB in the MATLAB framework to provide robust and user-friendly processing of the microscopy data. The software has one main function main.m, which calls other routines at every step of computational TFM (see schematic in Fig. 1). However, the user only needs to interact with TFMLAB's GUIs. Briefly, TFMLAB has the following steps: (1) microscopy image reading, (2) image processing, (3) displacement measurement and (4) force calculation. There is a GUI for parameter tuning before every step and result folders are stored between steps. While MATLAB orchestrates the workflow, some specific operations are performed by external software (see Table 1).

2.2. Software functionalities

The user must first run the script main.m in MATLAB and the steps depicted in Fig. 1 follow, asking for user interaction through different GUIs.

2.2.1. Microscopy image reading

First, the user must provide input data. The GUI *set_inputs* will ask for an input microscopy file. For flexibility, TFMLAB uses the Bio-Formats toolbox which can read multiple microscopy file formats [35]. From the input file, the software reads the channels corresponding to the cells (e.g., imaged after transduction with LifeAct for filamentous actin staining) and the fiduciary markers such as fluorescent beads and/or fibers (e.g., in case of a fibrillar collagen hydrogel imaged by means of second harmonic generation [27]). Any other channels (e.g. bright field, nuclei…) present in the microscopy file are ignored, as they are not necessary for TFM. The user can provide any number of images (time points) for a given sample as long as the field of view of the hydrogel remains constant. All the images are considered to conform different time points of the stressed state while the last image provided is considered to be the relaxed state. Function *microscope2Tif* then converts the input data into .tiff files. This is a standard format that maintains bit depth and is easy to access with any free software such as ImageJ. The data is split by channels (cell, beads and/or fibers) and time points (tp_01, tp_02,…, tp_R) and stored in the A_rawImagesTiff output folder.

2.2.2. Image processing

The GUI *set_imProcessing* allows the user to interactively tune image filtering and enhancement parameters. TFMLAB provides specific filters for cell, bead and fiber images, and tools for cell segmentation. There are multiple filtering options combining the MATLAB image processing toolbox and the DIPImage library [36] for versatility (see more details in Appendix B and examples in Section 3). The main function then calls the functions *timeSeries-ImFilt* and *timeSeriesCellSeg* to process all the input images and stores them in the B_filteredImages output folder. Rigid image registration is done during the first shift correction step, correcting for any microscope stage related drifts. First, a phase correlation operation is done to calculate the shift between all the images and a reference image with the function *timeSeriesShift-Calc*. If only two images are provided, the reference is the relaxed image. Otherwise, the reference is the middle time-point. The bead/fiber images are used for this operation. Second, the function *timeSeriesShiftCorrection* aligns the images from all the channels according to the calculated shift, and any non-overlapping image borders are cropped such that all the images have equal size at the end of the operation. The resultant images are stored in the C_shiftCorrectedImages output folder.

2.2.3. Displacement measurement

Displacement measurement is done by means of FFD-based image registration [26]. Briefly, a multivariate B-spline function warps a moving image (a marker image at a certain time point of the stressed state, M_k) to optimize the value of a distance metric that quantifies its similarity to a fixed image (the marker image that represents the relaxed state, M_R). Using the GUI *set_dispCalc* the user can choose between different optimizers, distance metrics and can tune the spacing between the control points of the B-spline curves to adapt it to experimental conditions of each sample (see details in Appendix C). The main function then calls *timeSeriesDispCalc*, which uses the Elastix toolbox for efficient image registration as previously described [29,30,42]. The cell segmentations S_k can be used as masks to ignore beads engulfed by the cells. The parameter and performance files of the image registration are stored in output folder D_1_calcElastix. With a second shift correction, the measured displacement fields $\mathbf{u}_k^{meas}$ are then corrected for any spurious rigid shifts in *timeSeries-DispShiftCorr* and stored in folder D_2_displacements. This shift correction operation is done by subtracting the mean from each displacement field component for the entire image stack. The images (beads/fibers, cell and cell segmentations) are also corrected
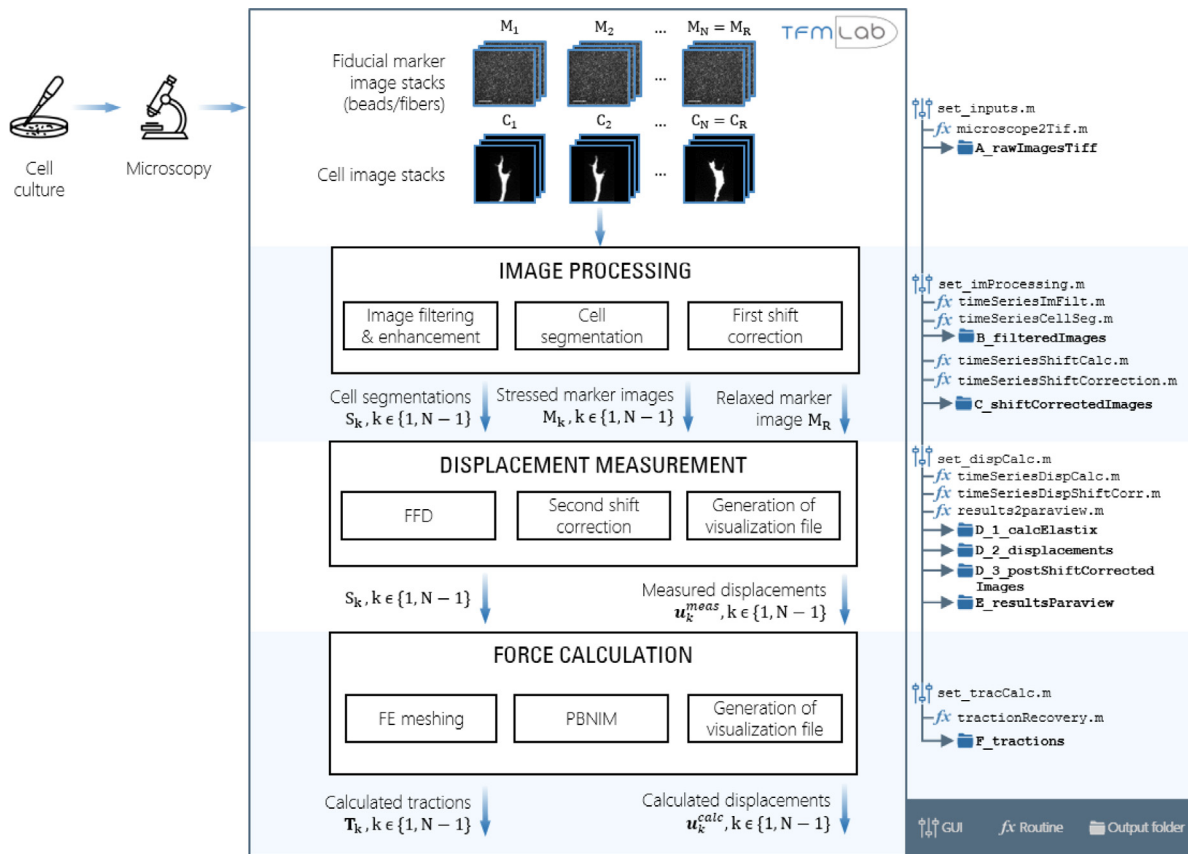
Jorge Barrasa-Fano, Apeksha Shapeti, Álvaro Jorge-Peñas et al.

SoftwareX 15 (2021) 100723

Fig. 1. Basic flow chart of TFMLAB. After cell culture and microscopy imaging (left hand side), the user can input marker (M_k) and cell (C_k) image stacks into TFMLAB (top). The main steps of the workflow (middle), the GUI scripts, the name of the key MATLAB functions that perform the steps and the output folders that the software generates (right hand side) are shown. Briefly, after image processing, displacements ($\boldsymbol{u}_k^{meas}$) in the marker images are measured by means of Free Form Deformation (FFD). These displacements and a finite element (FE) mesh of the cell and the hydrogel are used to calculate improved displacements $\boldsymbol{u}_k^{calc}$ and their associated tractions $\boldsymbol{T}_k$, which fulfill the equilibrium of forces in the hydrogel with the Physically-Based Nonlinear Inverse Method (PBNIM). Icons used freely available from [33,34].

Table 1
External software used in TFMLAB. Please note that Abaqus and FreeFEM are two FE solver alternatives. Only one of them is necessary for force calculation.

Software/toolbox	Freely available	Provided with TFMLAB	Use
Bioformats [35]	Yes	Yes	Microscopy image reading
DIPImage [36]	Yes	No	Image processing: image filtering and enhancement
Elastix [37]	Yes	Yes	Displacement measurement: FFD image registration
Paraview [38]	Yes	No	(Optional) results visualization
iso2mesh [39]	Yes	Yes	Force calculation: Automatic FE meshing
SuiteSparse [40]	Yes	Yes	Force calculation: PBNIM system of equations
Abaqus	No	No	Force calculation: FE solver
FreeFEM [41]	Yes	No	Force calculation: Alternative FE solver

accordingly and stored in D_3_postShiftCorrectedImages. Finally, function *results2paraview* generates .vtk files for vector field and cell segmentation visualization in the free software Paraview [38] and stores them in folder E_resultsParaview.

2.2.4. Force calculation

The user is asked to tune the force calculation parameters in GUI *set_tracCalc*. The cell segmentations S_k are used to create a finite element (FE) mesh of the cell surface and the hydrogel domain with the open source mesh generator iso2mesh [39]. The resultant 3D tetrahedral mesh is coarser away from the cell surface. The GUI allows for tuning the element sizes and smoothening (see details in Appendix D). Moreover, the user can define the mechanical parameters of the hydrogel such as the elastic modulus or the Poisson's ratio (assuming linear elastic,

isotropic behavior). Since the displacement field $\mathbf{u}_k^{meas}$ is measured only at certain discrete locations (beads/fibers), it is not error free and traction recovery is typically done by applying regularization techniques to avoid overfitting. In TFMLAB, we used our novel PBNIM method which proved to be more accurate than a non-regularized method on in silico and in vitro experiments [31,32]. Briefly, this method calculates a displacement field, $\mathbf{u}_k^{calc}$, that is minimally different from the measured one, $\mathbf{u}_k^{meas}$, and that is enforced to fulfill equilibrium of forces in the hydrogel domain as a constraint (see details in Appendix B). PBNIM was implemented in the function *tractionRecovery*. Since PBNIM is developed using a FEM framework, a FE solver is required. TFMLAB is compatible with two FE solvers, namely, the commercial Abaqus (Dassault Systemes Simulia Corporation, Providence, RI) and the open source software FreeFEM [41]. Users

Jorge Barrasa-Fano, Apeksha Shapeti, Álvaro Jorge-Peñas et al.

SoftwareX 15 (2021) 100723

can also pick between one of these FE solvers to compute tractions. Stiffness matrices are computed through the selected FE solver. If Abaqus is selected, TFMLAB provides it with an automatically written job file and imports the results back into the MATLAB workspace. Similarly, TFMLAB automatically interacts with FreeFEM, for which we developed specific FreeFEM routines. The minimization problem (Eq. (E.3)) is then analytically derived turning into a system of equations solved by function *spqr_solve*, which is part of the open source toolbox SuiteSparse [40]. This gives a displacement field $\mathbf{u}_k^{calc}$, which, unlike $\mathbf{u}_k^{meas}$, fulfills the equilibrium of forces in the hydrogel domain. Following the Finite Element Method (FEM) procedure and assuming a linear elastic behavior, stresses and tractions $\mathbf{T}_k$ are forwardly obtained (Eqs. (E.4)–(E.6)). Finally, .vtk files are generated for result visualization. All the results from this step are stored in the output folder F_tractions.

2.2.5. Output and other software features

After computing the full workflow, the user might be interested in repeating a specific step with different parameters while preserving the previous steps. GUIs allow for skipping steps starting from a specific step of the workflow as opposed to running the entire workflow again. All the parameters selected by the user are stored in .mat files for reproducibility. All the important output data (processed images, displacement fields, force fields, etc.) are stored in .mat and text files to allow for further analysis by the user. Together with the .vtk visualization files, TFMLAB also stores a .psvm template for Paraview for automatic 3D vector field rendering.

3. Sample code snippets analysis (optional)

3.1. Illustrative examples

In this section, we present an example of the performance of TFMLAB using experimentally acquired data from an in vitro model of angiogenesis in which Human Umbilical Vein Endothelial Cells (HUVECs) invade a polyethylene glycol (PEG) hydrogel containing suspended 200 nm fluorescent beads. More details on the experimental protocol and microscopy imaging can be found in Appendix A.

For this example, we acquired cell and bead image stacks ($290 \times 290 \times 45 \ \mu m$) by means of confocal microscopy before and after cell relaxation with Cytochalasin D (see Fig. 2a-d and Supplementary Videos 1 and 2), with a voxel size of $0.57 \times 0.57 \times 0.5 \ \mu m^3$. This dataset is provided with the TFMLAB package. An overview of the analysis of this data using TFMLAB is provided in Supplementary Video 3. First, the bead images were processed by applying a Difference of Gaussians (DoG) filter to enhance the spherical shape of the beads and noise removal, followed by a contrast stretching operation (see Fig. 2e-h). Next, the cell images were processed by applying a penalized least squares-based denoising [43] and the cell body was segmented by applying Otsu thresholding and by removing small binary objects (see Fig. 2i-k). Finally, shifts caused by the microscope stage were removed by applying phase-correlation-based rigid image registration (see Fig. 2l-o). Before shift correction, a distinct shift that affects all the beads of the image is present (Fig. 2l, n). After shift correction, the only bead movements remaining happen around the sprout, indicating cell–matrix interactions (Fig. 2m, o).

After image processing, non-rigid displacements are measured by FFD and the PBNIM method is applied to recover cellular forces. Fig. 3a shows TFMLAB's force recovery GUI *set_tracCalc*. The cell surface can be further smoothed prior to FE meshing to ease the meshing process. The user can tune and visualize the

Table 2
Computational times for every step of TFMLAB for the given example (one time point of a TFM experiment).

TFMLAB step	Computational time (s)
Image filtering	25.71
Shift calculation	2.33
Pre-shift correction	19.72
Cell segmentation	9.86
Displacement measurement	63.43
Post-shift correction	31.71
Force calculation	56.15
TOTAL (min)	**3.48**

resultant FE mesh before computing forces. For this example, the resultant FE mesh had ~90 000 4-node 3-D linear tetrahedral elements. The elastic modulus was set to 200 Pa based on experimental results and the Poisson's ratio was assumed to be 0.2. Table 2 shows the computational times to complete every step in TFMLAB on an Intel Core i7-7700 CPU 3.60 GHz with a Windows operating system and 16 GB of RAM. These numbers exclude the time invested in parameter tuning, which for an experienced user can be around 10 min with the majority of this time spent on image filtering, cell segmentation and FE meshing.

Results show a clear pulling pattern near the sprout tip inducing up to ~6 μm matrix displacements (see Fig. 3b). The recovered tractions present maxima at the tips of the cell's protrusion of around 150 Pa (see Fig. 3c). While this displacement field pattern around angiogenic sprouts has been reported in previous studies [30], [44–46], TFMLAB additionally incorporates quantification of traction magnitude and direction. This can be a crucial tool for the quantification of cell mechanical behavior in 3D, ECM-mimicking hydrogels, both for single cells as well as multicellular structures and for applications such as disease modeling, regenerative medicine and tissue engineering as well as developmental biology. Further advantages of TFMLAB are discussed in Section 4.

4. Impact

With this work, we provide an easy-to-use and versatile open source code to perform 4D TFM, making advanced cell mechanical data accessible to researchers with various (including limited) technical backgrounds. Noteworthy features of TFMLAB pertaining to the field of mechanobiology are listed below:

- It is an "all in one" software. Currently, a typical researcher running computational 3D TFM uses e.g. ImageJ for image processing and extra software for displacement and force calculation. This leads to time-consuming work adapting input data to ensure compatibility between software. In contrast, researchers have all the tools they need in TFMLAB.
- It includes state of the art methods of proven usefulness in multiple TFM studies [26–32].
- Its GUIs especially offer ease of usability for researchers with limited programming experience and enable viewing progress of the various steps and parameter tuning.
- As a submodule of TFMLAB, another commercial software (i.e. Abaqus) is available as a sophisticated and technical FE solver. However, to minimize dependency to commercial software, we have integrated FreeFEM, which is an open source alternative to Abaqus. Moreover, the user needs little knowledge on FE modeling as only basic parameter tuning (via the TFMLAB GUI) is required.
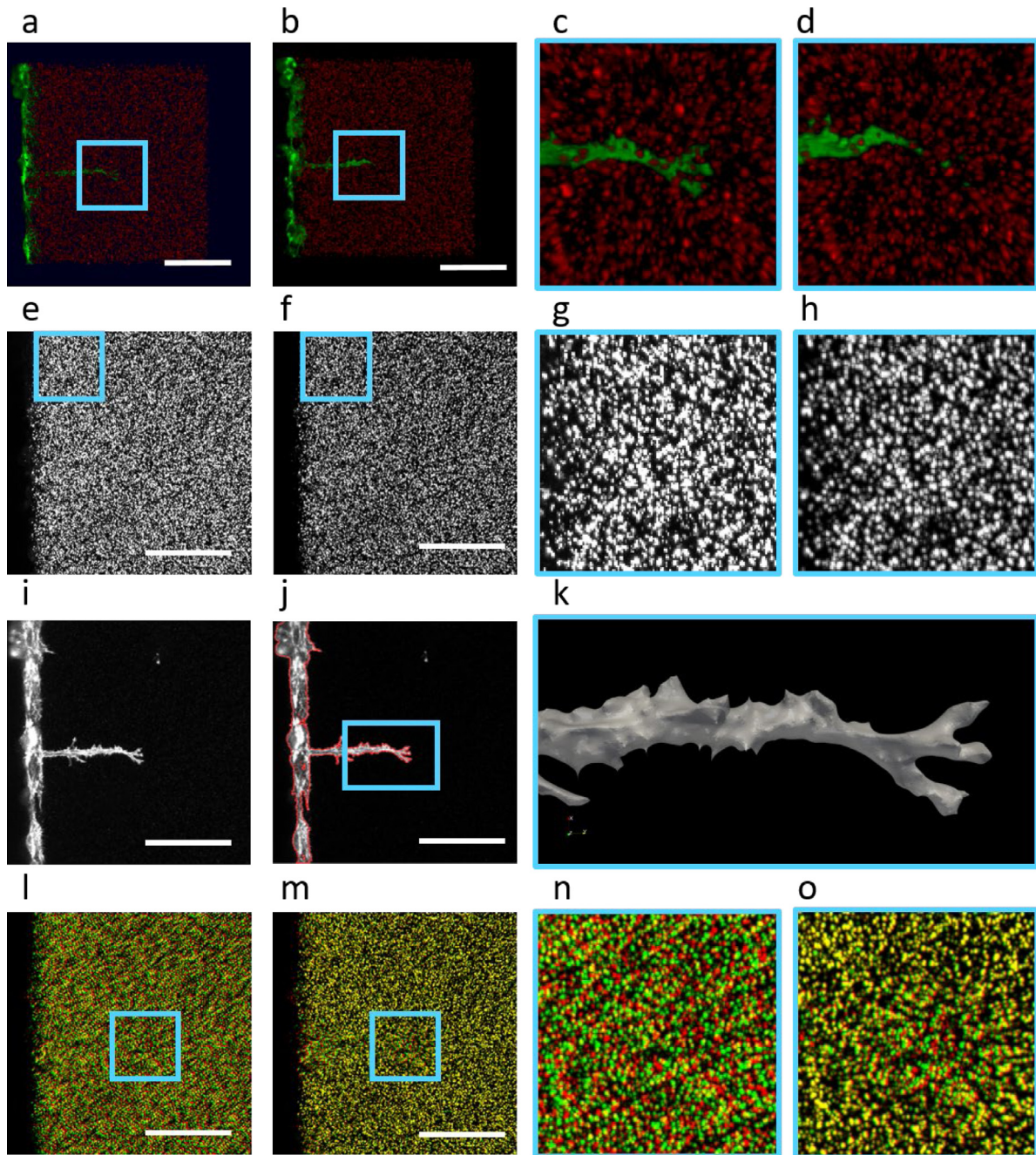
Fig. 2. Image processing steps in TFMLAB. (a,b) 3D renders of the stressed and relaxed images as obtained by means of confocal microscopy (cell channel in green, beads channel in red). (c,d) Zoomed in regions from the raw stressed and relaxed images. (e,f) Maximum projections of a bead image before and after image filtering and enhancement. (g,h) Zoomed in regions before and after image filtering and enhancement. (i) Maximum intensity projection of the raw cell image. (j) Maximum intensity projection of the filtered cell image and segmentation boundary (red). (k) 3D render (Paraview) of the resultant cell segmentation. (l,m) Maximum intensity projection overlay of the stressed (green) and relaxed (red) bead images before and after shift correction. (n,o) Zoomed in regions before and after shift correction. Scale bars: 100 μm.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- It is versatile as it can handle different 3D TFM approaches. Current research includes both bead-based TFM [20,47,48] and fiber-based TFM (with fiber networks e.g. being imaged by means of label-free techniques such as second harmonic generation or confocal reflection microscopy) [23,27, 49]. Our software includes specific image filters for both markers and the FFD algorithm handles both of them [27, 30]. Furthermore, a user can run a classical TFM problem (using a stressed image and a relaxed image, as in the example in Section 3) or time-lapse TFM (using a sequence of stressed images and one relaxed image). If the researcher's

experimental protocol does not include cell imaging, displacement calculation based on the fiducial markers is still possible.

The compactness, user-friendliness and versatility of our toolbox makes it usable in studying many physiological or pathological processes where cell–matrix mechanics play an important role. Such a tool can become key in unraveling cell mechanical time-dependent interactions with their 3D microenvironment, aspects at the forefront of recent mechanobiological research [25,30,49, 50].
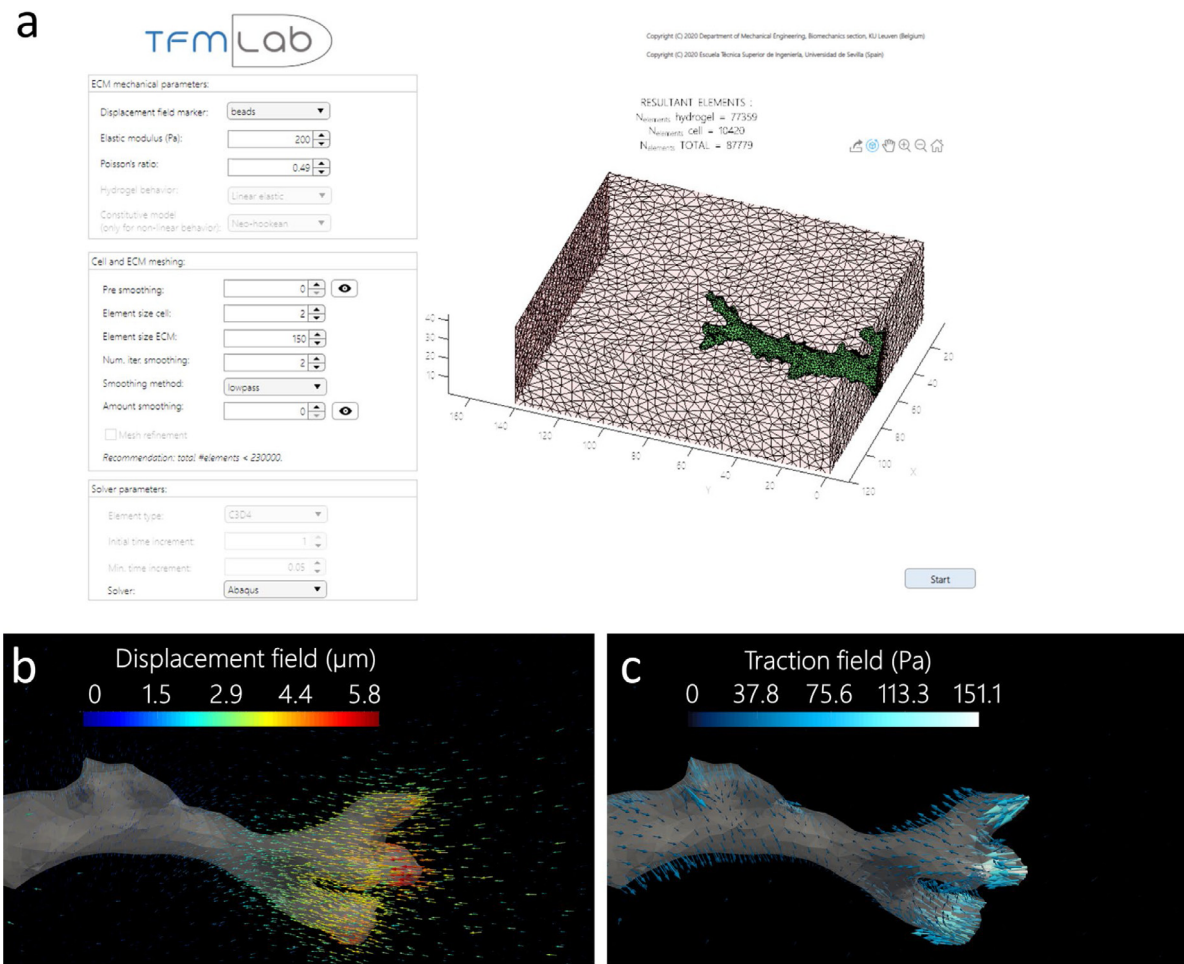
Fig. 3. (a) GUI for FE meshing and force calculation. (b) 3D render (Paraview) of the 3D displacement field (arrows) around the sprout tip as calculated by PBNIM. (c) 3D render (Paraview) of the traction field (arrows) around the tip of the sprout as calculated by PBNIM.

5. Conclusions

TFMLAB is an open-source MATLAB toolbox that seeks to bring 4D TFM closer to medical and biological researchers that do not necessarily have a strong technical background, thereby facilitating the transition of TFM technology from (biomedical) engineering — and/or (bio-)physics-dominated research environments towards medicine and biology. All the necessary steps for computing 4D cellular forces are included in one easy-to-use package (microscopy image reading and processing, displacement measurement, force recovery and visualization file generation). We demonstrated the usefulness and efficiency of the software in an example using real experimental data. We expect that researchers will find TFMLAB useful independent of their programming or technical background.

CRediT authorship contribution statement

Jorge Barrasa-Fano: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing - original draft, Writing - review & editing, Visualization, Funding acquisition. **Apeksha Shapeti:** Investigation, Writing - original draft, Writing - review & editing, Funding acquisition. **Álvaro Jorge-Peñas:** Conceptualization, Methodology, Software, Writing - review & editing. **Mojtaba Barzegari:** Software, Writing - review & editing. **José Antonio Sanz-Herrera:** Methodology, Software, Resources, Writing - review & editing, Supervision, Funding acquisition. **Hans Van Oosterwyck:** Resources, Writing - review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Appendix A. Experimental protocol

Commercially available pooled wild type human umbilical vein endothelial cells (HUVEC, Angio-Proteomie) were cultured in complete endothelial growth medium (EGM-2, Lonza) and used at passage 4. HUVECs were transduced with adenoviral LifeAct-GFP2 (Ibidi) on the day before hydrogel preparation. Enzymatically crosslinked poly-ethylene glycol (PEG) hydrogels comprised of an MMP-sensitive peptide modified PEG precursor (8-arm 40 kDa), Lys-RGD peptide (Pepmic), and 0.1 μM Sphingosine-1-Phosphate (Sigma-Aldrich) were suspended with 0.2 μm red fluorescent carboxylated polystyrene microspheres (ThermoFischer). PEG solution was pipetted into a previously described [30] imaging chamber attached to the glass bottom of a Petri dish, and further allowed to crosslink for 30 min at room temperature. A confluent cell monolayer was then achieved by seeding 50,000 LifeAct transduced HUVECs per gel and incubating the dishes vertically at 37 °C, 5% CO_2 to allow cell adhesion onto the PEG meniscus. Finally, EGM-2 was added and dishes were placed horizontally at standard conditions in the incubator for 16h before experimentation.

Sprouts invading the PEG hydrogel were imaged using a Leica SP8 inverted confocal microscope with a 25 × 0.95 NA water-immersion objective to obtain image stacks of 290 × 290 × 45 μm at 0.5 μm z-spacing. The green fluorescent cell channel and the red fluorescent beads channel were simultaneously imaged to obtain two sets of images; first when the beads are in a stressed state and the cells are contractile, and the second when the beads are in a relaxed state and the cells are non-contractile. The stressed state was imaged immediately after placing the dish upon the stage and locating the sprout of interest. The relaxed state was then induced by treating the cells with Cytochalasin D (dissolved in DMSO, Sigma Aldrich) at 4 μM for 50 min followed by image acquisition in the same location.

Appendix B. Available image filters

For filtering the bead images, TFMLAB takes advantage of the spherical shape of the fluorescent beads and combined a difference-of-Gaussian (DOG) filter [51], a contrast stretching operation and a Gaussian smoothing.

The DOG operator consists of subtracting two blurred versions of the original image, $I_{original}$:

$$I_{filtered} = \left(I_{original} \circledast G_{\sigma_1}\right) - \left(I_{original} \circledast G_{\sigma_2}\right) \quad (B.1)$$

where G_σ is a Gaussian filter with standard deviation σ, $\sigma_1 < \sigma_2$, and $\circledast$ is the convolution operator. The values of σ_1 and σ_2 correspond to parameters 'var1' and 'var2' in the GUI. After this operation, some contrast loss occurs. To correct for this the histogram of $I_{filtered}$ is clipped between a lower and upper intensity value followed by a linear contrast stretching operation (levels correspond to parameter 'contrast stretch [min,max]' in the GUI). Finally, the image is convolved with a Gaussian filtered with a standard deviation σ_3 (value corresponds to parameter 'smooth var' in the GUI) to retrieve the original shape of the fluorescent beads.

For processing cell and fiber images TFMLAB offers the possibility of increasing the intensity of dim areas of the image by applying a log transform. A logarithmic transform amplifies the intensity of the voxels with lower intensity values. This may be useful in cases in which the fluorescence of the cell or the signal from the fibers is low at certain regions. Images are then filtered using penalized least squares-based denoising algorithm described in [43]. This method is especially robust in preserving the edges in the image while suppressing the noise. Finally, a contrast stretching operation is also applied to compensate for the loss of contrast after smoothing.

Appendix C. Choosing a grid spacing for the FFD algorithm

The main parameter that determines the smoothness of the displacement field measured by means of FFD is the grid spacing. In FFD, we use a uniform grid of control points that define a B-spline non-rigid transformation. This grid is defined by the spacing between grid nodes. The higher the spacing, the more global the displacement measurement will be (and thus, one will obtain a smoother result). The lower the spacing, the more local displacements one can capture. For TFM applications, one typically searches for local deformations (normally around cell protrusions, which are small, compared to the size of the image). A grid spacing of between 8 and 16 voxels is typically acceptable to recover local deformations. A possible issue with choosing even lower grid sizes is that irregular deformations (for e.g. relatively high local displacements away from the cells) can appear in parts of the image where there are no beads that can guide the registration. In our experience, a grid spacing of around 5 μm (equivalent to 9 voxels given the voxel size used in this paper) provides displacement field patterns where we clearly see higher displacements around a cell's protrusions and a relatively smooth displacement decay away from the cell. Moreover, with this setting we do not normally obtain irregular deformations. This is the default value for the grid spacing in TFMLAB but the user can modify it in the GUI, prior to displacement field measurement.

Appendix D. Smoothing and choosing size for FE

For traction recovery, one can heavily manipulate the geometry of the cell with the smoothing options and with the size of the meshing elements. It is hard to recommend optimal smoothing and mesh size values since cell morphology is highly heterogeneous and the smoothness of the cell segmentation depends on the quality of the imaging. As a rule of thumb, one should minimize the amount of smoothing since too much smoothing can lead to losing important structures of the cell surface, which is a critical region where tractions are recovered. It is recommended to slightly smooth the cell surface to remove possible spikes that may appear after meshing (see an example in supplementary video 3), but always controlling that the general shape of the cell geometry is maintained. We chose the size of the finite elements to mesh the cell surface and the hydrogel domain based on our previous works [31,32]. We suggest that other labs perform a prior sensitivity analysis to adapt these parameters to their own image characteristics. For example, they can recover tractions for a set of decreasing values of the mesh size and identify the size at which the range of traction magnitude does not vary significantly. The smaller the size of the elements, the higher the computational cost. Therefore, one should find a compromise between numerical accuracy and computational efficiency.

Appendix E. Traction recovery by means of PBNIM

PBNIM was first developed in the context of nonlinear elasticity. A detailed description of the numerical implementation of this method is provided in [31]. TFMLAB uses an adapted implementation for linear elasticity, as described in [32]. Briefly, this method calculates a displacement field, $\mathbf{u}^{calc}$, that is minimally different from the measured one, $\mathbf{u}^{meas}$, and that is enforced to fulfill equilibrium of forces in the hydrogel domain as a constraint. This can be mathematically written as follows,

$$\min_{\mathbf{u}_c^{calc}, \mathbf{u}_h^{calc}} \left(\frac{1}{2} \left\| \mathbf{u}_c^{calc} - \mathbf{u}_c^{meas} \right\|_2^2 + \frac{1}{2} \left\| \mathbf{u}_h^{calc} - \mathbf{u}_h^{meas} \right\|_2^2 \right)$$

$$\text{s.t.} \quad (E.1)$$

$$\mathbf{F}_h = 0$$

where **F** is the nodal reaction forces vector which accounts for external or internal prescribed forces or displacements and c and h denote the cell boundary domain and the hydrogel domain, respectively. Using a FEM framework and including the equilibrium constraint as a Lagrange multiplier, Eq. (B.1) can be rewritten as,

$$\min_{\mathbf{u}_c^{calc}, \mathbf{u}_h^{calc}} \left(\frac{1}{2} \left\| \mathbf{u}_c^{calc} - \mathbf{u}_c^{meas} \right\|_2^2 + \frac{1}{2} \left\| \mathbf{u}_h^{calc} - \mathbf{u}_h^{meas} \right\|_2^2 + \Theta \cdot \eta \right) \quad \text{(E.2)}$$

where Θ is the equilibrium constraint equation and η is the Lagrange multiplier. The minimum of Eq. (E.2) can be calculated analytically and rewritten in matrix form as (the details or the formulation are given in [32]),

$$\begin{bmatrix} \mathbf{I} & 0 & \mathbf{K}_{ch} \\ 0 & \mathbf{I} & \mathbf{K}_{hh} \\ \mathbf{K}_{hc} & \mathbf{K}_{hh} & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u}_c^{calc} \\ \mathbf{u}_h^{calc} \\ \eta \end{bmatrix} = \begin{bmatrix} \mathbf{u}_c^{meas} \\ \mathbf{u}_h^{meas} \\ 0 \end{bmatrix} \quad \text{(E.3)}$$

where $\mathbf{K}_{hc}$, $\mathbf{K}_{ch}$ and $\mathbf{K}_{hh}$ are the parts of the stiffness matrix associated with the two domains c and h. TFMLAB assumes a negligible stiffness for the cell internal domain (Young's modulus of 10-6 Pa).

Next, the strain field at each Gauss point of an element (e) is computed as,

$$\varepsilon^{(e)} = \mathbf{B} \cdot \mathbf{u}^{calc\,(e)} \quad \text{(E.4)}$$

where **B** is the gradient matrix of the shape functions. The stress tensor is calculated at each Gauss point of an element (e) based on the linear elastic, isotropic and homogeneous constitutive properties as,

$$\boldsymbol{\sigma}^{(e)} = 2\mu \cdot \varepsilon^{(e)} + \lambda \cdot \text{tr}(\varepsilon^{(e)})\mathbf{I} \quad \text{(E.5)}$$

where μ and λ are the Lamé constants of the hydrogel. $\boldsymbol{\sigma}^{(e)}$ is then averaged from Gauss points to the nodes i of the FE mesh. Tractions are computed at the cell boundary domain using the Cauchy relation,

$$\mathbf{T}_i = \boldsymbol{\sigma}_i \cdot \mathbf{n}_i \quad \text{(E.6)}$$

where $\mathbf{T}_i$ is the traction vector at the nodes i of the FE mesh and $\mathbf{n}_i$ is the outward normal to node i.

Appendix F. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2021.100723.

References

[1] Kraning-Rush CM, Califano JP, Reinhart-King CA. Cellular traction stresses increase with increasing metastatic potential. PLoS One 2012;7(2).

[2] Jansen KA, Bacabac RG, Piechocka IK, Koenderink GH. Cells actively stiffen fibrin networks by generating contractile stress. Biophys. J. 2013;105(10):2240–51.

[3] Schwarz US, Soiné JRD. Traction force microscopy on soft elastic substrates: A guide to recent computational advances. Biochim Biophys Acta - Mol Cell Res 2015;1853(11):3095–104.

[4] Kutys ML, Chen CS. Forces and Mechanotransduction in 3D Vascular Biology. Curr Opin Cell Biol, 42, Elsevier Ltd; 2016, p. 73–9.

[5] Mulligan JA, Bordeleau F, Reinhart-King CA, Adie SG. Measurement of dynamic cell-induced 3D displacement fields in vitro for traction force optical coherence microscopy. Biomed Opt Express 2017;8(2):1152.

[6] Polacheck WJ, Chen CS. Measuring cell-generated forces: A guide to the available tools. Nature Methods 2016;13(5):415–23.

[7] Dembo M, Wang Y-L. Stresses at the cell-to-substrate interface during locomotion of fibroblasts. Biophys. J. 1999;76(4):2307–16.

[8] Balaban others NQ. Force and focal adhesion assembly: A close relationship studied using elastic micropatterned substrates. Nat Cell Biol 2001;3(5):466–72.

[9] Engler AJ, Sen S, Sweeney HL, Discher DE. Matrix elasticity directs stem cell lineage specification. Cell 2006;126(4):677–89.

[10] Mori Nobuhito. Frontpage - mpiv - MATLAB PIV toolbox. 2020, [Online]. Available: http://www.oceanwave.jp/softwares/mpiv/ [Accessed: 07-Apr-2020].

[11] Blair D, Dufresne E. Matlab particle tracking. 2020, [Online]. Available: http://site.physics.georgetown.edu/matlab/ [Accessed: 07-Apr-2020].

[12] Hansen PC. Regtools - file exchange - MATLAB central. 2020, [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/52-regtools [Accessed: 07-Apr-2020].

[13] The Franck Lab. DOWNLOADS | Francklaboratory. 2020, [Online]. Available: https://www.franck.engin.brown.edu/downloads [Accessed: 07-Apr-2020].

[14] Style others RW. Traction force microscopy in physics and biology. Soft Matter 2014;10(23):4047–55, Royal Society of Chemistry.

[15] Huang Y, Gompper G, Sabass B. A Bayesian traction force microscopy method with automated denoising in a user-friendly software package. Comput. Phys. Comm. 2020;107313.

[16] Han SJ, Oak Y, Groisman A, Danuser G. Traction microscopy to identify force modulation in subresolution adhesions. Nature Methods 2015;12(7):653–6.

[17] Tseng Q. Traction force microscopy - imagej plugins. 2020, [Online]. Available: https://sites.google.com/site/qingzongtseng/tfm [Accessed: 07-Apr-2020].

[18] Song D, Hugenberg N, Oberai AA. Three-dimensional traction microscopy with a fiber-based constitutive model. Comput. Methods Appl. Mech. Engrg. 2019;357:112579.

[19] Legant WR, Miller JS, Blakely BL, Cohen DM, Genin GM, Chen CS. Measurement of mechanical tractions exerted by cells in three-dimensional matrices. Nature Methods 2010;7(12):969–71.

[20] Gjorevski N, Piotrowski AS, Varner VD, Nelson CM. Dynamic tensile forces drive collective cell migration through three-dimensional extracellular matrices. Sci. Rep. 2015;5:11458.

[21] Cóndor M, García-Aznar JM. An iterative finite element-based method for solving inverse problems in traction force microscopy. Comput Methods Program Biomed 2019;182:105056.

[22] Lejeune E, Khang A, Sansom J, Sacks MS. FM-Track: A fiducial marker tracking software for studying cell mechanics in a three-dimensional environment. SoftwareX 2020;11.

[23] Steinwachs others J. Three-dimensional force microscopy of cells in biopolymer networks. Nature Methods 2016;13(2):171–6.

[24] Cóndor M, Steinwachs J, Mark C, García-Aznar JM, Fabry B. Traction force microscopy in 3-dimensional extracellular matrix networks. In: Current protocols in cell biology, Vol. 75. Hoboken, NJ, USA: John Wiley & Sons, Inc; 2017, p. 10.22.1–20, no. 1.

[25] Mark others C. Collective forces of tumor spheroids in three-dimensional biopolymer networks. Elife 2020;9.

[26] Jorge-Peñas others A. Free form deformation-based image registration improves accuracy of traction force microscopy. PLoS One 2015;10(12):1–22.

[27] Jorge-Peñas others A. 3D Full-field quantification of cell-induced large deformations in fibrillar biomaterials by combining non-rigid image registration with label-free second harmonic generation. Biomaterials 2017;136:86–97.

[28] Izquierdo-Álvarez A, Vargas DA, Jorge-Peñas Á, Subramani R, Vaeyens M-M, Van Oosterwyck H. Spatiotemporal analyses of cellular tractions describe subcellular effect of substrate stiffness and coating. Ann Biomed Eng 2019;47(2):624–37.

[29] Steuwe others C. Fast quantitative time lapse displacement imaging of endothelial cell invasion. PLoS One 2020;15(1):e0227286.

[30] Vaeyens others M-M. Matrix deformations around angiogenic sprouts correlate to sprout dynamics and suggest pulling activity. Angiogenesis 2020.

[31] Sanz-Herrera JA, Barrasa Fano J, Cóndor M, Van Oosterwyck H. Inverse method based on 3D nonlinear physically constrained minimisation in the framework of traction force microscopy.. Soft Matter 2020.

[32] Barrasa-Fano J, Shapeti A, De Jong J, Ranga A, Sanz-Herrera JA, Van Oosterwyck H. Advanced in silico validation framework for three-dimensional traction force microscopy and application to an in vitro model of sprouting angiogenesis. Acta Biomater 2021.

[33] Agus raharjo on iconfinder. 2020, [Online]. Available: https://www.iconfinder.com/agusraharj [Accessed: 20-Oct-2020].

[34] Minimal disease icons by picturewindow. 2020, [Online]. Available: https://www.iconfinder.com/iconsets/minimal-disease [Accessed: 20-Oct-2020].

[35] Linkert others M. Metadata matters: Access to image data in the real world. J. Cell Biol. 2010;189(5):777–82, The Rockefeller University Press.

[36] Dipimage. 2020, [Online]. Available: http://www.diplib.org/dipimage [Accessed: 04-May-2020].

[37] Klein S, Staring M, Murphy K, Viergever MA, Pluim JPW. Elastix: A toolbox for intensity-based medical image registration. IEEE Trans Med Imaging 2010;29(1):196–205.

[38] Ayachit U. The paraview guide: A parallel visualization application. Kitware 2015.

[39] Fang Qianqian, Boas DA. Tetrahedral mesh generation from volumetric binary and grayscale images. In: 2009 IEEE international symposium on biomedical imaging: from nano to macro. 2009, p. 1142–5.

[40] Davis TA. Algorithm 915, suitesparseqr: multifrontal multithreaded rank-revealing sparse qr factorization. ACM Trans Math Softw 2011;38(1):1–22.

[41] Hecht F. New development in freefem+. J Numer Math 2012;20(3–4):251–65.

[42] Jorge-Peñas A, Muñoz Barrutia A, De-Juan-Pardo EM, Ortiz-De-Solorzano C. Validation tool for traction force microscopy. Comput. Methods Biomech. Biomed. Eng. 2015;18:1377–85.

[43] Garcia D. Robust smoothing of gridded data in one and higher dimensions with missing values. Comput. Statist. Data Anal. 2010;54(4):1167–78.

[44] Du Y, Herath SCB, Wang Q, Wang D, Asada HH, Chen PCY. Three-dimensional characterization of mechanical interactions between endothelial cells and extracellular matrix during angiogenic sprouting. Sci. Rep. 2016;6(1):21362.

[45] Yoon others C. Myosin IIA–mediated forces regulate multicellular integrity during vascular sprouting. Mol Biol Cell 2019;30(16):1974–84.

[46] Du Y, Herath SCB, Wang QG, Asada H, Chen PCY. Determination of green's function for three-dimensional traction force reconstruction based on geometry and boundary conditions of cell culture matrices. Acta Biomater 2018;67:215–28.

[47] Mulligan JA, Feng X, Adie SG. Quantitative reconstruction of time-varying 3D cell forces with traction force optical coherence microscopy. Sci. Rep. 2019;9(1).

[48] Dong L, Oberai AA. Recovery of cellular traction in three-dimensional nonlinear hyperelastic matrices. Comput. Methods Appl. Mech. Engrg. 2017;314:296–313.

[49] Shakiba others D. The balance between actomyosin contractility and microtubule polymerization regulates hierarchical protrusions that govern efficient fibroblast–collagen interactions. ACS Nano 2020.

[50] Cóndor others M. Breast cancer cells adapt contractile forces to overcome steric hindrance. Biophys. J. 2019;116(7):1305–12.

[51] Collins RT. Mean-shift blob tracking through scale space. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition, Vol. 2*, 2003.