Supplementary Information: MATLAB code for stretcher.

The code to control the movement of the motor was written in MATLAB2016b. To ensure that all variables are removed from the electronic workspace, the matlab code starts with clear all statement. The next step is to define the port that the Arduino nano is connected to, in our case its 'COM6', and the board type is 'Nano'. To initiate hardware communication, between Matlab and the Ardunio, we must call the board and select the interested library functions we are looking to perform, in our case 'Servo'. We must also indicate the Arduino pin that the motor is connected to for communication, in our case it is 'D9'.

```
clear all;
%Arduino details ----------------------------
port = 'COM6'; % Arduino connected port
board = 'Nano'; % model of your Arduino board
arduino_board = arduino(port, board, 'Libraries', 'Servo'); %include servo library
linear_motor = servo(arduino_board, 'D9'); % communicate with motor through digital pin 9
%--------------------------------------------
```

After initiating the communication link between Matlab and Arduino, we wanted to set the motor hand at a position that allows easy insertion of the PDMS membrane into the stretcher; thus we created a variable called release. The value given to release, 0.64, is the position of the motor when the PDMS should be inserted into the stretcher, at a relaxed non-stretched position. This value was selected based on PDMS size. The 0.64 position is also the position at which we punch holes into the PDMS, to insure a perfect insert. The writePosition(linear_motor, release); command is writing the position of the motor into D9 pin of the Arduino so it is received by the motor.

```
%punch holes into pdms at release=0.64
% to increase stretch, lower the value (< than 0.64)
release=0.64; % release - is non-stretched state of pdms

writePosition(linear_motor, release); % release PDMS x=0
```

At the next segment of the code, we wanted to ensure that the well bottom was completely flat prior to stretching. We noted that slight compressions in the x-axis of the PDMS due to mounting into the stretcher resulted in an uneven well bottom. Thus, we requested user input to modify the release position, E_release. To control this, the shaft of the motor must be retracted backwards, thus providing a value that is less than 0.64, and when wanting to extend the motor shaft, the value must be greater than 0.64. This section of the code continuously loops, until the user is satisfied with the well flatness. Once the user sets the preferred value, this value is sent through D9 pin of the Arduino, accordingly adjusting the motor shaft position.

```
% Modify PDMS start position
prompt = 'Check PDMS surface flatness, input starting position\n >';
E_release= input(prompt,'s');
E_release=str2double(E_release);

%Write motor new start position for pdms according to well flatness
writePosition(linear_motor, E_release);
prompt = 'is PDMS surface Flat Y / N \n >';
Flat= input(prompt,'s')

while E_r<1;

    if (strcmp(Flat,'N'))
       prompt = 'Provide new value to adjust PDMS surface flatness\n >';
       E_release= input(prompt,'s');
       E_release=str2double(E_release);
       writePosition(linear_motor, E_release);
       E_r=0;
       prompt = 'is PDMS surface Flat Y / N \n >';
       Flat= input(prompt,'s')
    elseif (strcmp(Flat,'Y'))
       E_r=1;
    else prompt = 'is PDMS surface Flat Y / N \n >';
```

```
        Flat= input(prompt,'s'); E_r=0;
    end

end
```

The next segment of the code is specifying the desired percent of arm movement. A prompt requests a percent value; however, this value cannot exceed the stretching limits of the PDMS. The code will reject any values above 26% to ensure that the well does not break, and will request a new value from the user. Once the value is within the acceptable range, the code will calculate the distance it needs to travel to achieve the percentage requested, this value will be stored in a variable labeled x2.

x2 is calculated based on the initial location of the motor E_release. x1 and x2 are the locations to which the motor shaft will retract and extend to in order to achieve the required percentage of stretch.

```
% Request user input on motor arm movement
prompt = 'Percent arm movement, arm movement cannot exceed 26%, otherwise PDMS
will break\n >';
Percent_Stretch= input(prompt,'s');
Percent_Stretch=str2double(Percent_Stretch)

while i<1;

    if (Percent_Stretch<=26)
        x1=E_release; x2=E_release-(E_release*(Percent_Stretch/100)); i=1;

    else prompt = 'Percent of arm movement cannot exceed 26%, PDMS will break \n >';
        Percent_Stretch= input(prompt,'s')
        Percent_Stretch=str2double(Percent_Stretch)
    end
end
```

After the user successfully enters the value that is accepted by the program, the program will request stretching patterns, i.e. cyclic or static. The user is requested to enter 1, for static and 2 for cyclic stretch. If the user inputs any other values, the program will prompt an INVALID REQUEST, and will request the user to enter either 1 or 2 as values. This value will be stored in a variable called type that will be recalled in the final segment of the code to initiate stretch according to stored variables.

```
% Request user on type of stretch required Fixed or cyclic
prompt = 'Select:    1-Fixed Stretch    2-cyclic Stretch    \n >';
Stretch_type= input(prompt,'s');
type=str2double(Stretch_type)
while l<1;
   if (type==1 | type==2);
      l=1;
   else prompt = 'INVALID REQUEST: Please Select:    1-Fixed Stretch    2-continuous Stretch
\n >';
      Stretch_type= input(prompt,'s');
      type=str2double(Stretch_type);
   end
end
```

The user will also be requested to input the duration of stretch in minutes. In theory, the stretch can be run for as long as the researcher needs. However, we have not tested our system beyond 15 hours and have set this as the upper limit in our program for now. Therefore, the code will not accept any duration beyond 900 minutes, 15 hours, or less than 5 minutes. Once the time duration is entered by the user, the value will be stored in a variable called Time.

```
prompt = 'Run Time: range (in 5- 900 min) Value in min (180 min - 3hr, 360 min - 6 hr, 540
min -9 hr, 720 min -12hr, 900 min-15 hr)\n >';
Time= input(prompt,'s');
Time=str2double(Time);
while h<1;
   if Time>=5 && Time<=900;
      h=1;
   else prompt = 'INVALID REQUEST: Please select Run Time: range (5-900 min)\n >';
   Time= input(prompt,'s');
   Time=str2double(Time);
   end
end
```

The frequency of cycles is the number of seconds the motor takes to achieve the desired retraction and extension of motor shaft (i.e. a complete cycle). The user is requested to input a range between 2 to 4; this value will be stored in a variable called n1. If the entered value is not within the provided range it will be considered invalid. We are implementing this code to slow down the motor shaft. The motor can move 1% (0.500mm) in approximately 370 milliseconds. By slowing it down, we are subjecting the cells to dynamic stretching activity for at least 2 seconds before returning the stretch back to its starting position.

```
prompt = 'Frequency: range (2-4 sec/cycle)\n >';
Frequency= input(prompt,'s');
n1=str2double(Frequency);

while g<1;
   if n1>=2 && n1<=4
      g=1;
   else prompt = 'INVALID REQUEST: please select Frequency: range (2-4 sec/cycle)\n >';
      Frequency= input(prompt,'s');
      n1=str2double(Frequency);
   end
end
```

This segment of the program also allows the user to hold on stretch, for example for 1 sec, in case the user is interested in capturing images as the cells are stretched. If the user is interested in holding on stretch, variable nh will be set to 1, otherwise 0. The program will take Y or N (in capital) as input, any other inputs will be rejected and the program loops until it receives an input it recognizes, Y or N.

```
 % Request user input on Holding Stretch ( YES or NO)
prompt = 'Hold on Stretch for 1 sec (Y/N)\n >';
```

```
hold= input(prompt,'s');
while k<1;
  if strcmp( hold,'Y')
    nh=1; k=1;
  elseif strcmp( hold,'N')
    nh=0; k=1;
  else prompt = 'INVALID REQUEST; Hold on Stretch for 1 sec (Y/N)\n >';
  hold= input(prompt,'s');
  end
end
```

After gathering all the input from the user, the program is now ready to run. This last portion of the code is where the program takes in all the variables set by the user and starts executing the code through the communication link with the motor.

```
% if user requested FIXED stretch hold stretch for the requested time
if strcmp( Stretch_type,'1')
  tic;
  while toc<=(Time*60); % the motor will hold stretch according to inputted time
    writePosition(linear_motor, x2);
  end
  writePosition(linear_motor, E_release);  % motor will release stretch after time
completed

% if user requested CYCLIC stretch hold stretch for the requested time
elseif strcmp(Stretch_type,'2')
  tic;
  while toc<=(Time*60); % Motor will stretch according to inputted time
    writePosition(linear_motor, x1);  % 1st Position of motor (extended)
    pause (n1) % n1 is frequency of cycle
    writePosition(linear_motor, x2); % 2st Position of motor (retracted)
    pause (n1+nh)  % n1 is frequency of cycle and nh if user wants to hold on stretch
  end
  writePosition(linear_motor, E_release); % Stop Motor in non-stretched state
  toc % Count the time completed, the value provided is in Sec
  Completed_time=toc/60 % convert time elapsed in min.
end
```

MATLAB code:

```
clear all;
%Arduino details ----------------------------
port = 'COM4'; % port at which your arduino is connected
board = 'Nano'; % model of your arduino board
arduino_board = arduino(port, board, 'Libraries', 'Servo'); %include servo library
linear_motor = servo(arduino_board, 'D9'); % communicate with motor through digital pin 9
%-------------------------------------------
n1=0; % n1 - pause between cycles
nh=0; % nh =0 if no hold requested on stretch, 1 when hold is requested
j=0;
k=0;
l=0;
h=0;
g=0;
E_r=0;
%punch holes into pdms at release=0.64
% to increase stretch decrease value lower than 0.64
release=0.64; % release - is non-stretched state of pdms
%Request user Data

%x1 - Non stretched PDMS- motor lever position
%x2 - Stretched PDMS- motor lever position

writePosition(linear_motor, release); % release PDMS x=0

% Modify PDMS start position
prompt = 'Check PDMS surface flatness, input starting position\n >';
E_release= input(prompt,'s');
E_release=str2double(E_release);

%Write motor new start position for pdms
writePosition(linear_motor, E_release);
prompt = 'is PDMS surface Flat Y / N \n >';
Flat= input(prompt,'s')

while E_r<1;

    if (strcmp(Flat,'N'))
      prompt = 'Provide new value to adjust PDMS surface flatness\n >';
      E_release= input(prompt,'s');
      E_release=str2double(E_release);
      writePosition(linear_motor, E_release);
      E_r=0;
      prompt = 'is PDMS surface Flat Y / N \n >';
      Flat= input(prompt,'s')
      elseif (strcmp(Flat,'Y'))
         E_r=1;
      else prompt = 'is PDMS surface Flat Y / N \n >';
         Flat= input(prompt,'s'); E_r=0;
   end

end

prompt = 'Percent arm movement, arm movment can not exceeding 30%, otherwise PDMS
will break\n >';
Percent_Stretch= input(prompt,'s');
Percent_Stretch=str2double(Percent_Stretch)

while i<1;

   if (Percent_Stretch<=30) %10.7%   x=2.33mm
      x1=E_release; x2=E_release-(E_release*(Percent_Stretch/100)); i=1;

   else prompt = 'Percent of arm movement can not exceed 30%, PDMS will break \n >';
      Percent_Stretch= input(prompt,'s')
      Percent_Stretch=str2double(Percent_Stretch)
   end
end
% Request user on type of stretch required Fixed or cylic
prompt = 'Select:     1-Fixed Stretch     2-cyclic Stretch    \n >';
Stretch_type= input(prompt,'s');
type=str2double(Stretch_type)
while l<1;
   if (type==1 | type==2);
      l=1;
   else prompt = 'INVALID REQUEST: Please Select:     1-Fixed Stretch     2-continuous Stretch
\n >';
      Stretch_type= input(prompt,'s');
      type=str2double(Stretch_type);
   end
end

prompt = 'Run Time: range (in 5- 720 min) Value in min (180 min - 3hr, 360 min - 6 hr, 540
min -9 hr, 720 min -12hr, 900 min-15 hr)\n >';
Time= input(prompt,'s');
Time=str2double(Time);
while h<1;
   if Time>=5 && Time<=900;
      h=1;
   else prompt = 'INVALID REQUEST: Please select Run Time: range (5-720 min)\n >';
   Time= input(prompt,'s');
   Time=str2double(Time);
   end
end

prompt = 'Frequency: range (2-4 sec/cycle)\n >';
Frequency= input(prompt,'s');
n1=str2double(Frequency);

while g<1;
   if n1>=2 && n1<=4
      g=1;
   else prompt = 'INVALID REQUEST: please select Frequency: range (2-4 sec/cycle)\n >';
      Frequency= input(prompt,'s');
      n1=str2double(Frequency);
   end
end
 % Request user input on Holding Stretch ( YES or NO)
prompt = 'Hold on Stretch for 1 sec (Y/N)\n >';
hold= input(prompt,'s');
while k<1;
   if strcmp( hold,'Y')
      nh=1; k=1;
   elseif strcmp( hold,'N')
      nh=0; k=1;
   else prompt = 'INVALID REQUEST; Hold on Stretch for 1 sec (Y/N)\n >';
   hold= input(prompt,'s');
   end
end
% if user requested FIXED stretch hold stretch for the requested time
if strcmp( Stretch_type,'1')
   tic;
   while toc<=(Time*60); % loop according to requested time by user in Sec
     writePosition(linear_motor, x2);
   end
   writePosition(linear_motor, E_release);
   % if user requested CYCLIC stretch hold stretch for the requested time
elseif strcmp(Stretch_type,'2')
   tic;
   while toc<=(Time*60); % loop according to requested time by user in Sec
     writePosition(linear_motor, x1);
     pause (n1) % Requested Hold time by user
     writePosition(linear_motor, x2);
     pause (n1+nh)
   end
   writePosition(linear_motor, E_release); % Stop Motor on non-stretched state
   toc % Count the time completed in Sec
   Completed_time=toc/60 % time elapsed in min.
end
```