RESEARCH ARTICLE

MicroBundleCompute: Automated segmentation, tracking, and analysis of subdomain deformation in cardiac microbundles

Hiba Kobeissi[1,2], Javiera Jilberto[3], M. Çağatay Karakan[1,4,5], Xining Gao[5,6,7], Samuel J. DePalma[3], Shoshana L. Das[5,6,7], Lani Quach[3], Jonathan Urquia[8], Brendon M. Baker[3], Christopher S. Chen[5,7], David Nordsletten[3,9,10], Emma Lejeune[1,2]*

1 Department of Mechanical Engineering, Boston University, Boston, MA, United States of America, **2** Center for Multiscale and Translational Mechanobiology, Boston University, Boston, MA, United States of America, **3** Department of Biomedical Engineering, University of Michigan, Ann Arbor, MI, United States of America, **4** Photonics Center, Boston University, Boston, MA, United States of America, **5** Department of Biomedical Engineering, Boston University, Boston, MA, United States of America, **6** Harvard-MIT Program in Health Sciences and Technology, Institute for Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA, United States of America, **7** Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, MA, United States of America, **8** Department of Electrical and Computer Engineering, New York Institute of Technology, New York, NY, United States of America, **9** Department of Cardiac Surgery, University of Michigan, Ann Arbor, MI, United States of America, **10** Department of Biomedical Engineering, School of Imaging Sciences and Biomedical Engineering, King's Health Partners, King's College London, King's Health Partners, London, United Kingdom

* elejeune@bu.edu

Abstract

Advancing human induced pluripotent stem cell derived cardiomyocyte (hiPSC-CM) technology will lead to significant progress ranging from disease modeling, to drug discovery, to regenerative tissue engineering. Yet, alongside these potential opportunities comes a critical challenge: attaining mature hiPSC-CM tissues. At present, there are multiple techniques to promote maturity of hiPSC-CMs including physical platforms and cell culture protocols. However, when it comes to making quantitative comparisons of functional behavior, there are limited options for reliably and reproducibly computing functional metrics that are suitable for direct cross-system comparison. In addition, the current standard functional metrics obtained from time-lapse images of cardiac microbundle contraction reported in the field (i.e., post forces, average tissue stress) do not take full advantage of the available information present in these data (i.e., full-field tissue displacements and strains). Thus, we present "MicroBundleCompute," a computational framework for automatic quantification of morphology-based mechanical metrics from movies of cardiac microbundles. Briefly, this computational framework offers tools for automatic tissue segmentation, tracking, and analysis of brightfield and phase contrast movies of beating cardiac microbundles. It is straightforward to implement, runs without user intervention, requires minimal input parameter setting selection, and is computationally inexpensive. In this paper, we describe the methods underlying this computational framework, show the results of our extensive validation studies, and demonstrate the utility of exploring heterogeneous tissue deformations and strains as

OPEN ACCESS

Citation: Kobeissi H, Jilberto J, Karakan MÇ, Gao X, DePalma SJ, Das SL, et al. (2024) MicroBundleCompute: Automated segmentation, tracking, and analysis of subdomain deformation in cardiac microbundles. PLoS ONE 19(3): e0298863. https://doi.org/10.1371/journal.pone.0298863

Editor: Mohammad Amin Fraiwan, Jordan University of Science and Technology, JORDAN

Received: August 31, 2023

Accepted: February 1, 2024

Published: March 26, 2024

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0298863

Copyright: © 2024 Kobeissi et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The data underlying the results presented in the study are available from https://datadryad.org/stash/dataset/doi:10.5061/dryad.5x69p8d8g. The code to reproduce the results in this work is available from https://github.com/HibaKob/SyntheticMicroBundle and https://github.com/HibaKob/MicroBundleCompute.

Funding: This work was supported by the CELL-MET Engineering Research Center National Science Foundation ERC ECC-1647837. EL acknowledges support from the American Heart Association Career Development Award 856354, XG acknowledges support from the National Science Foundation Graduate Research Fellowship, BMB and SJD acknowledge support from NSF Award 2033654, and SJD acknowledges support from NIH T32-DE007057 and NIH T32-HL125242. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

functional metrics. With this manuscript, we disseminate “MicroBundleCompute” as an open-source computational tool with the aim of making automated quantitative analysis of beating cardiac microbundles more accessible to the community.

Introduction

Despite significant recent advances in cardiovascular disease prevention and diagnosis [1, 2], heart disease remains the leading cause of death among adults worldwide [3]. This is due, in part, to the fact that the native heart has a poor regenerative ability [4, 5], and thus damage to the heart muscle during an adverse medical event such as a myocardial infarction is irreversible [6]. Cardiac tissue engineering is a promising approach to address this unmet societal need [5]. In particular, cardiac tissue engineering with human induced pluripotent stem cell derived cardiomyocyte (hiPSC-CM) based technology [7] is a promising approach to disease modeling [8–11], drug discovery [5, 9, 11–13], and regenerative tissue engineering [14–16]. However, the development of viable hiPSC-CM technology is very much ongoing. In particular, one major challenge is that differentiated hiPSCs initially resemble fetal cardiomyocytes—they are morphologically and functionally different compared to adult cardiomyocytes [12]. Thus, developing technology to promote the maturation of hiPSC-CMs and, likewise, hiPSC-CM based tissue is an active area of research [16]. One impactful approach to promoting the maturation of hiPSC-CMs is the use of engineered tissue culture platform designs in both two [17, 18] and three dimensions [19–23] (Fig 1). Across these different platforms, there are multiple physical [17, 19, 22, 24], electrical [19, 20, 23–25], and chemical [26–28] knobs to tune to promote maturation and explore different physiological and pathological conditions. Even if we restrict our focus to microbundles (i.e., *aligned, electromechanically coupled, microscale cardiac tissue bundles formed with hydrogel materials suspended between pillars*), there is massive variability across different experimental setups [17, 19, 20, 22].

Driven by this diversity in experimental approaches and the rapid growth of the field, the mechanical behavior of cardiac microbundles is challenging to compare across studies. Fundamentally, this challenge is driven by multiple factors, ranging from the high volume of data collected with these testbeds [29], to challenges associated with reproducing results when software and data are not shared under open-source licenses, or when extracting quantities of interest from data requires significant manual processing. To date, there have been multiple non-destructive image-based methods for quantifying the contractile action of cardiac microbundles [21, 30–39], often inspired by related approaches to assessing the contractile behavior of cardiomyocytes [30, 40–50]. Broadly speaking, most of these tools can be grouped into four main categories: (1) edge detection systems [31, 32], (2) pillar tracking-based methods [33–37, 39], (3) inter-frame pixel disparity methods [21, 38], and (4) optical flow-based tracking [30, 39]. Each one of these methods has benefits and limitations that suit specific platforms, conditions, and research questions (some of these approaches are elaborated on in the “Materials and methods” Section). Despite this wide range of computational tools, many of which are available under open-source licenses [21, 33, 34, 38, 39, 51], few options can compare across multiple experimental testbeds and function on new datasets out of the box. In addition, the most popular approach to assessing microbundle contractile behavior—pillar tracking—does not necessarily capture the full richness of mechanical behavior in these systems.

Given this research landscape, there is a clear need for an open-source computational tool to extract functional metrics from time-lapse images of microbundle contraction. To address

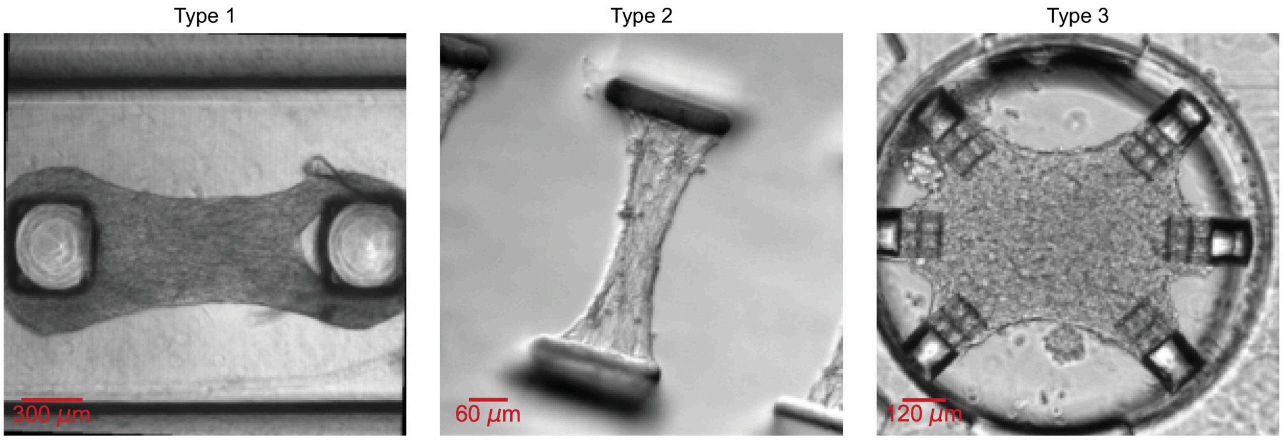

Fig 1. Examples of 3 microbundle experimental testbeds of different types. We briefly note here that "Type 1" and "Type 3" represent 3D culture platforms while "Type 2" portrays a significantly thinner (almost 2D) platform. We elaborate more on each type in the "Experimental data" Section and in S2 Appendix.

https://doi.org/10.1371/journal.pone.0298863.g001

this need, we have developed the "MicroBundleCompute" computational framework that we will present in this paper. In brief, "MicroBundleCompute" is disseminated as a Python package and is based on the Lucas-Kanade optical flow algorithm [52] for computing full-field displacements, subdomain-averaged strains, and displacement and strain derived metrics. In this manuscript, we introduce the methods underlying the "MicroBundleCompute" framework, discuss our approach to validating the pipeline primarily via realistic synthetic data, and then show the results of implementing "MicroBundleCompute" on multiple testbeds, shown in Fig 1. We not only portray the efficacy of our approach, but also show that examining full-field tissue deformation consistently reveals heterogeneous contractile behavior throughout the domain. Looking forward, our goal is to use this work as a starting point to move beyond previously developed ad hoc approaches to analyzing these data [53], and establish a computational foundation for performing cardiac microbundle assessment and quality control at scale.

The remainder of this document is organized as follows. In the "Materials and methods" Section, we describe our methods to generate synthetic movies of beating cardiac microbundles with known ground truth to validate our computational pipeline, the diverse set of real examples on which we test the software, and finally, our code pipeline along with the main output metrics. Then in the "Results and discussion" Section, we present the main findings of our code validation on one synthetic example and show sample outputs on a few real movies. Finally, in the "Conclusion" Section, we share our final thoughts regarding "MicroBundleCompute." Along with this document, we provide three supplementary documents where we explain in more details our validation pipeline (S1 Appendix), share our complete set of real examples (S2 Appendix), and finally provide more information on basic pillar tracking (S3 Appendix), an additional feature of our computational tool. Overall, the intention of this manuscript is to outline our method and approach to making a tool of broad utility for the cardiac tissue engineering research community.

Materials and methods

In a broader sense, capturing full-field soft tissue deformation is critical for a number of research applications ranging from inverse material characterization [54–57], to high-fidelity biomechanical modeling of in vivo mechanisms [58–60], and patient-specific modeling and procedure planning [61, 62]. And recently, there has been keen interest in implementing image registration-based techniques widely used in the field of computer vision [54, 63] for full-field measurements, including digital image correlation (DIC) [64–68], 3D-DIC [69, 70], digital volume correlation (DVC) [71–74], and optical flow algorithms [75–77].

With regards to applications to cardiac microbundles in specific, available image-based techniques for computing the contractile behavior of microbundles, as categorized in the "Introduction" into 4 broad approaches, focus mainly on quantifying temporal profiles for the entire construct and as such, provide averaged results that lump the spatially heterogeneous tissue behavior into a single value per frame (3 of the 4 approaches). However, tools developed based on optical flow (4^{th} approach) can extract full-field, as well as directional outputs, such as directional displacement fields and strains.

To elaborate more, edge-detection approaches rely on quantifying the microbundle shape change between a relaxed (reference) state and a contracted (deformed) state. For example, in Ronaldson-Bouchard et al. [31], tissue contractility was calculated by tracking the change in the tissue area while previously, in Hansen et al. [32], the difference between the ends of the tissue was used to measure contraction. Yet, these tools were custom-developed and are not readily available online for the broader research community. As for pillar tracking-based methods, there are currently a number of available tools [33, 34, 39] out of the identified implementations that were developed and kept in-house [35–37]. In these approaches, the pillar or cantilever head deflection is estimated to generate contraction waveforms and extract contraction kinetics including contraction frequency, force, as well as the time to achieve 10%, 50%, or 90% of the peak contraction or relaxation.

Available tissue tracking methods via pixel intensity disparity have been implemented based on different approaches. For example, in "MUSCLEMOTION" [51], which is offered as an ImageJ [78] plugin, the absolute difference in pixel intensity between a reference frame and a frame of interest is calculated, whereas the MATLAB-based [79] "CardiacContractileMotion" [21] identifies the tissue region in a relaxed baseline state and tracks, within this region of interest, changes in pixel motion with time. Another MATLAB-based [79] tool, "ContractQuant" [38], which was specifically developed for implementation with micron-scale 2D cardiac muscle bundles, uses cross-correlation to track pixel features and find the best match for a specified region of interest across consecutive frames. Outputs from these software are in general similar to those extracted with pillar tracking approaches and include contraction and relaxation profiles and velocities, as well as contraction and relaxation times.

Overall, these approaches are suitable when only averaged values, such as the mean value and the time rate or velocity of tissue shortening, shrink or contraction, are enough. However, tools developed based on optical flow can provide richer outputs. For example, Huebsch et al. [30] implemented block-matching optical flow [80] methods in MATLAB [79] to estimate absolute as well as directional full-field contractility. And very recently, a particle image velocimetry toolbox in MATLAB [79] was utilized in [39] to calculate displacement vectors of subregions or patches identified within a manually selected region of interest in the "reference" frame using a cross-correlation approach. And from these displacement fields, strain maps were subsequently derived. To the best of our knowledge, these methods appear to be robust and relatively versatile, yet at present they lack automation.

Within this scope, we present here our high-throughput optical flow-based computational framework to extract full-field deformation metrics from lab-grown cardiac microbundles. In the Sections that follow, we describe both the data used for testing and validating the "MicroBundleCompute" software, and the details of our computational methods. First, in the "Data" Section, we introduce the two general categories of cardiac microbundle data that we have used in developing the software: synthetic data with a known ground truth and experimental data. Next, in the “Code” Section, we explain the details of the code pipeline and describe core functionalities and output metrics.

Data

To validate our pipeline, we first invest significant effort into creating labeled data with a known ground truth. In the “Synthetic data generation” Section, we outline our process for synthetic data generation. As a brief note, this is complemented with additional information in S1 Appendix where we show comparisons to another form of synthetic data and manually labeled experimental data. In the “Experimental data” Section, we then show three distinct classes of microbundle experimental testbeds that we will use to showcase the function and versatility of our software.

Synthetic data generation. Here, we describe the steps to generate realistic synthetic brightfield movies of beating cardiac microbundles based on examples from “Type 1,” as described in the “Experimental data” Section. In Fig 2, we summarize the main steps of this pipeline. For each synthetic data example, we begin with a frame from an experimental movie (Fig 2a). As the first step of this pipeline, we manually trace the tissue region in a relaxed valley frame, and use the traced region to obtain both tissue geometry and image texture. From the traced region, we extract the coordinates of the external contour to generate Finite Element (FE) simulation geometry, and isolate the tissue texture that will be warped following the FE simulation results (Fig 2b).

To inform our FE simulations, we first generate a simplified three-dimensional microbundle geometry based on the contour coordinates of a mask extracted from the single representative valley frame (Fig 2c). Specifically, we extrude the 2D surface created by connecting these contour coordinates to a thickness of 400μm, a reasonable microbundle thickness given our target experimental setups [22]. To approximate the pillars, we implement the geometry and dimensions detailed in [22], which matches one of the main platforms used in our experimental dataset (“Type 1”). To create the FE mesh, we use Gmsh 4.10.5 [81], where the final mesh consists of 205, 524 tetrahedral elements which was deemed sufficient for our purpose, following a mesh refinement study. We provide a detailed schematic of the three-dimensional mesh geometry in S1 Appendix.

In Fig 2d, we briefly summarize the main components of the FE model as implemented in FEniCS 2019.1.0 [82, 83]. Following popular recent work in the field of soft tissue biomechanics [84–86], we model the cardiac tissue as a nearly-incompressible transversely isotropic hyperelastic material where deformation is driven by periodic activation [84, 86, 87]. Of note,

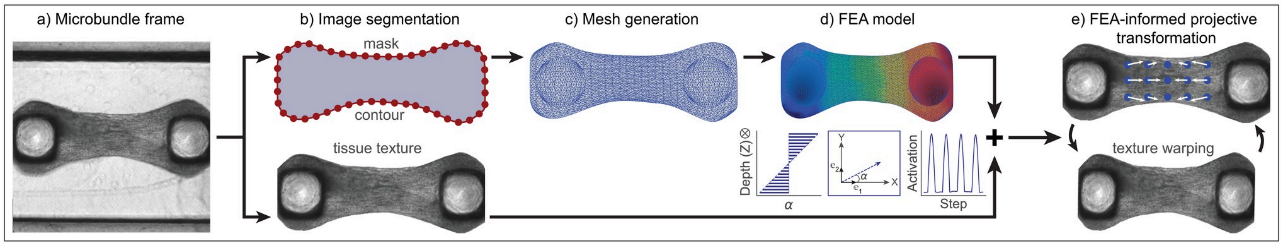

Fig 2. Schematic representation of the synthetic data generation pipeline. Illustrations of the main elements of the synthetic data generation pipeline in order of implementation: (a) an image of a microbundle movie frame; (b) a mask of the microbundle, extracted contour coordinates, and a segmented tissue texture; (c) a three-dimensional mesh for the FE model; (d) main variables within the FE simulations in order from left to right: profile of a linearly variable fiber direction with respect to depth, an illustration of the tissue depth direction, a uniform time series activation; (e) extracted surface displacement results and a warped image texture based on an estimated projective transformation.

https://doi.org/10.1371/journal.pone.0298863.g002

we create synthetic data with heterogeneous deformation fields (specifically, (1) a fully actively contracting tissue domain, (2) a passive circular inclusion at the center of the actively contracting tissue), and we vary the direction of contractile alignment (1) through the thickness or depth of the tissue or (2) along its length. We model the poly(dimethylsiloxane) (PDMS) pillars as passive Neo-Hookean material, and treat the interface between the cardiac tissue and the pillars as perfectly bonded. For each FE simulation, we extract, with respect to time, the *X*, *Y*, and *Z* positions of the mesh cell centers at the top surface of the microbundle for each step and save the results as text files. This simplified FE model serves a single purpose: to computationally generate synthetic data of realistically beating microbundles with known ground truth deformation. Then, following the schematic illustration in Fig 2e, we estimate a projective transformation based on the initial and deformed positions of the mesh cell centers and warp the image texture accordingly using the `warp` transform function in the scikit-image 0.19.3 Python library [88]. To enable a heterogeneous transformation, we subdivide the image domain and perform subdomain specific warping. Additional details are available in S1 Appendix.

Overall, our main synthetic dataset consists of 60 generated movies of beating experimentally derived image textures. To obtain these 60 examples, we use 15 different base texture images extracted from 5 experimental movies of "Type 1" as described in the "Experimental data" Section. We then deform these extracted textures with FE results obtained from 4 different FE simulations run under the variable conditions specified above. To perform quantitative evaluation, we extract a 90 × 90 pixel region from each domain center and make all direct comparisons based on this domain.

In addition, we perform additional validation against a single computationally generated synthetic example of "Type 2" data based on a more sophisticated tissue-specific FE model described in detail in Jilberto et al. [89] and S1 Appendix. For more information on implementing the Finite Element model, image warping, and the addition of Perlin noise [90], we refer the reader to S1 Appendix. We also make the entire synthetic dataset prior to the addition of Perlin noise available along with all the Python code and files that are necessary to re-generate our dataset available on GitHub (https://github.com/HibaKob/SyntheticMicroBundle).

Experimental data. Our experimental dataset can be systematically categorized into 3 different types (Fig 1). "Type 1" includes movies obtained from standard experimental microbundle strain gauge devices [91–93]. We refer to data collected from non-standard platforms termed FibroTUGs [94] as "Type 2" data. As for "Type 3," they represent data obtained from a highly versatile experimental platform [19, 20] and as such, include the most diverse examples in this collection.

Specifically, "Type 1" examples were prepared as previously detailed in [53]. Briefly, PDMS (Dow Silicones Corporation, Midland, MI) microbundle devices were first cast from 3D printed molds (Protolabs). Each device contains 6 wells, each with two pillars with rectangular cross sections and spherical caps, where the cardiac microbundles are seeded. Up to 2 days before seeding, the devices were sequentially treated with 0.01% poly-l-lysine (ScienCell) and then with 0.1% glutaraldehyde (EMS) to promote cell attachment to the caps. On the day of seeding, devices were cleaned with 70% ethanol and ultraviolet (UV) sterilized. Next, the device wells were incubated with a small volume of 2% Pluronic F-127 (Sigma) to prevent cell attachment at the base of the well. hiPSC-CMs, differentiated and purified as described by Lian et al. [95], were seeded with human ventricular cardiac fibroblasts in a Matrigel (Corning) and fibrin (Sigma) extracellular matrix (ECM) solution. Microbundles were maintained in growth medium, with replacement every other day. Time-lapse videos of tissue contractions were acquired 5–7 days after seeding at 30 Hz using a 4× objective on a Nikon Eclipse Ti

(Nikon Instruments Inc.) with an Evolve EMCCD camera (Photometrics), while maintaining a temperature of 37˚C and 5% CO_2.

As for the second type, FibroTUG microbundles were fabricated as described previously [17, 94]. First, arrays of PDMS cantilevers were fabricated by soft lithography as detailed in [94]. Then, fiber matrices, suspended between pairs of these cantilevers, were generated by selective photo-crosslinking of electrospun dextran vinyl sulfone (DVS) fiber matrices deposited onto the microfabricated PDMS cantilevers [17, 94]. Matrix and cantilever stiffnesses were tuned by adjusting photoinitiator concentrations and cantilever height, respectively, while matrix alignment was controlled by altering the translation speed of collection substrates during fiber deposition [17, 94]. Following functionalization of the electrospun fiber matrices with cell adhesive cRGD peptides, iPSC-CMs, differentiated and purified [17], were patterned onto matrices using microfabricated seeding masks cast from 3D-printed molds. Finally, time-lapse videos of the microbundle's spontaneous contractions were acquired at $\sim$ 65 Hz on Zeiss LSM800 equipped with an Axiocam 503 camera while maintaining a temperature of 37˚C and 5% CO_2.

Examples from "Type 3" were generated using the protocol previously described in [19, 20]. In brief, a combination of soft lithography and two-photon direct laser writing (DLW, Nanoscribe Photonic Professional GT+) was used to fabricate the seeding platforms. The process involves printing negative master molds using DLW, casting PDMS onto the molds, followed by sandwiching, curing and demolding. This results in 0.5—0.6 mm-thick PDMS devices with embedded microfluidic channels and deformable seeding wells. As a final step, cage-like microstructures were printed using DLW on the sides of the wells of the demolded PDMS devices to facilitate cell attachment. After device fabrication, differentiated hiPSC-CMs as per the procedures described in [19], were seeded into the wells with human mesenchymal stem cells in a collagen ECM solution, with the growth medium changed every other day. Time-lapse videos of the tissue contractions were acquired 4–9 days after seeding at 30 Hz using 4× or 10× objectives on a Nikon Eclipse Ti (Nikon Instruments Inc.) with an Evolve EMCCD camera (Photometrics) equipped with a temperature and CO_2 equilibrated environmental chamber.

In total, we include in this framework 24 real experimental data, 11 examples from "Type 1," 7 from "Type 2," and 6 from "Type 3." This diverse pool of examples allows us to not only demonstrate the adaptability of our computational pipeline to different input examples, but also gain valuable insight about the heterogeneous contractile action of cardiac tissue by extracting and observing relevant mechanical metrics, such as full-field displacements, subdomain-averaged strains, and displacement and strain-derived outputs, as shown in the "Experimental data examples" Section and in S2 Appendix. We note that details about each specific experimental example are provided in S2 Appendix as well as on Dryad [96] (https://doi.org/10.5061/dryad.5x69p8d8g) where the whole dataset, "Microbundle Time-lapse Dataset," is made available.

Code

In this Section, we describe the main working components of our "MicroBundleCompute" software for the automatic analysis of deformation in brightfield and phase contrast movies of cardiac microbundles. Because our goal is to implement an approach with simplicity, versatility, and adaptability in mind, our pipeline is structured with four modular components: (1) image pre-processing and mask creation, (2) deformation tracking, (3) post-processing (e.g., rotation, interpolation, strain analysis), and (4) visualization. In Fig 3, we provide a graphical summary of the major functionalities included in this pipeline and the computational

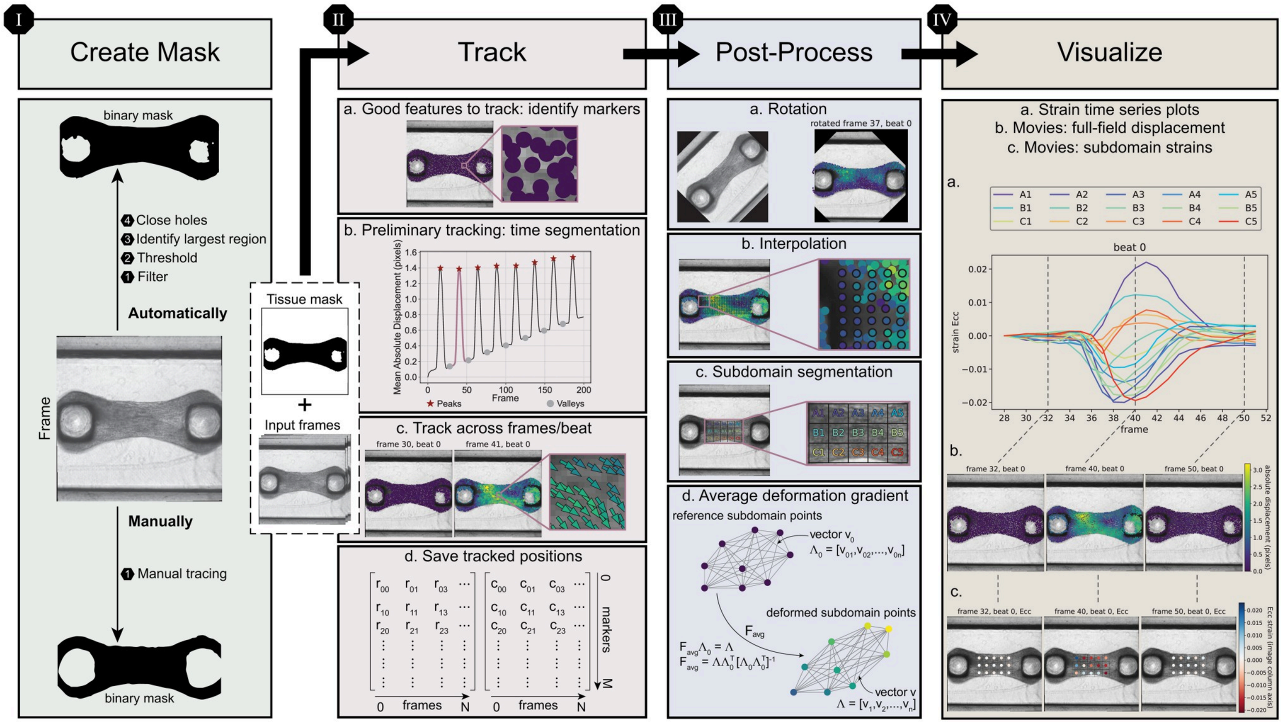
I
Create Mask
binary mask
❹ Close holes
❸ Identify largest region
❷ Threshold
❶ Filter
Automatically
Frame
Manually
❶ Manual tracing
binary mask
Tissue mask
+
Input frames
II
Track
a. Good features to track: identify markers
b. Preliminary tracking: time segmentation
Mean Absolute Displacement (pixels)
Frame
Peaks
Valleys
c. Track across frames/beat
frame 30, beat 0
frame 41, beat 0
d. Save tracked positions
frames
markers
III
Post-Process
a. Rotation
rotated frame 37, beat 0
b. Interpolation
c. Subdomain segmentation
A1 A2 A3 A4 A5
B1 B2 B3 B4 B5
C1 C2 C3 C4 C5
d. Average deformation gradient
reference subdomain points
vector v_0
$\Lambda_0 = [v_{01}, v_{02}, \dots, v_{0n}]$
deformed subdomain points
F_{avg}
$F_{avg}\Lambda_0 = \Lambda$
$F_{avg} = \Lambda\Lambda_0^T[\Lambda_0\Lambda_0^T]^{-1}$
vector v
$\Lambda = [v_1, v_2, \dots, v_n]$
IV
Visualize
a. Strain time series plots
b. Movies: full-field displacement
c. Movies: subdomain strains
a.
beat 0
strain Ecc
frame
b.
frame 32, beat 0
frame 40, beat 0
frame 50, beat 0
absolute displacement (pixels)
c.
frame 32, beat 0, Ecc
frame 40, beat 0, Ecc
frame 50, beat 0, Ecc
Ecc strain (image column axis)

Fig 3. Schematic illustration of the "MicroBundleCompute" computational framework. Two main inputs are required: I) a binary mask generated either automatically using the software or manually and consecutive movie frames of the beating microbundle. II) For tracking, (a) marker points are identified on the first frame and (b) tracked across all frames to identify individual beats and perform time segmentation. (c) This allows us to perform the analysis per beat and correct for the observed drift as discussed in the "Temporal segmentation" Section. (d) Finally, we save the row and column positions of the tracked markers per a single beat and use these saved outputs to compute full-field displacements and derive strain results. III) Post-processing functionalities include (a) rotation of the images and tracking results, (b) interpolation of the results at query points, and (c) segmentation into subdomains for which (d) average deformation gradients and subsequently, subdomain-averaged strains are calculated. IV) Finally, to visualize the results, the software outputs (a) time series plots per beat and movies of (b) full-field and (c) subdomain-averaged results.

https://doi.org/10.1371/journal.pone.0298863.g003

workflow. As a brief note, the software GitHub repository (https://github.com/HibaKob/MicroBundleCompute) contains instructions on how to install and run the code, detailed explanations of each main function, and a more thorough description of the formatting of output files.

The implementation of these methods is divided across four Python files: `create_tissue_mask`, `image_analysis`, `strain_analysis`, and `optional_preprocessing`. The bulk of core functionality is contained in `image_ analysis`, and all functions of the code are designed to be modular when possible such that they can be replaced in the future if a need arises. In addition, many of the specific post-processing steps are optional, and adaptation to compute additional quantities of interest should be straightforward. As we describe our pipeline, we will specify which Python file a given function is located in. Essential functions included in our pipeline are as follows, following the order illustrated in Fig 3.

Tracking region mask generation. Within `create_tissue_mask`, we provide two different options for creating a binary mask of the tissue region: manual or automatic (Fig 3I). At present, we provide 3 basic segmentation functions for automatic microbundle mask creation: 1) a straightforward threshold-based mask, 2) a threshold-based mask that is applied after a Sobel filter, and 3) a threshold-based mask that is applied to either the minimum or the maximum (specified by the user as an input) of all movie frames. We implement our tissue segmentation pipeline mainly based on the `threshold_otsu` and `sobel` functions provided within the `filters` module in scikit-image 0.19.3 Python library [88].

In addition to programming alternative automated mask generation functions, software users can provide an externally generated tissue mask (e.g., a manually generated mask) by including a file named "tissue_mask.txt" where the mask is a two-dimensional array in which the tissue domain is denoted by "1" and the background domain is denoted by "0" to allow for the analysis of domains that may fall outside the original scope of this endeavor.

Sparse optical flow algorithm for tracking. Within `image_analysis`, we provide all of the essential functions to automatically run the tracking algorithm illustrated in Fig 3II. Our tracking pipeline is built on OpenCV's [97] pyramidal implementation of the Lucas-Kanade sparse optical flow algorithm [52] where we identify markers as Shi-Tomasi "good features to track" corner points [98] that fall within the specified tissue mask. In brief, these corner points are points where a slight shift in location leads to "large" changes in intensity along both the horizontal and vertical axes. There are two key parameters required to tune OpenCV's `goodFeaturesToTrack` function: `qualityLevel`, a minimum score to measure if a feature can be tracked well, and `minDistance`, the minimum permitted Euclidean distance between two identified corners. We initialize `minDistance` = 3 and `qualityLevel` = 0.1. Then, we iteratively decrease `qualityLevel` until `coverage` > 40, where `coverage` is defined as the average number of pixels associated with each tracking point, for up to 15 iterations. As for `minDistance`, we define a `local_coverage` measure, which is the `coverage` computed on 20 × 20 pixel subdivisions. The `minDistance` is automatically incremented by 1 as long as the largest 3 `local_coverage` values are less than or equal to 50 and the number of iterations does not exceed 2. We note that we adjust `qualityLevel` and `minDistance` simultaneously.

For running OpenCV's [97] Lucas-Kanade optical flow [52] function (`calcOpticalFlowPyrLK`), we automatically tune the parameter `winSize`, which dictates the size of the integration window. Crucially, the window size in both horizontal and vertical directions, w_x and w_y, should be larger than the maximum tracked pixel motion between frames. To specify `winSize`, we adopt a pragmatic approach where we initialize `winSize` = 5, perform a preliminary tracking step, calculate the maximum absolute displacement, and compare its magnitude to the initial window size. If the calculated displacement is larger, we increase `winSize` by 5, and continue to iterate until the condition is met or the number of iterations exceeds 15. Critically, keeping `winSize` from being larger than necessary reduces error during tracking. In the remainder of this Section, we will describe our methods for leveraging this basic sparse optical flow algorithm to effectively and automatically analyze microbundle domains.

Temporal segmentation. After automatic identification of features and tracking algorithm parameters, we run a preliminary tracking step. Representative results of preliminary tracking are shown in Fig 3II.b as a plot of mean absolute displacement vs. frame number. We use these preliminary tracking results to perform temporal segmentation where we delineate individual tissue beats. To accomplish this, we use the SciPy signal processing library function `find_peaks` [99]. The `find_peaks` input parameters, `distance`, the minimum horizontal distance between neighbouring peaks, and `prominence`, minimum prominence for a perturbation to be recognized as a peak, are identified automatically. Specifically, we initialize `distance` and `prominence` to values of 20 and 0.1 respectively. Then `distance` is updated to take a value equal to 1.5× the horizontal distance separating two consecutive intersection points between the time series and its mean. For the `prominence` parameter, we keep it constant at a value of 0.1, which we found to be suitable for all the example videos that we have analyzed to date. After peaks are identified, we define valleys as the midpoints of two consecutive peaks. As such, to be able to identify a pair of valleys, or in other words a single beat, a minimum of 3 beats should be present in any movie to enable automatic analysis and an accurate approximation of beat period. Temporal segmentation into individual beats is then performed based on the temporal location of each valley.

Beyond the determination of microbundle time-related properties (e.g., beat period), temporal segmentation is an essential part of our pipeline as we use it to work around the tracked feature drift observed over the duration of the movie (see Fig 3II.b for an illustration of drift).

After identifying individual beats, we split, based on the first and last beat frame numbers, the main two arrays storing column (horizontal) and row (vertical) locations of marker points obtained during preliminary tracking into multiple arrays corresponding to each segmented beat. Likewise, instead of frame 0 being the reference configuration for the whole tracking duration, the fiducial marker positions in the first frame of each beat become the baseline for all future output calculations within the beat.

Optional rotation and interpolation. After the tracking step is complete, we include two optional features for post-processing: sample rotation and fiducial marker displacement interpolation. First, we include an option to rotate both the images and the tracking results based on a specified center of rotation and desired horizontal axis vector. The center of rotation and desired horizontal axis vector can be either specified manually or identified automatically based on the geometry of the tissue mask. As a brief note, rotation is performed after tracking as the process involves interpolation which can lead to loss of image resolution. Also, we automatically rotate the tissue domain before performing strain subdomain calculations to match the global row (vertical) and column (horizontal) coordinate system. The second optional feature, interpolation, allows the user to interpolate the tracking results returned at the automatically identified fiducial marker points to user-specified locations, on a structured grid for example. This step can be performed after tracking and optional rotation, and will output the interpolated displacement fields to specified sampling points for either visualization or downstream analysis.

Subdomain spatial segmentation. In contrast to standard approaches to analyzing cardiac microbundles [51], our approach is unique in that we compute full-field quantities of interest over the tissue domain. In order to better analyze these full-field results and reliably post-process displacement fields to compute strain, we perform spatial segmentation to define tissue subdomains over which we can report average strain quantities [100, 101]. This subdomain spatial segmentation is implemented in the `strain_analysis` file within "MicroBundleCompute." We provide two options to specify the subdomain extents defined as a rectangle: 1) automatic subdomain generation via clipping the input tissue mask or 2) manually providing subdomain extent coordinates. Given rectangular subdomain extents, we then delineate individual subdomain tiles by specifying either the target number of tiles in each column and row or by specifying the target tile dimensions in pixels. Representative subdomain segmentation results are illustrated in Fig 3III.c.

Strain computation. With these defined subdomain regions, we then compute the average deformation gradient $\mathbf{F}_{\mathbf{avg}}$ of each subdomain and use it to compute relevant strain metrics. As stated previously, we compute average subdomain strain rather than full-field strain due to: (1) the desire to reduce the influence of imaging artifacts and noise, and (2) increased ease of comparison between samples. Within each subdomain, we define the standard continuum mechanics deformation gradient $\mathbf{F}$ as follows:

$$\mathbf{F}d\mathbf{X} = d\mathbf{x} \tag{1}$$

where $\mathbf{F}$ maps a vector $d\mathbf{X}$ in the initial or reference configuration to its deformed configuration $d\mathbf{x}$ [102]. To apply this within the context of our tracking pipeline, we define a set of n vectors, $\mathbf{\Lambda_0}$, that connect each potential pair of fiducial markers that lie within the extents of each subdomain in the reference configuration. We define the reference configuration with respect to each cardiac tissue beat as the first frame in the segmented beat frame. Thus, $\mathbf{\Lambda_0}$ is defined in frames that represent the most relaxed tissue state. We then compute $\mathbf{\Lambda}$ following the same structure as $\mathbf{\Lambda_0}$ for each subsequent movie frame, where the updated fiducial marker positions capture the subdomain deformed configuration. With this definition, we can set up the over-determined system of equations:

$$\mathbf{F}_{\text{avg}}\mathbf{\Lambda}_0 = \mathbf{\Lambda} \quad \text{where} \quad \mathbf{\Lambda}_0 = [\mathbf{v}_{01}, \mathbf{v}_{02}, \ldots, \mathbf{v}_{0\mathbf{n}}] \quad \text{and} \quad \mathbf{\Lambda} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_\mathbf{n}] \tag{2}$$

where $\mathbf{F_{avg}}$ is a 2 × 2 matrix, and $\mathbf{\Lambda_0}$ and $\mathbf{\Lambda}$ are 2 × n matrices of vectors in the initial (reference) and current (deformed) configurations, respectively. Of note, when the initial and current frames are identical, $\mathbf{\Lambda_0} = \mathbf{\Lambda}$, $\mathbf{F_{avg}} = \mathbf{I}$, a 2 × 2 identity matrix. To solve this over-determined system, we can use the normal equation to find the best fit average deformation gradient as:

$$\mathbf{F}_{\text{avg}} = \mathbf{\Lambda}\mathbf{\Lambda}_0^{\text{T}}[\mathbf{\Lambda}_0\mathbf{\Lambda}_0^{\text{T}}]^{-1}\,. \tag{3}$$

We schematically illustrate our method to compute the mean deformation gradient in Fig 3III.d. With the computed $\mathbf{F_{avg}}$, finding the average Green-Lagrange strain tensor is straightforward:

$$\mathbf{E}_{\text{avg}} = \frac{1}{2}\left(\mathbf{C}_{\text{avg}} - \mathbf{I}\right) \quad \text{where} \quad \mathbf{C}_{\text{avg}} = \mathbf{F}_{\text{avg}}^{\text{T}}\mathbf{F}_{\text{avg}} \tag{4}$$

and we can then compute strain on a per subdomain per beat basis to obtain subdomain time series results for $\mathbf{F_{avg}}$ and subsequently $\mathbf{E_{avg}}$.

Data structure preparation. Tracking all identified fiducial markers for an extended number of frames produces a large quantity of output data for each movie. Thus, we selectively save output results such that they are both comprehensible and easily accessible for downstream data analysis. First, we save information regarding the column (horizontal) and row (vertical) positions of the tracked marker points per beat. Specifically, we store one row-position text file and one col-position text file for each beat formatted as a $M \times N$ array where M is the number of markers that were tracked and N is the number of frames in the beat. Similarly, we output mean deformation gradient results as text files saving the column and row positions of the center for each subdomain and the 4 components of the 2 × 2 mean deformation gradient per subdomain per beat.

Data analysis, key metrics, and visualization. In Fig 3IV, we show key output metrics and their visualizations. In brief, we provide tools to visualize full-field displacement and average subdomain strain, and provide key quantities of interest such as maximum strain, beat period, and synchrony. In all cases, we build on the popular matplotlib package [103] for producing all visualizations. In addition to the metrics directly enabled by our novel pipeline, we provide other commonly pursued relevant outputs including beat frequency, beat mean amplitude, and tissue width at the domain center. To convert the numerical outputs from dimensions of pixels and frame numbers to physical units, the user can specify: 1) frames per second and 2) the length scale in units of μm/pixel.

Quality checking and rejection of unsuitable examples. In our extensive experience testing our code on different synthetic and real examples, we have identified three main instances that negatively influence the fidelity of our outputs and decrease our confidence in analysis results:

1. blurred input movie frames that prevent effective identification of corner points for the tracking described in the "Sparse optical flow algorithm for tracking" Section.
2. movies that start from a contracted tissue position that confounds the temporal segmentation step described in the "Temporal segmentation" Section.
3. movies where all displacement is on *sub*-pixel length scales lead to large relative errors given our choice of tracking algorithm.

The specific influence of these conditions are explored in S1 Appendix. To address these conditions, we provide both warnings in the code when we detect that these conditions arise and functions to correct these scenarios (e.g., removing initial movie frames to address case 2).

Note on pillar tracking. In the broader cardiac microbundle and microtissue literature [33, 39], pillar force and tissue stress are two commonly computed metrics. Broadly speaking, pillar force is computed by tracking pillar displacement and converting displacement to force via cantilever beam equations where the pillars are treated as beams with known elastic modulus and geometry [104]. Stress is then computed from pillar force by dividing force by tissue cross sectional area where tissue width is measured from the in-plane images [34] and tissue depth is assumed based on typical values observed via three dimensional imaging modalities [105]. We are able to adapt our framework to readily track pillar displacement by simply specifying pillar regions as the area to track rather than the tissue domain. Due to lack of novelty, we do not emphasize this functionality in this paper. However, we do provide additional details on this topic in S3 Appendix.

Results and discussion

In this Section, we show a summary of our main results. In the "Synthetic data examples" Section, we present representative validation examples from validating our pipeline on an example from the synthetic dataset of "Type 1" with known ground truth behavior, and in the "Experimental data examples" Section, we share examples of implementing our computational framework on 3 different experimental platforms (i.e., 3 different data types as explained in the "Experimental data" Section). As a brief note, we provide a more comprehensive set of results in S1 Appendix on validation, S2 Appendix on additional real data examples, and S3 Appendix on pillar tracking.

Synthetic data examples

Here, we briefly summarize the results of our validation studies. In Fig 4, we show the performance of our pipeline on a single representative synthetic example ST_1. In S1 Appendix, we provide a comprehensive summary of all 416 synthetic validation examples based on data of "Type 1" as well as a single synthetic example of "Type 2" data. In our validation studies, we compare "MicroBundleCompute" displacement and strain outputs against the known ground truth for synthetic data as originally generated (without noise) and for the same examples with added Perlin noise of different magnitude ratios and octaves. As a brief note, other types of noise such as shot noise [106] would be appropriate alternatives to Perlin noise for the purpose of this investigation. In preliminary studies, we found that Perlin noise with a range of magnitudes and octaves was a sufficiently general choice to provide a robust challenge for our framework. The code required to reproduce all synthetic data examples of "Type 1" can be found on GitHub (https://github.com/HibaKob/SyntheticMicroBundle).

In Fig 4a, we specify the synthetic data domain for which the validation studies were performed, a 90 × 90 domain warped with FEA-informed displacements as briefly described in the "Synthetic data generation" Section and presented in more details in S1 Appendix. In Fig

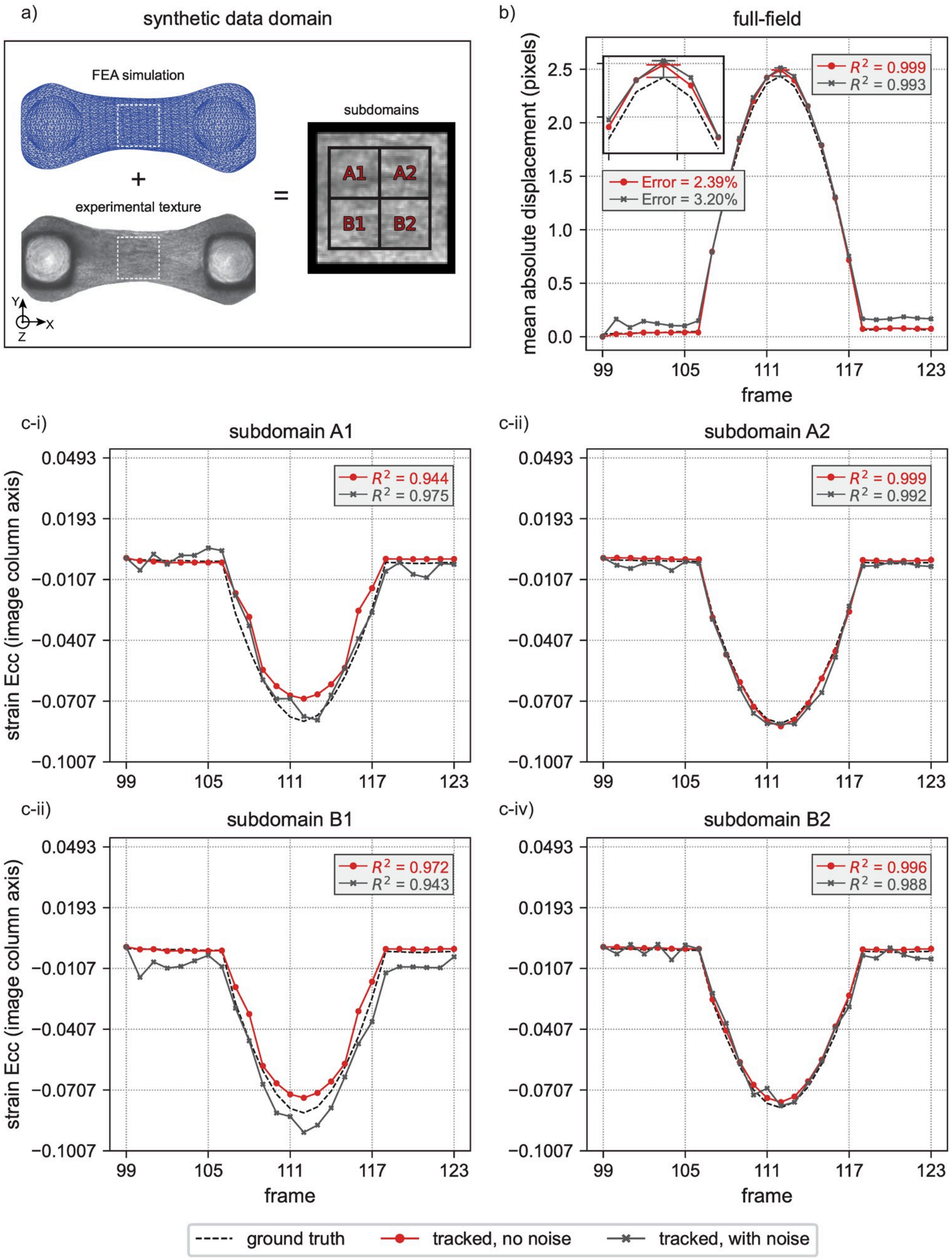

Fig 4. Validation against a single synthetic example. Validation results against synthetic example ST_1 in its original (no noise) state as well as after the addition of Perlin noise (magnitude ratio of 12% and octaves of 40): (a) a schematic representation of the synthetic data domain and subdomain divisions for strain calculations; (b) tracked and FEA-extracted (ground truth) mean absolute displacement of the full-field data for a single beat (beat 4), where maximum error for synthetic data with noise is observed; (c-i)–(c-iv) tracked and FEA-extracted (ground truth) subdomain-averaged E_{cc} strain for beat 4 for each subdomain. Overall, the R^2 values indicate relatively good agreement between the tracking output and ground truth data. For more information on validation with synthetic data, refer to S1 Appendix.

https://doi.org/10.1371/journal.pone.0298863.g004

4b, we plot the mean absolute displacement with respect to frame number for beat 4, the beat with the maximum error at the peak displacement (full contraction) for the synthetic example with Perlin noise of magnitude ratio of 12% and octaves of 40. We specifically highlight these Perlin noise parameters because they mimic, to a great extent, noise artifacts found in the real experimental data. A direct comparison between both tracked and ground truth mean absolute displacements per beat revealed that the percentage error at peak displacement is less than 2.5% for the example shown in Fig 4b, with the addition of Perlin noise slightly raising this error to 3.2%. In addition, we assessed the ability of our computational framework to "predict" the mean absolute displacement by calculating the coefficient of determination (R^2) and found good agreement. Overall, considering all validation examples in S1 Appendix prior to the addition of Perlin noise, peak mean displacement errors were found to be less than 7% for *super*-pixel displacements (Table S1 _1 and Fig S1_5 in S1 Appendix) whereas *sub*-pixel displacements had higher errors that reach up to 15% (Table S1_1 and Fig S1_6 in S1 Appendix) with a minimum R^2 value of 0.939. Furthermore, it is evident that the addition of Perlin noise produces higher shifts within the tracked outputs, with higher errors reported for higher magnitude ratios and lower octaves (Figs S1_23 and S1_24 in S1 Appendix). Specifically, as shown in Table S1_4 and Fig S1_23 in S1 Appendix, the percentage error at the peak mean absolute displacement increases to around 14.4% for *super*-pixel displacement examples. For the *sub*-pixel displacement examples, mean absolute displacement errors become unreasonably high in the majority of the synthetic examples and are on the order of $10^3 - 10^4$ (Fig S1_24 in S1 Appendix). And, in some of these *sub*-pixel displacement examples where the added Perlin noise is unrealistically extreme (higher magnitude ratios and lower octaves), the code fails to produce meaningful outputs as indicated by missing data points in Fig S1_24 in S1 Appendix.

In Fig 4ci–4civ, we show the error on the column-column direction (i.e., horizontal) Green-Lagrange strain (E_{cc}) per subdomain for beat 4. We refer to this direction as the "column-column" direction to be consistent with the row (vertical) and column (horizontal) directions defined by our input images. Here, the reported R^2 values reveal that the minimum value of 0.943 occurs in subdomain `B1`. And, in general, these plots indicate that strain magnitudes tend to be underestimated. Furthermore, errors on E_{cc} follow the observed trend for displacement errors, where synthetic examples of *sub*-pixel displacements exhibit higher errors with some cases having negative R^2 values (Tables S1_2 and S1_3 and Fig S1_8 in S1 Appendix).

In S1 Appendix, we share the validation results for all synthetic examples. We note that strain outputs are sensitive to subdomain divisions, specifically subdomain size. Determining a suitable subdomain size is a delicate process that is governed by two main opposing factors. The subdomain size should be large enough such that each subdivision contains an appropriate number of automatically identified fiducial markers to ensure that the computations are less sensitive to noise. On the contrary, the subdomain size should be small enough to avoid the loss or reduction of information due to averaging the heterogeneous deformation, or put more explicitly, obtaining attenuated or zero strain values due to lumping regions that are experiencing opposing deformations, for example extension versus compression. In general, based on our comprehensive experience with implementing the code on a number of synthetic and real examples, we recommend a subdomain side length that is between 30 and 40 pixels. From a methodological perspective, we propose that the user observes the displacement field and avoids having subdomains that span regions where the change in displacement flips sign. We recommend that manual examination of subdomain size be carried out once per dataset, for example, select a single movie from a batch of 100 to confirm that the subdomain division is appropriate. In the future, we plan to investigate the approach adopted in [76, 77] where strains are directly informed from affine warping functions optimized via the Lucas-Kanade algorithm [75] without computing displacement fields.

As described in this Section and in S1 Appendix, we have performed extensive validation studies for the "MicroBundleCompute" computational framework against a total of 416 synthetic examples of "Type 1" (16 examples generated under baseline conditions and the remaining 400 created by adversely altering the original 16 examples via the addition of Perlin noise) and a single synthetic example of "Type 2" with known ground truth. These validation studies: 1) corroborate the output of the software against FEA-labelled data, 2) test its performance against realistic synthetic examples, and 3) evaluate its robustness against challenging examples with excessive noise artifacts. The obtained results reveal that for *super*-pixel displacements, our software is quite robust to all tested cases with relatively low magnitudes and high octaves, or in other words, when the Perlin noise patterns resemble speckle noise rather than pronounced textures (see Fig S1_4 in S1 Appendix). However, for examples with entirely *sub*-pixel displacements, the performance of the software degrades. In summary, "MicroBundle-Compute" breaks down when the synthetic data has noise artifacts that appear similar to the original texture, and for *sub*-pixel displacements. Yet, given that real experimental examples of beating microbundles generally produce displacements that exceed a single pixel and that Perlin noise examples with higher octaves more faithfully represent naturally occurring noise in real microbundle data than lower values, we anticipate that "MicroBundleCompute" will output reliable mechanical metrics in real experimental settings on condition that the natural contrast of the microbundle textures is visibly present and in focus in the time-lapse videos.

Experimental data examples

We provide here a summary of implementing "MicroBundleCompute" on the experimental dataset described in the "Experimental data" Section and S2 Appendix, and shared on Dryad (https://doi.org/10.5061/dryad.5x69p8d8g) under a Creative Commons CC0 1.0 Universal Public Domain Dedication. Specifically, the "Microbundle Time-lapse Dataset" [96] contains all raw videos in ".tif" format for the 24 experimental time-lapse images of beating cardiac microbundles, 23 of which are brightfield videos, while the remaining single example is a phase contrast video. Besides the raw videos and the experimental metadata describing the conditions under which they were obtained, we include the tissue mask used for each example, whether generated automatically via our computational pipeline or manually via tracing in ImageJ [78]. These time-lapse videos and masks were used to generate the results shown here and in S2 Appendix. We note briefly that it is only possible to develop automatic mask segmentation functions for examples where there is imaging consistency and when we have an ample number of examples to identify a pattern. Future extensions of this framework will include automatic mask functions tailored to specific experimental needs.

In Fig 5, we show visualizations of output results generated by running "MicroBundleCompute" on 3 experimental examples, each from a different data type. Of note, we only visualize 3 of many potential software outputs: (1) full-field absolute displacement, (2) spatially distributed subdomain-averaged Green-Lagrange strain E_{cc} (automatically rotated to align with the column-column horizontal direction and plotted at beat 0 strain peak), and (3) time series plot of E_{cc} strains with respect to beat 0 frames. These are representative examples from the comprehensive list of outputs described in the software GitHub repository (https://github.com/HibaKob/MicroBundleCompute). While displacement and strain visualizations reveal spatial information on tissue heterogeneous behavior and spatial contraction patterns, time series strain plots highlight this spatial synchrony (or lack of synchrony) of subdomain contraction across each beat.

To complement the results in Fig 5, we show, in S2 Appendix, representative results obtained for all 24 real examples following the same format. Critically, all of the results presented in the supplementary document, for all 3 of the data types, were obtained without the need for any parameter tuning. As is evident in Fig S2_1 in S2 Appendix, we can group data from "Type 1" into 2 different categories: 1) examples 1–6 which constitute relatively challenging examples to the code since the imaging brightness/contrast is not optimized to fully

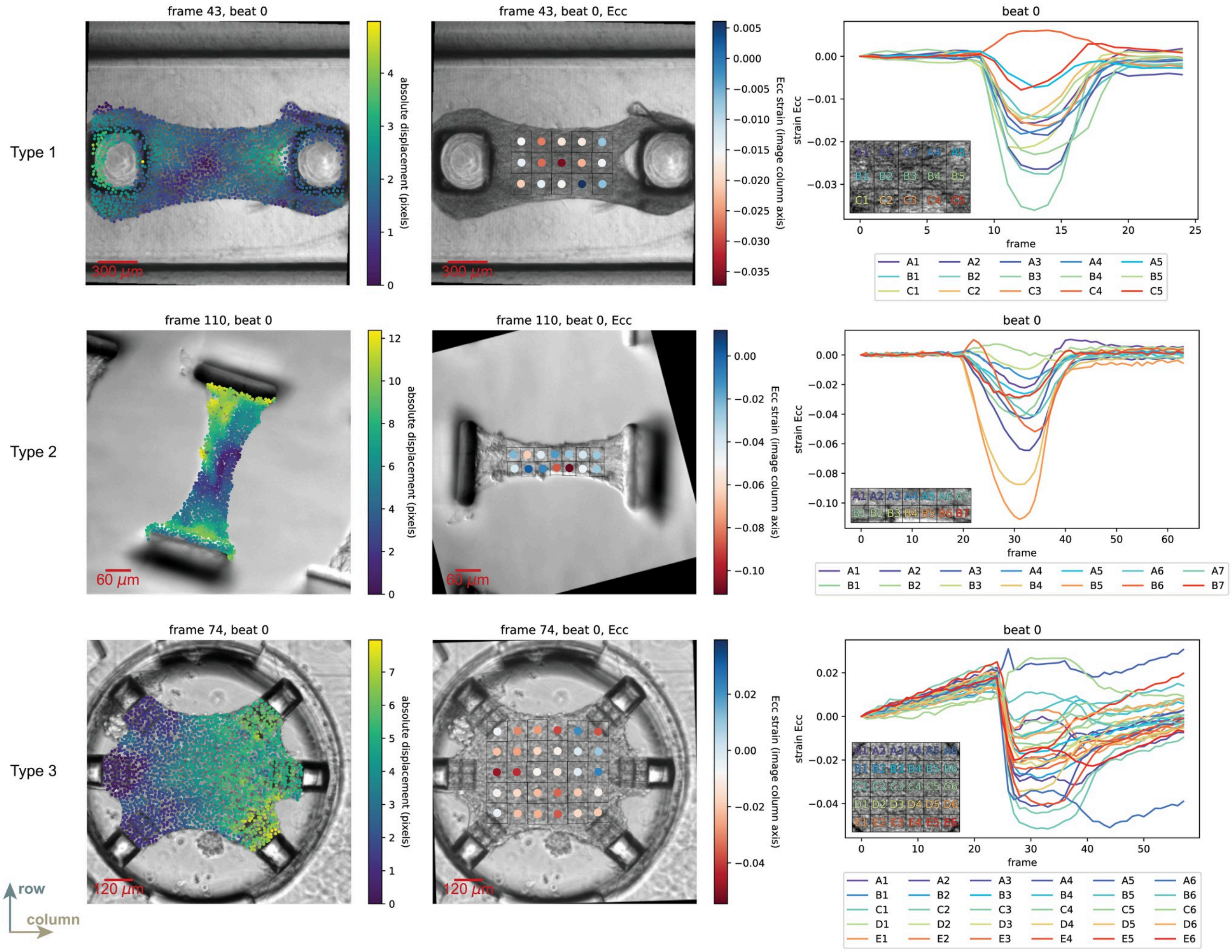

Fig 5. Examples of "MicroBundleCompute" outputs. We show sample outputs of the code when run on 3 examples from the 3 different data types, specifically Example 9 from "Type 1," Example 5 from "Type 2," and Example 1 from "Type 3" as shared on Dryad [96]. Note that the strain output for "Type 2" data is automatically rotated such that E_{cc} aligns with the column-column direction. Additionally, the "Type 3" example shows an actuated microbundle at 0.5 Hz by applying sawtooth pressure waves with ∼—6 kPa peak amplitude (equivalent to ∼ 2.5% strain) on the right side using a microfluidic pump (Elveflow OB1), and hence, the considerable discrepancy of the E_{cc} time series plot from 0 at the end of beat 0.

https://doi.org/10.1371/journal.pone.0298863.g005

accentuate the natural tissue texture, and 2) examples 7–11 where a texture is clearly visible in the microbundle region, making them a set of optimal examples for running the code. Despite the fact that "Type 1" data comprises the most consistent and well studied experimental platform, the full-field absolute displacement plots reveal wide variation in spatially distributed behavior. However, magnitude ranges are relatively consistent across samples with mean absolute displacement values at the peaks varying between 1.2 and 2.3 pixels (4.8 and 9.2 μm). We anticipate that this observation might be attributed to variations in pillar-tissue attachment behavior across different examples (see Fig S3_1 in S3 Appendix). However, further investigations are required to develop a systematic way to assess the effect of this pillar-tissue interaction on microbundle contractility and analyze the extracted metrics in this context.

Similar spatial heterogeneity is observed for the "Type 2" samples (Fig S2_2 in S2 Appendix), where examples 1–5, which were prepared under the same experimental conditions (soft, aligned matrix), exhibit discrepant displacement distributions but fairly consistent magnitude ranges (mean absolute displacement values at the maximum contraction vary between 5.0 and 7.9 pixels or equivalently, 4.54 and 7.17 μm). For examples 6 and 7, which are prepared with stiff matrices, the results reveal lower contraction magnitudes where an aligned stiff matrix (example 6) shows higher contractions than the randomly distributed one (example 7) [94]. Of note, by examining all results of cardiac microbundle "Type 1" and "Type 2" data, a noticeable diagonal contraction pattern is apparent, especially in examples 2, 7, 9, and 10 from "Type 1" (Fig S2_1 in S2 Appendix) and examples 2, 3, and 5 from "Type 2" (Fig S2_2 in S2 Appendix). This observation, enabled by full-field tracking, indicates that future study to investigate the association between microbundle contraction, fiber alignment, and emergent load paths between the pillars, would be meaningful future work.

The diversity of "Type 3" experimental data prevents direct comparison between samples. However, the time series strain visualizations (Fig S2_3 in S2 Appendix) reveal that cardiac microbundles grown on these experimental constructs typically contract more synchronously than microbundles of "Type 1" and "Type 2," where the strain time series plots in Figs S2_1 and S2_2 in S2 Appendix respectively, show aspects of temporal heterogeneity for which peak contractions do not occur at the same frame within all subdomains per beat. Finally, the visualized average subdomain E_{cc} strains, as well as the remaining two strain components (row-row direction Green-Lagrange strain E_{rr} and column-row direction Green-Lagrange strain E_{cr}) that are computed and saved but not included in the set of representative outputs that we visualize here, give insight about the regions within the microbundle that are contracting or bulging in a given direction across each beat.

As we mention in the "Note on pillar tracking" Section and describe in more detail in S3 Appendix, it is straightforward to implement our computational framework to track pillar displacements, adding to the versatility of the "MicroBundleCompute" software framework. In Fig 6, we show, side by side, the outputs obtained via pillar tracking (Fig 6a-i and 6b-i) and via tracking the entire tissue domain (Fig 6a-ii, 6a-iii and 6b-ii, 6b-iii) for an example from "Type 1" and "Type 2" data respectively. Pillar tracking enables the calculation of an average absolute or directional pillar displacement based tissue force, which can be used to infer an average tissue stress given approximate tissue width and thickness as described in more detail in S3 Appendix. On the other hand, tissue tracking reveals abundant information regarding the inherent heterogeneous nature of microbundle beating. For example, full-field displacement fields (Fig 6a-ii and 6b-ii) show regions where maximum or minimum tissue contractions are taking place. Furthermore, while subdomain-averaged strains also underline the spatial heterogeneity of the tissue contractions, visualizing them with respect to time (or frame number) reveals the nature of temporal synchrony across the different subdivisions of the beating microbundle (Fig 6a-iii and 6b-iii). We note that a similar comparison can be carried out for all "Type 1" and "Type 2" data for which pillar tracking results are included in S3 Appendix, while tissue tracking results are shared in S2 Appendix.

Finally, we include in Fig 7 an example of "Type 3" data which clearly indicates that the microbundle is experiencing positive E_{cc} strains at the center, specifically in subdomains A2 and A3 as indicated in the inset legend on the lower left corner of the strain time series plot. According to Wang et al. [107], this observation suggests that a necking instability is forming on this tissue over time, which is also supported by the thinning tissue width at the center. Within this context, being able to extract and examine strain distributions and magnitudes enables further investigations into understanding and determining the factors, such as pillar stiffness and ECM density [107], that produce microbundles that are more stable against necking.

Based on the results shown in this Section and in S2 Appendix, conventional metrics comprising tissue force and tissue stress obtained via basic pillar tracking offer insightful yet

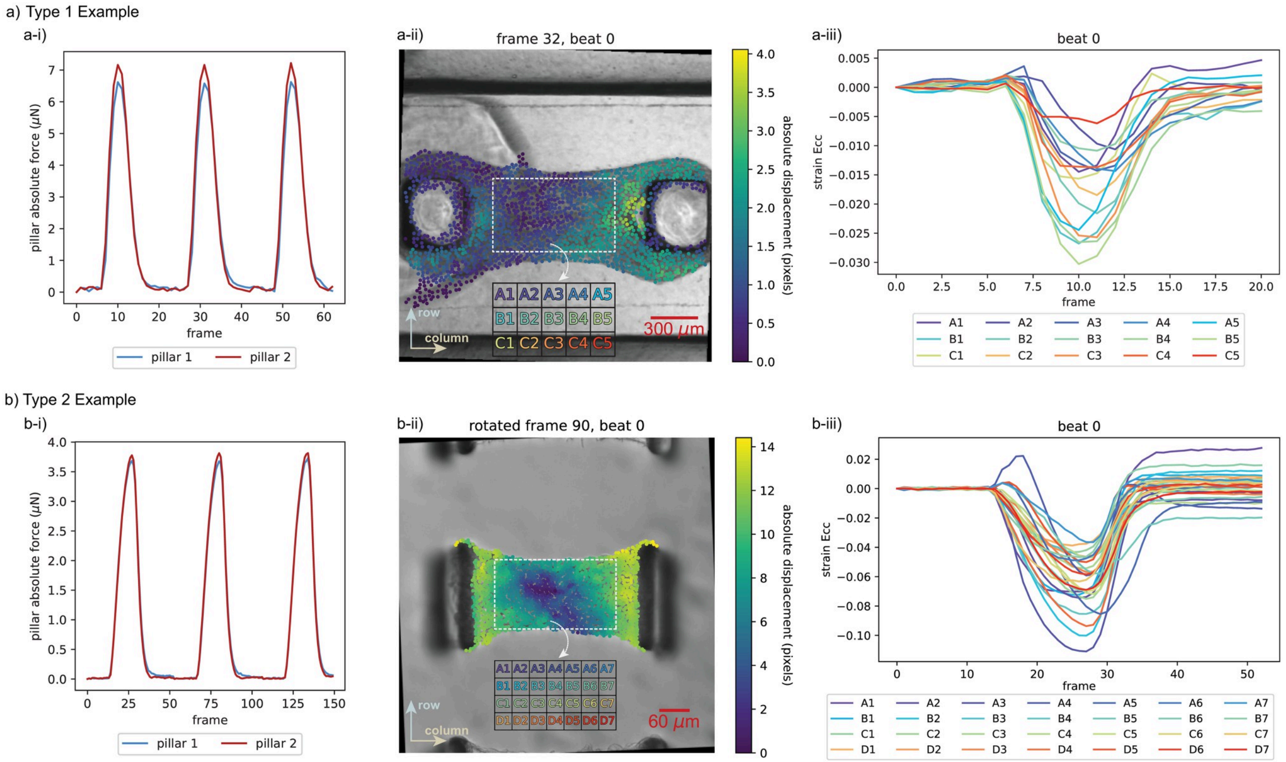

Fig 6. Direct comparison of pillar tracking and tissue tracking. Implementing the pillar tracking functionality of "MicroBundleCompute" on (a) Example 10 from "Type 1" data, (a-i) we obtain the pillar displacement based tissue force to describe the microbundle beating behavior, whereas full-field tissue tracking reveals the heterogeneous (a-ii) full displacement field as well as (a-iii) subdomain-averaged strains computed within the region marked by the dashed white box in (a-ii) where marker points within this region only are considered. In (b) we show the same outputs for Example 3 from "Type 2." We note that, for this example, we show the rotated displacement output to be consistent with the subdomain segmentation orientation. To view the original non-rotated displacement results, refer to Fig S2_2 in S2 Appendix.

https://doi.org/10.1371/journal.pone.0298863.g006

lumped information about the microbundle behavior. Complemented with reliable and reproducible full-field data, such as displacement distributions and subdomain-averaged strains, as in [53] for example, these metrics become more useful to assessing the highly heterogeneous cardiac tissue behavior and understanding the complex underlying mechanisms driving this behavior. Furthermore, this spatial information would allow us to study injury

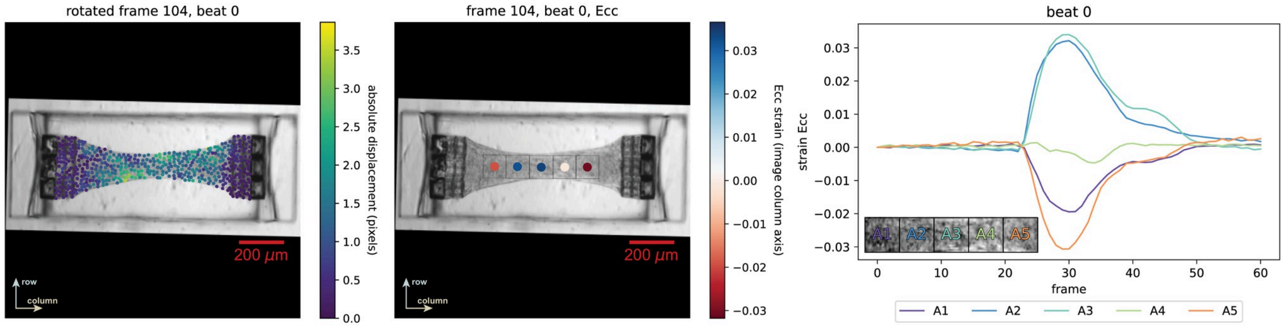

Fig 7. Outputs of "MicroBundleCompute" run on Example 6 from "Type 3" reveal regions of tissue extension at the center. We briefly note that for displacement outputs, rotating the results is optional, while for strain outputs, the frames are rotated such that the major axis of the microbundle automatically aligns with the column (i.e., horizontal) axis. To view the original orientation of this example, refer to Fig S2_3 in S2 Appendix.

https://doi.org/10.1371/journal.pone.0298863.g007

models of tissue [53], as well as further investigate mechanical observations from pillar tracking, for example, the unbalanced pillar forces noted in Figs S3_1 and S3_2 in S3 Appendix. Finally, with sufficient collected data, this approach would enable the development of appropriate statistical models of the whole tissue. These endeavors are planned as part of our future work.

Conclusion

In this work, we describe our approach to creating a computational tool for analyzing brightfield and phase contrast movies of beating microbundles. In brief, we describe our process for converting movies of beating microbundles to meaningful quantitative metrics, validating our approach against synthetically generated data, and testing it on a diverse pool of real experimental examples. In addition to ease of use, limited requirements for user intervention, relatively short run time, and no parameter tuning, "MicroBundleCompute" is easy to implement out of the box on new experimental datasets. Because it relies solely on the natural contrast of brightfield and/or phase contrast microbundle images and the resulting intensity gradients, it is also straightforward to integrate with existing experimental workflows. To enable broad adoption, we share the software under open-source license and look forward to receiving feedback from different users on how to adapt the code to tailor to their specific needs and enhance the overall user experience.

Looking forward, we aim to constantly improve the code. Future extensions will include automated quality control and pre-processing of input data, as well as enhancements to current functionality such as automatic adjustment of input movies that do not start from a fully relaxed frame. As viable alternatives arise, we also plan to benchmark it against available cardiac microbundle tissue analysis tools. In addition, we plan to continue developing our pipeline to address alternative quantitative metrics and different imaging modalities such as calcium imaging [108] and integrate these results with structural information such as sarcomere geometry and alignment [101, 109]. From the results shown in the "Experimental data examples" Section as well as S2 and S3 Appendices, it is also clear that there is significant variation across individual microbundle behavior both within and across testbeds. One key motivation for applying the "MicroBundleCompute" framework to these data moving forward is that it will make it possible to better understand and analyze this heterogeneity. In addition, extracting comparable mechanical metrics reliably and reproducibly across different testbeds allows for the identification of the favorable configurations and conditions that promote hiPSC-CM based tissue maturation, and ultimately, converge to an optimum system. Overall, our intention is for other researchers to directly benefit from disseminating this work. As a final note, here we demonstrate the utility and functionality of "MicroBundleCompute" for a particular highly used engineered cardiac tissue format: cardiac microbundles. In concurrent work [110], we leverage the fundamental core of this computational framework and make some minor modifications and extensions to extract mechanical metrics from actuated 2D muscle sheets. Looking forward, we will continue to generalize our framework to provide non-invasive, label-free, and high-throughput tools to facilitate contraction measurements across different engineered contractile tissue platforms to benefit the tissue engineering research community.

Supporting information

S1 Appendix. Additional details on pipeline validation against synthetic data. Generating synthetic examples of both "Type 1" and "Type 2" as well as validating "MicroBundleCompute" against these synthetic data in their original form and with added Perlin noise are

described and discussed in more detail. **Table S1_1. Summary of validating tracked displacement against synthetic data of "Type 1" without any added Perlin noise.** Tracked mean absolute displacement is compared against a known ground truth. **Table S1_2. Summary of validating E_{cc} strain against synthetic data of "Type 1" without any added Perlin noise.** Computed E_{cc} strain from tracked displacement data is compared against a known ground truth. **Table S1_3. Tabulated summary of validating E_{cc} strain outputs of "MicroBundleCompute" against all synthetic data of "Type 1" without any added Perlin noise.** Computed E_{cc} strain from tracked displacement data is compared against a known ground truth. **Table S1_4. Summary of validating tracked displacement against synthetic data of "Type 1" based on homogeneous activation with added Perlin noise.** Tracked mean absolute displacement is compared against a known ground truth and the "best" and "worst" results are reported. **Table S1_5. Summary of validating E_{cc} strain from tracked displacement data against synthetic data of "Type 1" based on homogeneous activation with added Perlin noise.** Computed E_{cc} strain from tracked displacement data is compared against a known ground truth and the "best," "average," and "worst" findings are reported. **Fig S1_1. Typical microbundle dimensions of "Type 1."** Schematic representation of the "Type 1" microbundle mesh implemented in our Finite Element simulations. **Fig S1_2. Convergence study of the grid size used to warp the microbundle textures to generate synthetic data.** Both *x* and *y* positions of the mesh cell centers are compared to a ground truth with respect to grid size. **Fig S1_3. A tabulated summary of the implemented conditions to obtain 400 noisy synthetic examples.** Main differences between the generated synthetic examples lie in the type of activation (homogeneous versus heterogeneous) used within the FEA model and the specific parameters of the added Perlin noise. **Fig S1_4. The effect of adding Perlin noise of different magnitude ratios and number of octaves on the tracked output of "MicroBundleCompute."** Real data contains inherent noise that a range of Perlin noise parameters can mimic its effects. **Fig S1_5. Displacement validation against examples based on homogeneous activation.** A detailed comparison is shown between tracked mean absolute displacement and a known ground truth for 8 different synthetic examples. **Fig S1_6. Displacement validation against examples based on heterogeneous activation.** A detailed comparison is shown between tracked mean absolute displacement and a known ground truth for 8 different synthetic examples. **Figs S1_7–S1_22. Detailed validation results of E_{cc} strain for all 16 synthetic examples without the addition of Perlin noise.** Computed E_{cc} strain from tracked displacements are compared to a known ground truth within each subdomain. **Fig S1_23. Effect of adding Perlin noise to synthetic data of "Type 1" based on FE simulations with homogeneous activation.** Perlin noise generated for 5 different magnitude ratios and 5 different octaves are added to synthetic data to test the performance of "MicroBundleCompute" in tracking displacements that exceed a single pixel. **Fig S1_24. Effect of adding Perlin noise to synthetic data of "Type 1" based on FE simulations with heterogeneous activation.** Perlin noise generated for 5 different magnitude ratios and 5 different octaves are added to synthetic data to test the performance of "MicroBundleCompute" in tracking *sub*-pixel displacements. **Fig S1_25. Displacement validation against synthetic data of "Type 2."** Tracked displacements in *X* and *Y* are compared to a known ground truth. **Fig S1_26. Strain validation against synthetic data of "Type 2."** Computed E_{cc} strains based on tracked displacements are compared to a known ground truth. **Fig S1_27. Validation via manual tracking against "Type 2" data.** Selected points are manually tracked and the displacements in *X* and *Y* are compared to the equivalent outputs obtained by running "MicroBundleCompute".
(PDF)

S2 Appendix. Testing "MicroBundleCompute" with additional examples of real data. The results of implementing "MicroBundleCompute" on a total of 24 experimental time-lapse images of cardiac microbundles, 11 examples of "Type 1," 7 of "Type 2," and 6 of "Type 3." **Table S2_1. A summary of the experimental conditions associated with each example movie.** The details of experimental conditions include image acquisition parameters as well as pillar stiffness values for "Type 1" and "Type 2" data. **Table S2_2. Additional details for each example of "Type 1" data.** These details include example-specific information, code implementation details, and subdomain segmentation parameters. **Table S2_3. Additional details for each example of "Type 2" data.** These details include example-specific information, code implementation details, and subdomain segmentation parameters. **Table S2_4. Additional details for each example of "Type 3" data.** These details include example-specific information and subdomain segmentation parameters. **Fig S2_1. Example outputs of "MicroBundle-Compute" run on "Type 1" experimental data.** The provided outputs include full-field mean absolute displacement and subdomain-averaged Green-Lagrange E_{cc} strain at the first tracked peak, as well as a time series plot of E_{cc} strain for the first tracked beat. **Fig S2_2. Example outputs of "MicroBundleCompute" run on "Type 2" experimental data.** The provided outputs include full-field mean absolute displacement and subdomain-averaged Green-Lagrange E_{cc} strain at the first tracked peak, as well as a time series plot of E_{cc} strain for the first tracked beat. **Fig S2_3. Example outputs of "MicroBundleCompute" run on "Type 3" experimental data.** The provided outputs include full-field mean absolute displacement and subdomain-averaged Green-Lagrange E_{cc} strain at the first tracked peak, as well as a time series plot of E_{cc} strain for the first tracked beat.
(PDF)

S3 Appendix. Additional details on pillar tracking. The implementation of the pillar tracking pipeline within "MicroBundleCompute" is explained in more details and demonstrated on 11 examples of "Type 1" and 7 examples of "Type 2." **Fig S3_1. Pillar tracking on "Type 1" examples.** Depicted results include pillar absolute force (μN) obtained on both pillars. **Fig S3_2. Pillar tracking on "Type 2" examples.** Depicted results include pillar absolute force (μN) obtained on both pillars.
(PDF)

S1 Movie. Synthetic example based on "Type 2" data. This movie provides the synthetic example generated based on "Type 2" data as described in S1 Appendix.
(TIF)

S2 Movie. Movie of tracked absolute displacement for Example 9 from "Type 1" data. This movie is provided as a supplement to Fig S2_1, Example 9, in S2 Appendix.
(MP4)

S3 Movie. Movie of tracked E_{cc} strain for Example 9 from "Type 1" data. This movie is provided as a supplement to Fig S2_1, Example 9, in S2 Appendix.
(MP4)

S4 Movie. Movie of tracked absolute displacement for Example 10 from "Type 1" data. This movie is provided as a supplement to Fig S2_1, Example 10, in S2 Appendix.
(MP4)

S5 Movie. Movie of tracked E_{cc} strain for Example 10 from "Type 1" data. This movie is provided as a supplement to Fig S2_1, Example 10, in S2 Appendix.
(MP4)

S6 Movie. Movie of tracked absolute displacement for Example 3 from "Type 2" data. This movie is provided as a supplement to Fig S2_2, Example 3, in S2 Appendix.
(MP4)

S7 Movie. Movie of tracked E_{cc} strain for Example 3 from "Type 2" data. This movie is provided as a supplement to Fig S2_2, Example 3, in S2 Appendix.
(MP4)

S8 Movie. Movie of tracked absolute displacement for Example 5 from "Type 2" data. This movie is provided as a supplement to Fig S2_2, Example 5, in S2 Appendix.
(MP4)

S9 Movie. Movie of tracked E_{cc} strain for Example 5 from "Type 2" data. This movie is provided as a supplement to Fig S2_2, Example 5, in S2 Appendix.
(MP4)

S10 Movie. Movie of tracked absolute displacement for Example 1 from "Type 3" data. This movie is provided as a supplement to Fig S2_3, Example 1, in S2 Appendix.
(MP4)

S11 Movie. Movie of tracked E_{cc} strain for Example 1 from "Type 3" data. This movie is provided as a supplement to Fig S2_3, Example 1, in S2 Appendix.
(MP4)

S12 Movie. Movie of tracked absolute displacement for Example 6 from "Type 3" data. This movie is provided as a supplement to Fig S2_3, Example 6, in S2 Appendix.
(MP4)

S13 Movie. Movie of tracked E_{cc} strain for Example 6 from "Type 3" data. This movie is provided as a supplement to Fig S2_3, Example 6, in S2 Appendix.
(MP4)

Acknowledgments

We gratefully acknowledge the collaborative opportunities facilitated by the CELL-MET Engineering Research Center. We especially thank the administrative team at CELL-MET for coordinating the Research Experiences for Undergraduates (REU) program. We would also like to acknowledge the staff at Boston University libraries for providing advice regarding data dissemination practices.

Author Contributions

Conceptualization: Hiba Kobeissi, Emma Lejeune.

Data curation: Hiba Kobeissi, Javiera Jilberto, M. Çağatay Karakan, Xining Gao, Samuel J. DePalma, Shoshana L. Das, Lani Quach, Emma Lejeune.

Formal analysis: Hiba Kobeissi, Emma Lejeune.

Funding acquisition: Brendon M. Baker, Christopher S. Chen, David Nordsletten, Emma Lejeune.

Investigation: Hiba Kobeissi, Emma Lejeune.

Methodology: Hiba Kobeissi, Javiera Jilberto, M. Çağatay Karakan, Xining Gao, Samuel J. DePalma, Shoshana L. Das, Emma Lejeune.

Project administration: Hiba Kobeissi, Emma Lejeune.

Resources: Brendon M. Baker, Christopher S. Chen, David Nordsletten, Emma Lejeune.

Software: Hiba Kobeissi, Javiera Jilberto, Jonathan Urquia, Emma Lejeune.

Supervision: Brendon M. Baker, Christopher S. Chen, David Nordsletten, Emma Lejeune.

Validation: Hiba Kobeissi, Javiera Jilberto, Emma Lejeune.

Visualization: Hiba Kobeissi, Javiera Jilberto, Emma Lejeune.

Writing – original draft: Hiba Kobeissi, Javiera Jilberto, M. Çağatay Karakan, Xining Gao, Samuel J. DePalma, Shoshana L. Das, David Nordsletten, Emma Lejeune.

Writing – review & editing: Hiba Kobeissi, Brendon M. Baker, Christopher S. Chen, Emma Lejeune.

References

1. Brush JE, Sherbino J, Norman GR. Diagnostic reasoning in cardiovascular medicine. bmj. 2022; 376. PMID: 34987062
2. Vardas PE, Asselbergs FW, van Smeden M, Friedman P. The year in cardiovascular medicine 2021: digital health and innovation. European heart journal. 2022; 43(4):271–279. https://doi.org/10.1093/eurheartj/ehab874 PMID: 34974610
3. Kaptoge S, Pennells L, De Bacquer D, Cooney MT, Kavousi M, Stevens G, et al. World Health Organization cardiovascular disease risk charts: revised models to estimate risk in 21 global regions. The Lancet Global Health. 2019; 7(10):e1332–e1345. https://doi.org/10.1016/S2214-109X(19)30318-3
4. Nguyen PD, de Bakker DE, Bakkers J. Cardiac regenerative capacity: an evolutionary afterthought? Cellular and molecular life sciences. 2021; 78(12):5107–5122. https://doi.org/10.1007/s00018-021-03831-9 PMID: 33950316
5. Bursill CA, Smith NJ, Palpant N, Tan I, Sunde M, Harvey RP, et al. Don't Turn Off the Tap! The Importance of Discovery Science to the Australian Cardiovascular Sector and Improving Clinical Outcomes Into the Future. Heart, Lung and Circulation. 2022. https://doi.org/10.1016/j.hlc.2022.06.669 PMID: 35961820
6. Thygesen K, Alpert JS, White HD. Universal Definition of Myocardial Infarction. Circulation. 2007; 116 (22):2634–2653. https://doi.org/10.1161/CIRCULATIONAHA.107.187397 PMID: 17951284
7. Takahashi K, Tanabe K, Ohnuki M, Narita M, Ichisaka T, Tomoda K, et al. Induction of pluripotent stem cells from adult human fibroblasts by defined factors. cell. 2007; 131(5):861–872. https://doi.org/10.1016/j.cell.2007.11.019 PMID: 18035408
8. Brandão KO, Tabel VA, Atsma DE, Mummery CL, Davis RP. Human pluripotent stem cell models of cardiac disease: from mechanisms to therapies. Disease models & mechanisms. 2017; 10(9):1039–1059. https://doi.org/10.1242/dmm.030320 PMID: 28883014
9. Nakao S, Ihara D, Hasegawa K, Kawamura T. Applications for induced pluripotent stem cells in disease modelling and drug development for heart diseases. European Cardiology Review. 2020; 15. https://doi.org/10.15420/ecr.2019.03 PMID: 32180835
10. Daly AC, Davidson MD, Burdick JA. 3D bioprinting of high cell-density heterogeneous tissue models through spheroid fusion within self-healing hydrogels. Nature communications. 2021; 12(1):753. https://doi.org/10.1038/s41467-021-21029-2 PMID: 33531489
11. Dou W, Malhi M, Zhao Q, Wang L, Huang Z, Law J, et al. Microengineered platforms for characterizing the contractile function of in vitro cardiac models. Microsystems & Nanoengineering. 2022; 8(1):26. https://doi.org/10.1038/s41378-021-00344-0 PMID: 35299653
12. Hnatiuk AP, Briganti F, Staudt DW, Mercola M. Human iPSC modeling of heart disease for drug development. Cell chemical biology. 2021; 28(3):271–282. https://doi.org/10.1016/j.chembiol.2021.02.016 PMID: 33740432
13. Ronaldson-Bouchard K, Vunjak-Novakovic G. Organs-on-a-chip: a fast track for engineered human tissues in drug development. Cell stem cell. 2018; 22(3):310–324. https://doi.org/10.1016/j.stem.2018.02.011 PMID: 29499151
14. Hirt MN, Hansen A, Eschenhagen T. Cardiac tissue engineering: state of the art. Circulation research. 2014; 114(2):354–367. https://doi.org/10.1161/CIRCRESAHA.114.300522 PMID: 24436431

15. Masumoto H, Ikuno T, Takeda M, Fukushima H, Marui A, Katayama S, et al. Human iPS cell-engineered cardiac tissue sheets with cardiomyocytes and vascular cells for cardiac regeneration. Scientific reports. 2014; 4(1):1–7. https://doi.org/10.1038/srep06716 PMID: 25336194

16. Karbassi E, Fenix A, Marchiano S, Muraoka N, Nakamura K, Yang X, et al. Cardiomyocyte maturation: advances in knowledge and implications for regenerative medicine. Nature Reviews Cardiology. 2020; 17(6):341–359. https://doi.org/10.1038/s41569-019-0331-x PMID: 32015528

17. DePalma SJ, Davidson CD, Stis AE, Helms AS, Baker BM. Microenvironmental determinants of organized iPSC-cardiomyocyte tissues on synthetic fibrous matrices. Biomater Sci. 2021; 9:93–107. https://doi.org/10.1039/d0bm01247e PMID: 33325920

18. Batalov I, Jallerat Q, Kim S, Bliley J, Feinberg AW. Engineering aligned human cardiac muscle using developmentally inspired fibronectin micropatterns. Scientific Reports. 2021; 11(1):1–14. https://doi.org/10.1038/s41598-021-87550-y PMID: 34075068

19. Jayne RK, Karakan MÇ, Zhang K, Pierce N, Michas C, Bishop DJ, et al. Direct laser writing for cardiac tissue engineering: a microfluidic heart on a chip with integrated transducers. Lab on a Chip. 2021; 21 (9):1724–1737. https://doi.org/10.1039/D0LC01078B PMID: 33949395

20. Karakan MÇ. A Direct-Laser-Written Heart-on-a-Chip Platform for Generation and Stimulation of Engineered Heart Tissues [Ph.D. thesis]. Boston University; 2023.

21. Ronaldson-Bouchard K, Yeager K, Teles D, Chen T, Ma S, Song L, et al. Engineering of human cardiac muscle electromechanically matured to an adult-like phenotype. Nature protocols. 2019; 14 (10):2781–2817. https://doi.org/10.1038/s41596-019-0189-8 PMID: 31492957

22. Javor J, Sundaram S, Chen CS, Bishop DJ. A Microtissue Platform to Simultaneously Actuate and Detect Mechanical Forces via Non-Contact Magnetic Approach. Journal of Microelectromechanical Systems. 2021; 30(1):96–104. https://doi.org/10.1109/JMEMS.2020.3036978

23. Zhao Y, Rafatian N, Feric NT, Cox BJ, Aschar-Sobbi R, Wang EY, et al. A platform for generation of chamber-specific cardiac tissues and disease modeling. Cell. 2019; 176(4):913–927. https://doi.org/10.1016/j.cell.2018.11.042 PMID: 30686581

24. Ruan JL, Tulloch NL, Razumova MV, Saiget M, Muskheli V, Pabon L, et al. Mechanical stress conditioning and electrical stimulation promote contractility and force maturation of induced pluripotent stem cell-derived human cardiac tissue. Circulation. 2016; 134(20):1557–1567. https://doi.org/10.1161/CIRCULATIONAHA.114.014998 PMID: 27737958

25. Feric NT, Pallotta I, Singh R, Bogdanowicz DR, Gustilo MM, Chaudhary KW, et al. Engineered cardiac tissues generated in the biowire II: a platform for human-based drug discovery. Toxicological sciences. 2019; 172(1):89–97. https://doi.org/10.1093/toxsci/kfz168 PMID: 31385592

26. Huang CY, Maia-Joca RPM, Ong CS, Wilson I, DiSilvestre D, Tomaselli GF, et al. Enhancement of human iPSC-derived cardiomyocyte maturation by chemical conditioning in a 3D environment. Journal of molecular and cellular cardiology. 2020; 138:1–11. https://doi.org/10.1016/j.yjmcc.2019.10.001 PMID: 31655038

27. Lee J, Manoharan V, Cheung L, Lee S, Cha BH, Newman P, et al. Nanoparticle-based hybrid scaffolds for deciphering the role of multimodal cues in cardiac tissue engineering. ACS nano. 2019; 13 (11):12525–12539. https://doi.org/10.1021/acsnano.9b03050 PMID: 31621284

28. Huebsch N, Charrez B, Neiman G, Siemons B, Boggess SC, Wall S, et al. Metabolically driven maturation of human-induced-pluripotent-stem-cell-derived cardiac microtissues on microfluidic chips. Nature Biomedical Engineering. 2022; 6(4):372–388. https://doi.org/10.1038/s41551-022-00884-4 PMID: 35478228

29. Pointon A, Pilling J, Dorval T, Wang Y, Archer C, Pollard C. From the cover: high-throughput imaging of cardiac microtissues for the assessment of cardiac contraction during drug discovery. Toxicological Sciences. 2017; 155(2):444–457. https://doi.org/10.1093/toxsci/kfw227 PMID: 28069985

30. Huebsch N, Loskill P, Mandegar MA, Marks NC, Sheehan AS, Ma Z, et al. Automated video-based analysis of contractility and calcium flux in human-induced pluripotent stem cell-derived cardiomyocytes cultured over different spatial scales. Tissue Engineering Part C: Methods. 2015; 21(5):467–479. https://doi.org/10.1089/ten.tec.2014.0283 PMID: 25333967

31. Ronaldson-Bouchard K, Ma SP, Yeager K, Chen T, Song L, Sirabella D, et al. Advanced maturation of human cardiac tissue grown from pluripotent stem cells. Nature. 2018; 556(7700):239–243. https://doi.org/10.1038/s41586-018-0016-3 PMID: 29618819

32. Hansen A, Eder A, Bönstrup M, Flato M, Mewe M, Schaaf S, et al. Development of a drug screening platform based on engineered heart tissue. Circulation research. 2010; 107(1):35–44. https://doi.org/10.1161/CIRCRESAHA.109.211458 PMID: 20448218

33. Tamargo MA, Nash TR, Fleischer S, Kim Y, Vila OF, Yeager K, et al. milliPillar: a platform for the generation and real-time assessment of human engineered cardiac tissues. ACS biomaterials science &

engineering. 2021; 7(11):5215–5229. https://doi.org/10.1021/acsbiomaterials.1c01006 PMID: 34668692

34. Rivera-Arbeláez JM, Cofiño-Fabres C, Schwach V, Boonen T, Ten Den SA, Vermeul K, et al. Contractility analysis of human engineered 3D heart tissues by an automatic tracking technique using a standalone application. Plos one. 2022; 17(4):e0266834. https://doi.org/10.1371/journal.pone.0266834 PMID: 35421132

35. Oyunbaatar NE, Lee DH, Patil SJ, Kim ES, Lee DW. Biomechanical characterization of cardiomyocyte using PDMS pillar with microgrooves. Sensors. 2016; 16(8):1258. https://doi.org/10.3390/s16081258 PMID: 27517924

36. Dostanić M, Windt LM, Stein JM, van Meer BJ, Bellin M, Orlova V, et al. A Miniaturized EHT Platform for Accurate Measurements of Tissue Contractile Properties. Journal of Microelectromechanical Systems. 2020; 29(5):881–887. https://doi.org/10.1109/JMEMS.2020.3011196

37. Thavandiran N, Hale C, Blit P, Sandberg ML, McElvain ME, Gagliardi M, et al. Functional arrays of human pluripotent stem cell-derived cardiac microtissues. Scientific reports. 2020; 10(1):6919. https://doi.org/10.1038/s41598-020-62955-3 PMID: 32332814

38. Tsan YC, DePalma SJ, Zhao YT, Capilnasiu A, Wu YW, Elder B, et al. Physiologic biomechanics enhance reproducible contractile development in a stem cell derived cardiac muscle platform. Nature Communications. 2021; 12(1):6167. https://doi.org/10.1038/s41467-021-26496-1 PMID: 34697315

39. Méry A, Ruppel A, Revilloud J, Balland M, Cappello G, Boudou T. Light-driven biological actuators to probe the rheology of 3D microtissues. Nature Communications. 2023; 14(1):717. https://doi.org/10.1038/s41467-023-36371-w PMID: 36759504

40. Steadman B, Moore K, Spitzer K, Bridge J. A video system for measuring motion in contracting heart cells. IEEE Transactions on Biomedical Engineering. 1988; 35(4):264–272. https://doi.org/10.1109/10.1375 PMID: 3360456

41. Lim CC, Yang H, Yang M, Wang CK, Shi J, Berg EA, et al. A novel mutant cardiac troponin C disrupts molecular motions critical for calcium binding affinity and cardiomyocyte contractility. Biophysical journal. 2008; 94(9):3577–3589. https://doi.org/10.1529/biophysj.107.112896 PMID: 18212018

42. Hossain MM, Shimizu E, Saito M, Rao SR, Yamaguchi Y, Tamiya E. Non-invasive characterization of mouse embryonic stem cell derived cardiomyocytes based on the intensity variation in digital beating video. Analyst. 2010; 135(7):1624–1630. https://doi.org/10.1039/c0an00208a PMID: 20517541

43. Hayakawa T, Kunihiro T, Dowaki S, Uno H, Matsui E, Uchida M, et al. Noninvasive evaluation of contractile behavior of cardiomyocyte monolayers based on motion vector analysis. Tissue Engineering Part C: Methods. 2012; 18(1):21–32. https://doi.org/10.1089/ten.tec.2011.0273 PMID: 21851323

44. Ahola A, Kiviaho AL, Larsson K, Honkanen M, Aalto-Setälä K, Hyttinen J. Video image-based analysis of single human induced pluripotent stem cell derived cardiomyocyte beating dynamics using digital image correlation. Biomedical engineering online. 2014; 13(1):1–18. https://doi.org/10.1186/1475-925X-13-39 PMID: 24708714

45. Hayakawa T, Kunihiro T, Ando T, Kobayashi S, Matsui E, Yada H, et al. Image-based evaluation of contraction–relaxation kinetics of human-induced pluripotent stem cell-derived cardiomyocytes: Correlation and complementarity with extracellular electrophysiology. Journal of Molecular and Cellular Cardiology. 2014; 77:178–191. https://doi.org/10.1016/j.yjmcc.2014.09.010 PMID: 25257913

46. Rajasingh S, Thangavel J, Czirok A, Samanta S, Roby KF, Dawn B, et al. Generation of functional cardiomyocytes from efficiently generated human iPSCs and a novel method of measuring contractility. PLoS One. 2015; 10(8):e0134093. https://doi.org/10.1371/journal.pone.0134093 PMID: 26237415

47. Czirok A, Isai DG, Kosa E, Rajasingh S, Kinsey W, Neufeld Z, et al. Optical-flow based non-invasive analysis of cardiomyocyte contractility. Scientific reports. 2017; 7(1):10404. https://doi.org/10.1038/s41598-017-10094-7 PMID: 28871207

48. Shradhanjali A, Riehl BD, Duan B, Yang R, Lim JY. Spatiotemporal characterizations of spontaneously beating cardiomyocytes with adaptive reference digital image correlation. Scientific reports. 2019; 9(1):1–10. https://doi.org/10.1038/s41598-019-54768-w PMID: 31804542

49. Scalzo S, Afonso MQ, da Fonseca NJ, Jesus IC, Alves AP, Mendonça CA, et al. Dense optical flow software to quantify cellular contractility. Cell Reports Methods. 2021; 1(4). https://doi.org/10.1016/j.crmeth.2021.100044 PMID: 35475144

50. Cheng Z, Yang Y, Jiang K, Nie H, Yang X, Tu Z, et al. Quantification of cardiomyocyte contraction in vitro and drug screening by MyocytoBeats. Journal of Cardiovascular Translational Research. 2023; p. 1–10. PMID: 36715820

51. Sala L, Van Meer BJ, Tertoolen LG, Bakkers J, Bellin M, Davis RP, et al. MUSCLEMOTION: a versatile open software tool to quantify cardiomyocyte and cardiac muscle contraction in vitro and in vivo. Circulation research. 2018; 122(3):e5–e16. https://doi.org/10.1161/CIRCRESAHA.117.312067 PMID: 29282212

52. Bouguet JY, et al. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel corporation. 2001; 5(1-10):4.

53. Das SL, Sutherland BP, Lejeune E, Eyckmans J, Chen CS. Mechanical response of cardiac microtissues to acute localized injury. American Journal of Physiology-Heart and Circulatory Physiology. 2022; 323(4):H738–H748. https://doi.org/10.1152/ajpheart.00305.2022 PMID: 36053751

54. Raghupathy R. Form from function: generalized anisotropic inverse mechanics for soft tissues. University of Minnesota; 2011.

55. Genovese K, Casaletto L, Humphrey JD, Lu J. Digital image correlation-based point-wise inverse characterization of heterogeneous material properties of gallbladder in vitro. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2014; 470(2167):20140152. https://doi.org/10.1098/rspa.2014.0152

56. Narayanan B, Olender ML, Marlevi D, Edelman ER, Nezami FR. An inverse method for mechanical characterization of heterogeneous diseased arteries using intravascular imaging. Scientific Reports. 2021; 11(1):22540. https://doi.org/10.1038/s41598-021-01874-3 PMID: 34795350

57. Wang Z, Estrada JB, Arruda EM, Garikipati K. Inference of deformation mechanisms and constitutive response of soft material surrogates of biological tissue by full-field characterization and data-driven variational system identification. Journal of the Mechanics and Physics of Solids. 2021; 153:104474. https://doi.org/10.1016/j.jmps.2021.104474

58. Weiss D, Cavinato C, Gray A, Ramachandra AB, Avril S, Humphrey JD, et al. Mechanics-driven mechanobiological mechanisms of arterial tortuosity. Science advances. 2020; 6(49):eabd3574. https://doi.org/10.1126/sciadv.abd3574 PMID: 33277255

59. Sree VD, Rausch MK, Tepole AB. Linking microvascular collapse to tissue hypoxia in a multiscale model of pressure ulcer initiation. Biomechanics and modeling in mechanobiology. 2019; 18(6):1947–1964. https://doi.org/10.1007/s10237-019-01187-5 PMID: 31203488

60. Kong F, Pham T, Martin C, McKay R, Primiano C, Hashim S, et al. Finite element analysis of tricuspid valve deformation from multi-slice computed tomography images. Annals of biomedical engineering. 2018; 46(8):1112–1127. https://doi.org/10.1007/s10439-018-2024-8 PMID: 29663193

61. Joldes G, Bourantas G, Zwick B, Chowdhury H, Wittek A, Agrawal S, et al. Suite of meshless algorithms for accurate computation of soft tissue deformation for surgical simulation. Medical Image Analysis. 2019; 56:152–171. https://doi.org/10.1016/j.media.2019.06.004 PMID: 31229760

62. de Zélicourt DA, Marsden A, Fogel MA, Yoganathan AP. Imaging and patient-specific simulations for the Fontan surgery: current methodologies and clinical applications. Progress in Pediatric Cardiology. 2010; 30(1-2):31–44. https://doi.org/10.1016/j.ppedcard.2010.09.005 PMID: 25620865

63. Wang Z, Kieu H, Nguyen H, Le M. Digital image correlation in experimental mechanics and image registration in computer vision: Similarities, differences and complements. Optics and Lasers in Engineering. 2015; 65:18–27. https://doi.org/10.1016/j.optlastec.2014.11.002

64. Chu T, Ranson W, Sutton MA. Applications of digital-image-correlation techniques to experimental mechanics. Experimental mechanics. 1985; 25:232–244. https://doi.org/10.1007/BF02325092

65. Zhang D, Arola DD. Applications of digital image correlation to biological tissues. Journal of Biomedical Optics. 2004; 9(4):691–699. https://doi.org/10.1117/1.1753270 PMID: 15250755

66. Annaidh AN, Bruyère K, Destrade M, Gilchrist MD, Otténio M. Characterization of the anisotropic mechanical properties of excised human skin. Journal of the mechanical behavior of biomedical materials. 2012; 5(1):139–148. https://doi.org/10.1016/j.jmbbm.2011.08.016

67. Blaber J, Adair B, Antoniou A. Ncorr: open-source 2D digital image correlation matlab software. Experimental Mechanics. 2015; 55(6):1105–1122. https://doi.org/10.1007/s11340-015-0009-1

68. Yang J, Bhattacharya K. Fast adaptive mesh augmented lagrangian digital image correlation. Experimental Mechanics. 2021; 61(4):719–735. https://doi.org/10.1007/s11340-021-00695-9

69. Solav D, Moerman KM, Jaeger AM, Genovese K, Herr HM. MultiDIC: An Open-Source Toolbox for Multi-View 3D Digital Image Correlation. IEEE Access. 2018; 6:30520–30535. https://doi.org/10.1109/ACCESS.2018.2843725

70. Murienne BJ, Nguyen TD. A comparison of 2D and 3D digital image correlation for a membrane under inflation. Optics and lasers in engineering. 2016; 77:92–99. https://doi.org/10.1016/j.optlaseng.2015.07.013 PMID: 26543296

71. Bay BK, Smith TS, Fyhrie DP, Saad M. Digital volume correlation: three-dimensional strain mapping using X-ray tomography. Experimental mechanics. 1999; 39:217–226. https://doi.org/10.1007/BF02323555

72. Dall'Ara E, Barber D, Viceconti M. About the inevitable compromise between spatial resolution and accuracy of strain measurement for bone tissue: A 3D zero-strain study. Journal of biomechanics. 2014; 47(12):2956–2963. https://doi.org/10.1016/j.jbiomech.2014.07.019 PMID: 25113806

73. Acosta Santamaría VA, Flechas García M, Molimard J, Avril S. Three-dimensional full-field strain measurements across a whole porcine aorta subjected to tensile loading using optical coherence tomography–digital volume correlation. Frontiers in Mechanical Engineering. 2018; 4:3. https://doi.org/10.3389/fmech.2018.00003

74. Bersi MR, Acosta Santamaría VA, Marback K, Di Achille P, Phillips EH, Goergen CJ, et al. Multimodality imaging-based characterization of regional material properties in a murine model of aortic dissection. Scientific reports. 2020; 10(1):9244. https://doi.org/10.1038/s41598-020-65624-7 PMID: 32514185

75. Lucas BD, Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence—Volume 2. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1981. p. 674–679.

76. Boyle JJ, Kume M, Wyczalkowski MA, Taber LA, Pless RB, Xia Y, et al. Simple and accurate methods for quantifying deformation, disruption, and development in biological tissues. Journal of the Royal Society Interface. 2014; 11(100):20140685. https://doi.org/10.1098/rsif.2014.0685 PMID: 25165601

77. Boyle JJ, Soepriatna A, Damen F, Rowe RA, Pless RB, Kovacs A, et al. Regularization-free strain mapping in three dimensions, with application to cardiac ultrasound. Journal of biomechanical engineering. 2019; 141(1):011010. https://doi.org/10.1115/1.4041576 PMID: 30267039

78. Bourne R, Bourne R. ImageJ. Fundamentals of digital imaging in medicine. 2010; p. 185–188. https://doi.org/10.1007/978-1-84882-087-6_9

79. Inc TM. MATLAB; 2022. Available from: https://www.mathworks.com.

80. Ghanbari M. The cross-search algorithm for motion estimation (image coding). IEEE Transactions on Communications. 1990; 38(7):950–953. https://doi.org/10.1109/26.57512

81. Geuzaine C, Remacle JF. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. International journal for numerical methods in engineering. 2009; 79(11):1309–1331. https://doi.org/10.1002/nme.2579

82. Alnæs M, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, et al. The FEniCS project version 1.5. Archive of Numerical Software. 2015; 3(100):9–23.

83. Logg A, Mardal KA, Wells G. Automated solution of differential equations by the finite element method: The FEniCS book. vol. 84. Germany: Springer Science & Business Media; 2012.

84. Pezzuto S, Ambrosi D, Quarteroni A. An orthotropic active–strain model for the myocardium mechanics and its numerical approximation. European Journal of Mechanics—A/Solids. 2014; 48:83–96. https://doi.org/10.1016/j.euromechsol.2014.03.006

85. Gurev V, Pathmanathan P, Fattebert JL, Wen HF, Magerlein J, Gray RA, et al. A high-resolution computational model of the deforming human heart. Biomechanics and modeling in mechanobiology. 2015; 14:829–849. https://doi.org/10.1007/s10237-014-0639-8 PMID: 25567753

86. Finsberg H, Xi C, Tan JL, Zhong L, Genet M, Sundnes J, et al. Efficient estimation of personalized biventricular mechanical function employing gradient-based optimization. International journal for numerical methods in biomedical engineering. 2018; 34(7):e2982. https://doi.org/10.1002/cnm.2982 PMID: 29521015

87. Land S, Gurev V, Arens S, Augustin CM, Baron L, Blake R, et al. Verification of cardiac mechanics software: benchmark problems and solutions for testing active and passive material behaviour. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2015; 471 (2184):20150641. https://doi.org/10.1098/rspa.2015.0641 PMID: 26807042

88. van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, et al. scikit-image: image processing in Python. PeerJ. 2014; 2:e453. https://doi.org/10.7717/peerj.453 PMID: 25024921

89. Jilberto J, DePalma SJ, Lo J, Kobeissi H, Lejeune E, Baker BM, et al. A data–driven computational modeling for engineered cardiac microtissues. Acta Biomaterialia. 2023. https://doi.org/10.1016/j.actbio.2023.10.025 PMID: 37879587

90. Perlin K. An Image Synthesizer. Siggraph Computer Graphics. 1985; 19(3):287–296. https://doi.org/10.1145/325165.325247

91. Boudou T, Legant WR, Mu A, Borochin MA, Thavandiran N, Radisic M, et al. A microfabricated platform to measure and manipulate the mechanics of engineered cardiac microtissues. Tissue Engineering Part A. 2012; 18(9-10):910–919. https://doi.org/10.1089/ten.tea.2011.0341 PMID: 22092279

92. Xu F, Zhao R, Liu AS, Metz T, Shi Y, Bose P, et al. A microfabricated magnetic actuation device for mechanical conditioning of arrays of 3D microtissues. Lab on a Chip. 2015; 15(11):2496–2503. https://doi.org/10.1039/c4lc01395f PMID: 25959132

93. Bielawski KS, Leonard A, Bhandari S, Murry CE, Sniadecki NJ. Real-time force and frequency analysis of engineered human heart tissue derived from induced pluripotent stem cells using magnetic sensing.

Tissue Engineering Part C: Methods. 2016; 22(10):932–940. https://doi.org/10.1089/ten.tec.2016.0257 PMID: 27600722

94. DePalma SJ, Jilberto J, Stis AE, Huang DD, Lo J, Davidson CD, et al. Microenvironmental mechanical inputs dictate the assembly and function of iPSC-derived cardiomyocyte microtissues. Nature Communications. 2023(under review).

95. Lian X, Zhang J, Azarin SM, Zhu K, Hazeltine LB, Bao X, et al. Directed cardiomyocyte differentiation from human pluripotent stem cells by modulating Wnt/β-catenin signaling under fully defined conditions. Nature protocols. 2013; 8(1):162–175. https://doi.org/10.1038/nprot.2012.150 PMID: 23257984

96. Kobeissi H, Jilberto J, Karakan MÇ, Gao X, DePalma SJ, Das SL, et al.. Microbundle Time-lapse Dataset; Dryad; 2023.

97. Bradski G, Kaehler A. Learning OpenCV: Computer vision with the OpenCV library. Sebastopol, CA: O'Reilly Media, Inc.; 2008.

98. Shi J, et al. Good features to track. In: 1994 Proceedings of IEEE conference on computer vision and pattern recognition. IEEE; 1994. p. 593–600.

99. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods. 2020; 17:261–272. https://doi.org/10.1038/s41592-019-0686-2 PMID: 32015543

100. Stout DA, Bar-Kochba E, Estrada JB, Toyjanova J, Kesari H, Reichner JS, et al. Mean deformation metrics for quantifying 3D cell–matrix interactions without requiring information about matrix material properties. Proceedings of the National Academy of Sciences. 2016; 113(11):2898–2903. https://doi.org/10.1073/pnas.1510935113 PMID: 26929377

101. Zhao B, Zhang K, Chen CS, Lejeune E. Sarc-Graph: Automated segmentation, tracking, and analysis of sarcomeres in hiPSC-derived cardiomyocytes. PLOS Computational Biology. 2021; 17(10):1–27. https://doi.org/10.1371/journal.pcbi.1009443 PMID: 34613960

102. Holzapfel GA. Nonlinear solid mechanics: a continuum approach for engineering science; 2002.

103. Hunter JD. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering. 2007; 9 (3):90–95. https://doi.org/10.1109/MCSE.2007.55

104. Legant WR, Pathak A, Yang MT, Deshpande VS, McMeeking RM, Chen CS. Microfabricated tissue gauges to measure and manipulate forces from 3D microtissues. Proceedings of the National Academy of Sciences. 2009; 106(25):10097–10102. https://doi.org/10.1073/pnas.0900174106 PMID: 19541627

105. Kabadi PK, Vantangoli MM, Rodd AL, Leary E, Madnick SJ, Morgan JR, et al. Into the depths: Techniques for in vitro three-dimensional microtissue visualization. Biotechniques. 2015; 59(5):279–286. https://doi.org/10.2144/000114353 PMID: 26554505

106. Gilbert EN, Pollak HO. Amplitude distribution of shot noise. The Bell System Technical Journal. 1960; 39(2):333–350. https://doi.org/10.1002/j.1538-7305.1960.tb01603.x

107. Wang H, Svoronos AA, Boudou T, Sakar MS, Schell JY, Morgan JR, et al. Necking and failure of constrained 3D microtissues induced by cellular tension. Proceedings of the National Academy of Sciences. 2013; 110(52):20923–20928. https://doi.org/10.1073/pnas.1313662110 PMID: 24324149

108. Guatimosim S, Guatimosim C, Song LS. Imaging calcium sparks in cardiac myocytes. Light Microscopy: Methods and Protocols. 2011; p. 205–214. https://doi.org/10.1007/978-1-60761-950-5_12 PMID: 21153794

109. Mohammadzadeh S, Lejeune E. SarcGraph: A Python package for analyzing the contractile behavior of pluripotent stem cell-derived cardiomyocytes. Journal of Open Source Software. 2023; 8(85):5322. https://doi.org/10.21105/joss.05322

110. Rios B, Bu A, Sheehan T, Kobeissi H, Kohli S, Shah K, et al. Mechanically programming anisotropy in engineered muscle with actuating extracellular matrices. Device. 2023; 1(4). https://doi.org/10.1016/j.device.2023.100097