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 analysis of substrate strain and 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, test/elasticity/testData should be moved into the folder test/elasticity as downloaded from GitHub.

Representation of Vector Fields: Traction Stress and Resulting Displacements are as tensors of S by S by 2, where S must be 104, 160, or 256. The two channels carry the x and y component of stress or displacement vectors. Displacements are measured in pixels. The tensor must be dense representation of the field, i.e., each position within the cell border must be represented by a valid number. Any gaps must be filled in by interpolation. For images recorded at a limited resolution, displacements may be measured at a lower resolution than S then expanded by interpolation to generate tensors of S by S by 2. For example, images of scattered beads at 480x480 pixels may be processed by PIV to generate 80x80 dense arrays, then expanded by interpolation to obtain a tensor of 160x160x2.

Data Files: Data files of simulated cells are placed in subfolders named either trainData or testData, in .mat format. These files contain either the stress or displacement field tensors, named as trac and dspl respectively, and x and y coordinates of the cell border, as arrays named brdx and brdy compatible with the MATLAB plot function. Sample experimental data files for NIH 3T3 cells are located in the folder of cells.

Preprocessing of Measured Displacement Fields: Pairs of bead images before and after cell removal are first cropped/scaled to form a square image of 104, 160, or 256 pixels, before analyzing the displacements using the PIV plugin of ImangJ (or FiJi). See PIV_parameters104.pdf for the parameters for processing 728x728 bead images to generate a text file with a displacement vector every 7 pixels. The resulting .txt file is processed with the command dspl = pivToDspl(pivFn,728,7) to generate the displacement field tensor dspl of 104x104x2, where pivFn is the full path of the .txt file.

For bead images of 480x480, PIV is applied with parameters shown in PIV_parameters80.pdf to generate a text file with a displacement vector every 6 pixels. The resulting .txt file is then processed with the command dspl = pivToDspl(pivFn,480,6) to generate a displacement field tensor of 80x80x2. Noise filtration may be applied to this tensor using the filtDspl command, before applying interpolation to obtain a tensor of 160x160x2. A simple example for expanding a 80x80 field named dspl80 to a 160x160 field named dspl160 is shown 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 with a single row of text of alternating x and y coordinates separated by spaces. size is the size of the image where the border was drawn and spacing is the ratio between the sizes of the original image and final image. For example, borders drawn on a 728x728 image should be processed with the command [brdx,brdy]=txtToBrd(brdFn,728,7) to generate border for 104x104 fields. Sample experimental data files for NIH 3T3 cells are located in the folder of cells/processedDataFiles.

The displacement field likely requires additional filtering using the command filtDspl to reduce the noise and residual alignment errors. Type "help filtDspl" for the documentation.

Calculation of Traction Stress from Measured Displacements: Apply the function trac = predictTrac(dspl,E) where dspl is the displacement field tensor as described above and E is Young's Modulus of the substrate in the unit of Pascals. The function loads pretrained neural networks in files in the search path named tracnet104.mat, tracnet160.mat, or tracnet256.mat, for three difference tensor sizes. Output traction stress is in the unit of Pascals. Neural networks as provided were trained for a Poisson ratio of 0.45.

Folders Containing Data:

cells: contains experimental data of NIH 3T3 cells, including text files used for generating the displacement fields and cell border. The folder dspl contains displacement generated by PIV then filtered with filtDspl.m.

Subfolder data: txt files for cell border and output from the PIV plugin of ImageJ, both over a surface of 728x728 pixels with vectors spaced every 7 pixels. Apply the script dsplGen.m to scale the data down by 7x to generate tightly packed displacement vectors over an area of 104x104

Subfolder dspl: Data generated from the data folder, .mat files of displacement fields and border coordinates of a collection of cells

test: contains data of simulated cells for testing the ability of tracnet to predict the traction stress of simulated cells of different shapes, including generic, keratocyte, and neuron. In the subfolder for each shape, cutoff.xlsx contains the percentile traction stress responsible for generating 90% of the total strain energy for each cell, as generated by the script calcCutoff.m. For example, 95 means top 5% of the traction stresses are responsible for generating 90% of the total strain energy. The script calcError.m calculates normalized error against ground truth under no-noise and noisy conditions.

Additional tests are performed under other subfolders:

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

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

sizeScale: traction stresses and corresponding displacements for simulated generic cells are sampled for an image size of 104x104, 160x160, or 256x256

tracScale: traction stresses and corresponding displacements for simulated generic cells are scaled to generate a maximal magnitude of 1,000, 2,000, 4,000, or 8,000 Pascals

train: contains data of simulated cells and radial stress patterns used for training the neural network. Each cell is augmented with 8 rotation angles. .m files used for building the network architecture and training the neural network are also included.

Folders Containing Functions and Scripts (use the help command to retrive documentation):

main

calcDspl.m: calculate displacement field generated by a traction stress field

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

errorDisp.m: calculate normalized error of a displacement field against measured displacements

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

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

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

plotDspl.m: generate quiver plot and heat map of a displacement field

plotError.m: plot the error of a field against ground truth
plotTrac.m: generate quiver plot and heat map of a traction stress field
predictTrac.m: predict traction stress field from a displacement field
tracEnergyCutoff.m: determine the threshold percentile traction stress for generating a specified percentage of strain energy
txtToBrd.m: generate x and y border coordinates from a text file

piv-fttc

dsplToTxt.m: convert a displacement field tensor into a text file for input into FTTC ImageJ plugin
fttcToTrac.m: convert the output from FTTC ImageJ plugin into a traction stress field tensor
pivToDspl.m: convert the output from PIV ImageJ plugin into a displacement field tensor
predictTracFTTC.m: predict traction stress field from a displacement 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.

train

createTracnet.m: create the neural network architecture for tracnet
plotTrain.m: view training dataset as quiver plots and heat maps
retrainTracnet.m (script): complete training or apply transfer learning to an existing tracnet
trainTracnet.m (script): train tracnet from scratch

test

addNoise.m: apply Gaussian noise to a displacement field
(In various subfolders)
calcError.m (script): calculate the normalized root mean squared error of prediction relative to the ground truth, generate an error table TError in the workspace for simulated cells in testData

Cells

dsplGen.m (script): process the data folder to generate the files in folder dspl, scaling down the area to 104x104 pixels then filtering the displacement field with filtDspl.m
calcError.m (script): first predict traction stress field from measured displacement field, then use the predicted stress field to generate predicted displacement field and compare against the measured displacement field, calculate normalized mean squared error for each cell in the collection, generate an error table of TError in the workspace
showTrac.m: generate and render the traction stress field of specific cells based on the displacement field found in the dspl folder, also render the magnitude error between predicted and measured displacements