

Deep Learning for Traction Force Microscopy User Guide

Requirement: MATLAB 2019b or later capable of performing 3D convolution for neural networks. In addition, installation of the FTTC plugin for ImageJ is recommended for the comparison of deep learning method with a widely used conventional method (<https://sites.google.com/site/qingzongtseng/tfm>).

Repositories: all the MATLAB programs are distributed via GitHub. Due to the sheer size of the dataset, training and testing data are distributed via Box, using the same folder structure, at <https://cmu.box.com/s/n34hbfopwa3r6rftvtfn4ckc403hk43d>. Folders in the data repository should be copied to the corresponding folder locations for the programs. For example, the folder test/elasticity/testData should be moved into the folder of test/elasticity after downloaded from GitHub.

Representation of Vector Fields: Traction Stress and the corresponding substrate strain are represented as tensors of S by S by 2 planes, where S must be 104, 160, or 256 and the two planes carry x and y component of stress or strain vectors. Strain is measured in pixels. The tensor must be dense representation of the field, i.e., each position within the cell border must be represented in the tensors and gaps must be filled in by interpolation. For images recorded at a limited resolution, strain may be measured at a resolution lower than that specified above then expanded by interpolation to generate tensors of S by S by 2. For example, bead images of 480×480 pixels may be processed by PIV to generate 80×80 dense arrays, then expanded by interpolation to obtain tensors of $160 \times 160 \times 2$.

Data Files: Data files used for training and testing, generated by simulation, are provided in the subfolders of trainData and testData respectively, in MATLAB .mat format. Stresses are represented by the tensor variable `trac` and strains are represented by the variable `trac`. In addition, x and y coordinates of the cell border are represented by arrays `brdx` and `brdy` respectively. Sample experimental data files using NIH 3T3 cells are located in the folder of cells. These files are not required for the application of the neural network but are provided in case the user would like to develop new machine learning models.

Preprocessing of Measured Strain Fields: Pairs of bead images before and after cell removal are first cropped/scaled to form square images of 104×104 , 160×160 , or 256×256 pixels, then analyzed using the PIV plugin of ImangJ (or Fiji) to generate strain fields. See the file PIV_parameters104.pdf in the piv-fttc folder for the parameters that have been used for processing 728×728 bead images. The resulting output from the PIV plugin is a text file with a strain vector every 7 pixels, which is then processed with the command `dspl = pivToDspl(pivFn,728,7)` to generate the strain field tensor `dspl` of $104 \times 104 \times 2$; `pivFn` is the full path of the .txt file placed in quotes. Similarly, for bead images of 480×480 , PIV is applied with parameters shown in PIV_parameters80.pdf to generate a text file with a strain vector every 6 pixels. The resulting .txt file is then processed with the command `dspl = pivToDspl(pivFn,480,6)` to generate a strain field tensor `dspl` of $80 \times 80 \times 2$, followed by interpolation to convert `dspl` to $160 \times 160 \times 2$. A simple example MATLAB script that expands a 80×80 field to 160×160 is provided as 80To160.m.

Coordinates of cell border, `brdx` and `brdy`, are generated with the command `[brdx,brdy] = txtToBrd(brdFn,size,spacing)` where `brdFn` is the full path of a text file (placed in quotes) that contains alternating x and y coordinates of border points separated by spaces or a carriage return. Size is the size of cell image and spacing is the scaling factor between the cell image and the piv field. For example, borders drawn on a 728×728 image should be processed with the command `[brdx,brdy]=txtToBrd(brdFn,728,7)` to generate the border for 104×104 fields. There is no set density for the points in cell border; the program simply connects consecutive points with a line segment. Sample processed files for NIH 3T3 cells are located in the folder of cells/processedDataFiles.

Calculation of Traction Stress from Substrate Strain: The strain field `dspl` may be filtered using the `filtDspl` command to suppress noise and residual alignment errors. Type "`help filtDspl`" for the documentation. The field of traction stress is then generated by `trac = predictTrac(dspl,E)` where `dspl` is the strain field tensor and `E` is Young's Modulus of the substrate in Pascals. The function loads pretrained neural networks named `tracnet104.mat`, `tracnet160.mat`, or `tracnet256.mat` depending on the size of the strain tensor `dspl`. The folder containing `tracnet` files must be placed in the search path. Output traction stress `trac` is in the unit of Pascals. Neural networks as provided were trained for a Poisson ratio of 0.45.

Folders Containing Data:

cells: experimental data of NIH 3T3 cells.

Subfolder data: txt files for cell border and output from the PIV plugin of ImageJ, both covering an area of 728x728 pixels with vectors spaced every 7 pixels. The script dsplGen.m then shrinks the field by 7x to generate a field of tightly packed strain vectors with a size of 104x104

Subfolder dspl: Data generated from the data folder, .mat files of strain fields and border coordinates of a collection of cells. The strain field is filtered with filtDspl.m

test: contains data of simulated cells for testing the performance of tracnet to predict the traction stress of simulated cells of different shapes, including generic, keratocyte, and neuron. The script calcError.m calculates normalized error against ground truth under no-noise and noisy conditions.

Additional tests are performed under other subfolders:

elasticity: Testing simulated generic cells placed on substrates with Young's Modulus of 2,500, 5,000, 10,000, 20,000, or 40,000 Pascals

magnification: Testing traction stresses and corresponding strains for simulated generic cells either enlarged or reduced by 25% to simulate the use of different microscope lenses

sizeScale: Testing traction stresses and corresponding strains for simulated generic cells sampled at a pixel density of 104x104, 160x160, or 256x256 pixels per image

tracScale: Testing traction stresses and corresponding strains for simulated generic cells generating a maximal stress of 1,000, 2,000, 4,000, or 8,000 Pascals

train: contains strain and the corresponding stress patterns used for training the neural network. The dataset includes simulated cells both with and without radial symmetry in separate folders. Asymmetric cells were augmented with 16 rotational angles. Scripts for building the network architecture and training the neural network are also included.

Folders Containing Functions and Scripts

“function” must be called with proper arguments enclosed in parentheses

“script” requires setting parameters in the first few lines of the code

use the help command to retrieve documentation

main

calcDspl.m: function to calculate strain field generated by a traction stress field

calcEnergy.m: function to calculate strain energy of a traction stress field and the resulting strain

errorDisp.m: function to calculate normalized error of measured strain field against ground truth

errorTrac.m: function to calculate normalized error of a predicted traction stress field against ground truth

filtDspl.m: function to filter a strain field to remove alignment errors, apply median filter to exterior vectors, and replace vectors that deviate too much from neighbors

filtTrac.m: function to filter a traction stress field to remove vectors outside the border (and optionally noise vectors)

plotDspl.m: function to generate quiver plot and heat map of a strain field

plotError.m: function to plot the error of a field against ground truth

plotTrac.m: function to generate quiver plot and heat map of a traction stress field

predictTrac.m: function to predict traction stress field from a strain field

tracEnergyCutoff.m: function to determine the threshold percentile traction stress for generating a specified percentage of strain energy

txtToBrd.m: function to generate x and y border coordinates from a text file

piv-fttc

80To160: sample script to generate a 160x160x2 strain field from a strain field of 80x80x2

dsplToTxt.m: function to convert a strain field tensor into a text file for input into FTTC ImageJ plugin

fttcToTrac.m: function to convert the output from FTTC ImageJ plugin into a traction stress field tensor

pivToDspl.m: function to convert the output from PIV ImageJ plugin into a strain field tensor

predictTracFTTC.m: function to predict traction stress field from a strain field using the conventional approach of Fourier Transform Traction Cytometry (FTTC), to allow the comparison with deep learning. The program serves only as a wrapper to simplify the input into the code originally developed by Sybass et al.

reg_fourier_TFM.m: function from the TFM_Package posted by Danuser lab for performing regularized fourier transform traction force reconstruction

train

createTracnet.m: function to create the neural network architecture for tracnet

plotTrain.m: function to view training dataset as quiver plots and heat maps

retrainTracnet.m: script to complete training or apply transfer learning to an existing tracnet

trainTracnet.m: script to train tracnet from scratch

test

addNoise.m: function to apply Gaussian noise to a strain field

calcError.m: script to calculate the normalized root mean squared error of prediction relative to the ground truth, generate an error table TError in the workspace for all the simulated cells in the folder testData

Cells

dsplGen.m: script to process the data folder to generate the files in folder dspl, scaling down the area to 104x104 pixels then filtering the strain field with filtDspl.m. Edit folder names in the script as appropriate.

calcError.m: script to predict traction stress field from measured strain field, then use the predicted stress field to generate predicted strain field and compare against the measured strain field, calculate normalized mean squared error, to be applied to each cell in the collection and generate an error table of TError in the workspace

showTrac.m: function to generate and render the traction stress field of specific cells based on the strain field found in the dspl folder, also render the magnitude error between predicted and measured strains