

Supporting Information

for *Adv. Sci.*, DOI 10.1002/adv.202204479

Curved Nanofiber Network Induces Cellular Bridge Formation to Promote Stem Cell
Mechanotransduction

Qian Sun, Fang Pei, Man Zhang, Bo Zhang, Ying Jin, Zhihe Zhao and Qiang Wei**

Supporting Information

Curved Nanofiber Network Induces Cellular Bridge Formation to Promote Stem Cell Mechanotransduction

Qian Sun,^{a, b, +} Fang Pei,^{a, b, +} Man Zhang,^c Bo Zhang,^a Ying Jin,^{a, *} Zhihe Zhao,^a Qiang Wei^{b, c, *}

a. Department of Orthodontics, State Key Laboratory of Oral Diseases, National Clinical Research Center for Oral Diseases, West China Hospital of Stomatology, Sichuan University, Chengdu 610041, China.

b. College of Polymer Science and Engineering, State Key Laboratory of Polymer Materials and Engineering, Sichuan University, Chengdu 610065, China.

c. College of Biomedical Engineering, Sichuan University, Chengdu, 610065, China

Email: wei@scu.edu.cn (Q. W.), jinying@scu.edu.cn (Y. J.)

+ Q. S. and F. P. contributed equally to this work.

Specific surface area of the curved and straight fibers

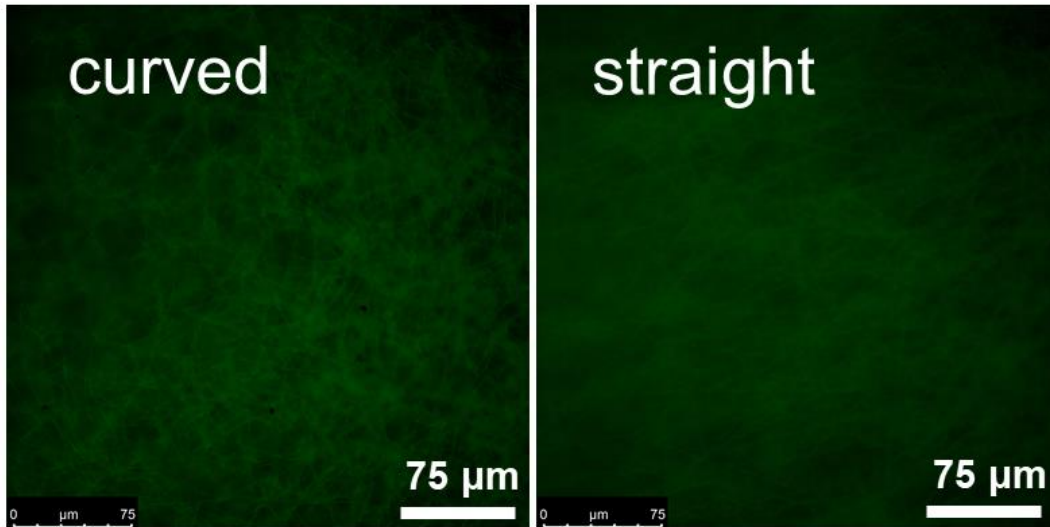


Figure S1. The fluorescent images of the FITC-BSA adsorbed on the curved and straight fibers.

The cell spreading of PDLSCs on the curved and straight fibers.

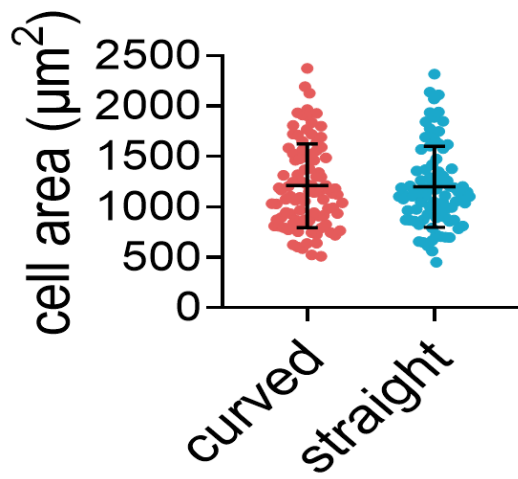


Figure S2. Cell spread area of the cell cultured on the related surfaces for 24 h (n = 100, 2 technical replicates).

Cells bridge on the curved fibers.

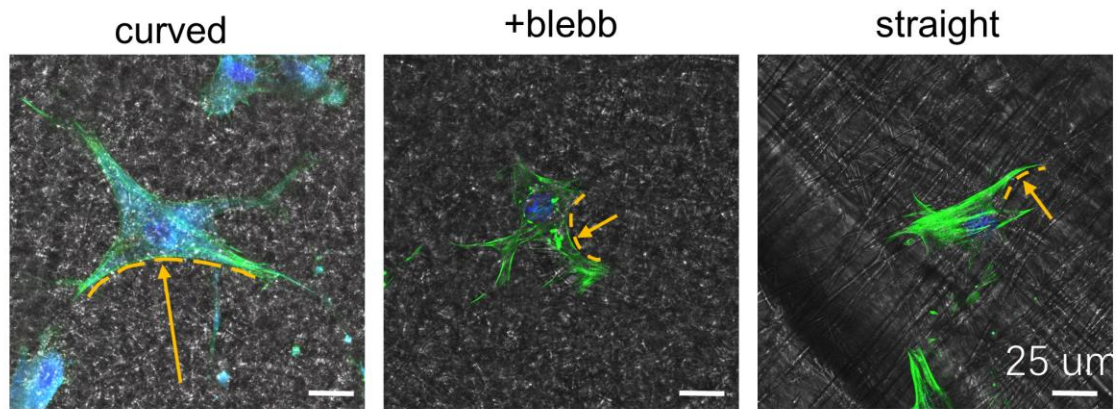


Figure S3. The representative images of PDLSCs on the curved and straight fibers. Green represents F-actin, blue represents nuclei. The dotted line defines the cell bridge and the yellow arrow indicates the radius of curvature.

The stress fiber tension drives cells to bridge.

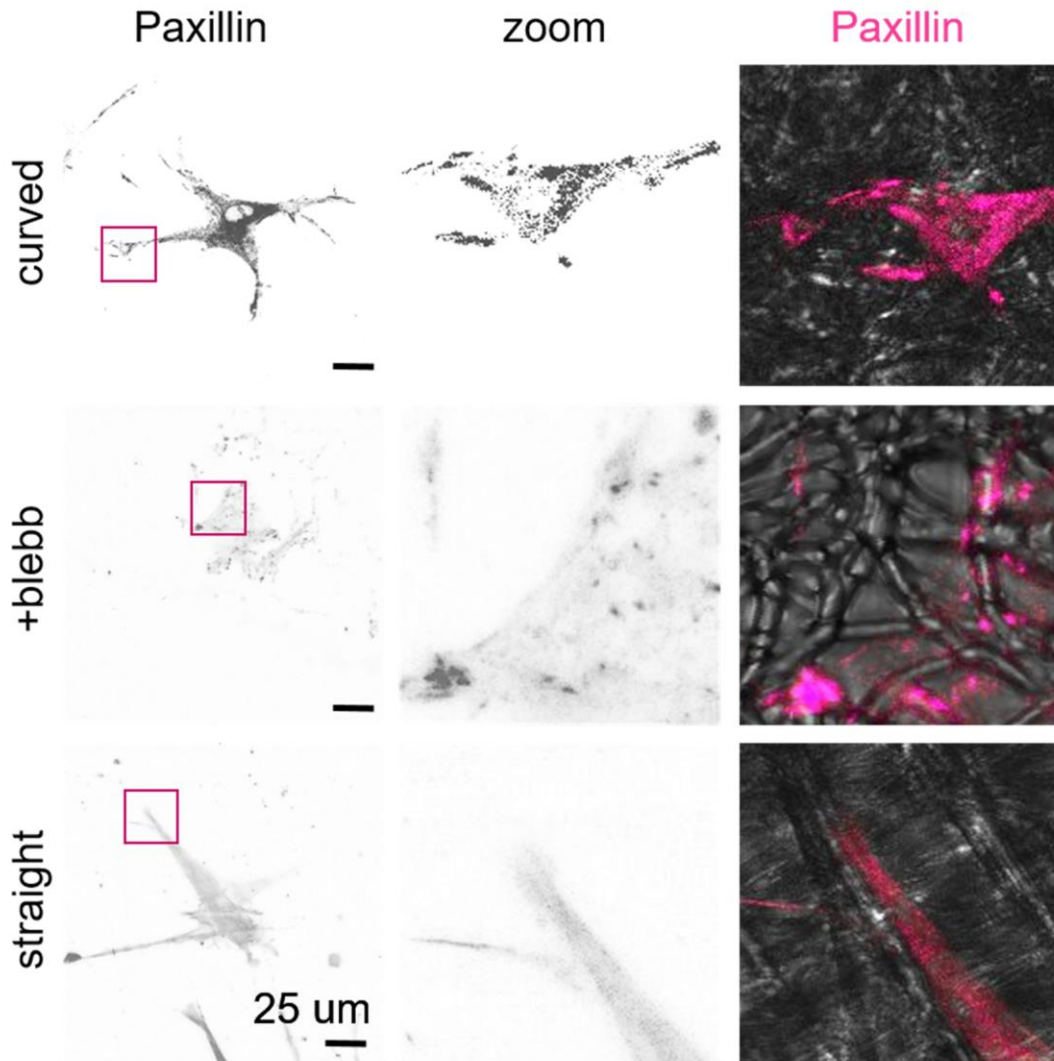


Figure S4. The representative fluorescence images of paxillin staining after 24 h of cell culture on the curved and straight fibers or on the curved fibers with blebbistatin (blebb, 20 μM) treatment.

Re-marked the edge of the cell adhesive and non-adhesive regions.

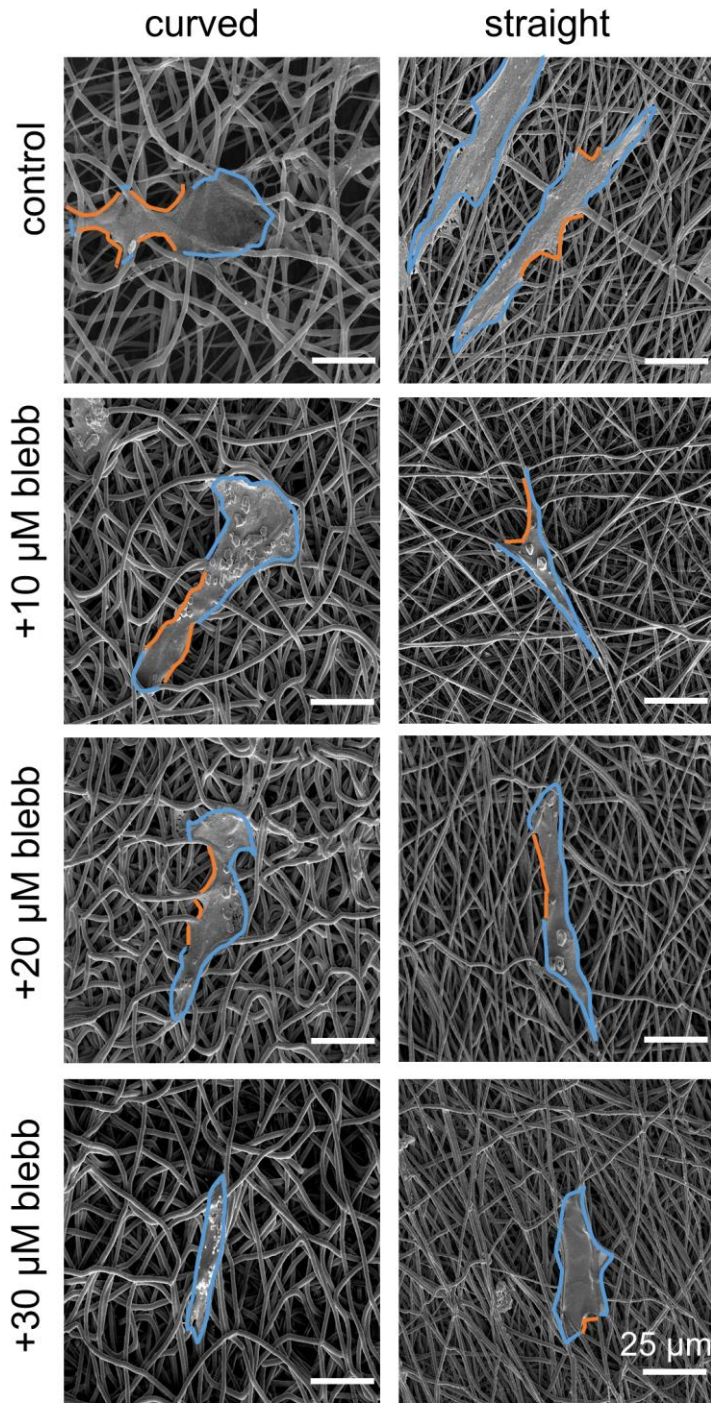


Figure S5. The representative SEM images of the single cell treated with blebbistatin on the curved and straight fibers, as indicating the adhesive edge in blue and non-adhesive edge in orange.

Cell bridges in the periodontal ligament.

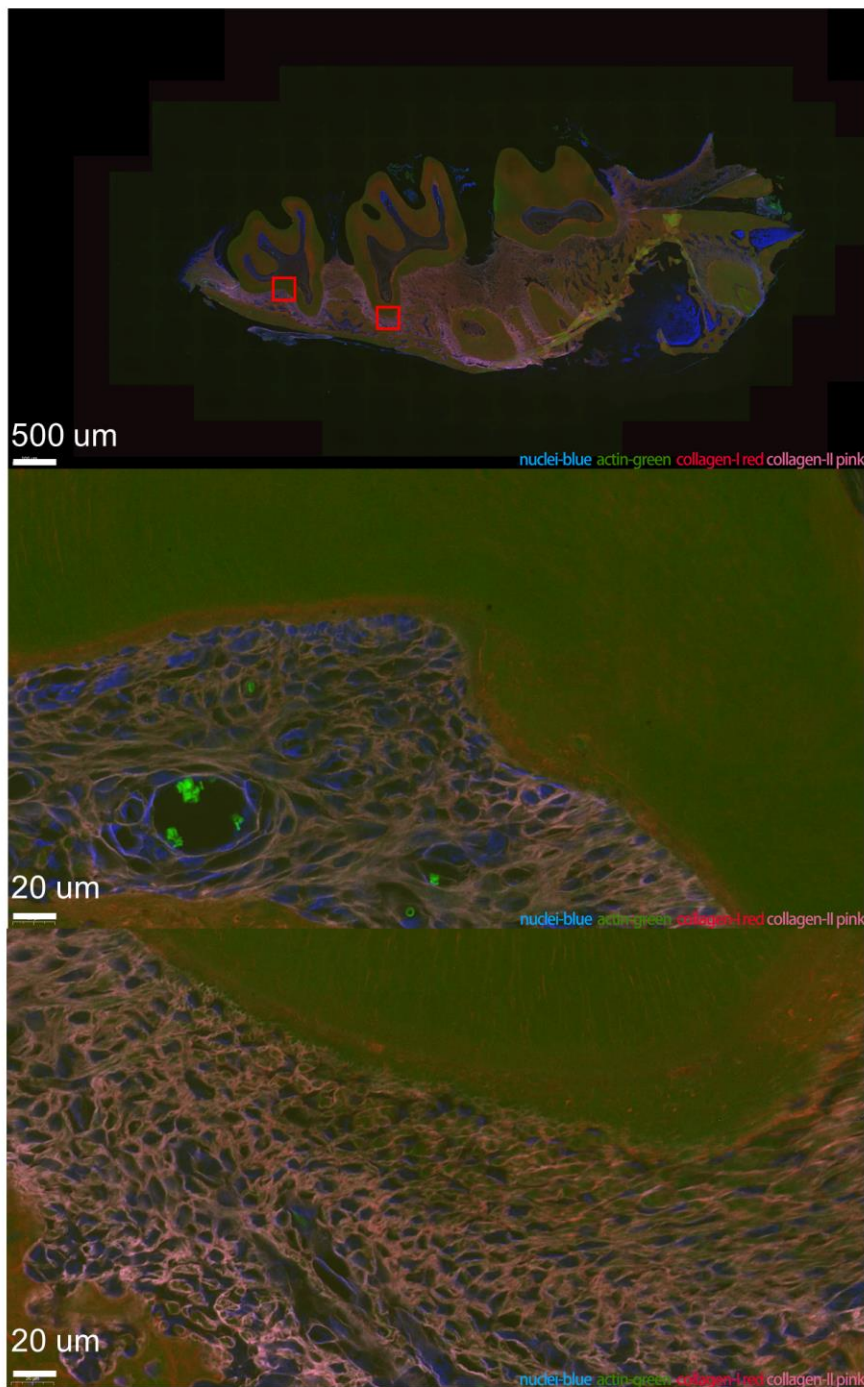


Figure S6. The representative fluorescence images of nuclei (blue), F-actin (green), collagen I (red) and collagen II (pink) staining of the rat periodontal ligament.

The differentiation potencies of the derived PDLSCs.

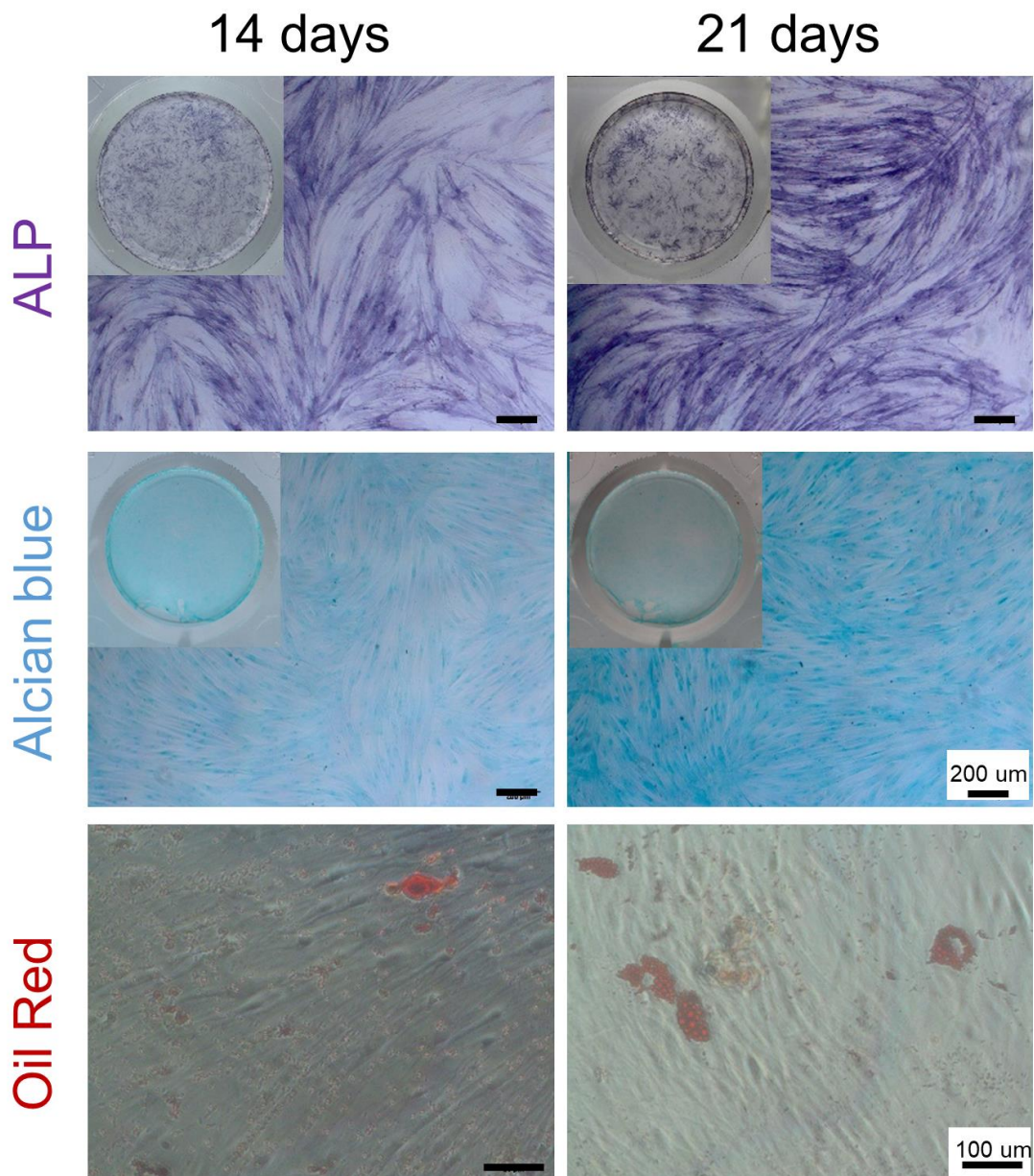


Figure S7. Cells derived from the healthy extracted premolars tissue are well differentiated towards osteogenesis, chondrogenesis, and adipogenesis.

Uncropped Western blots.

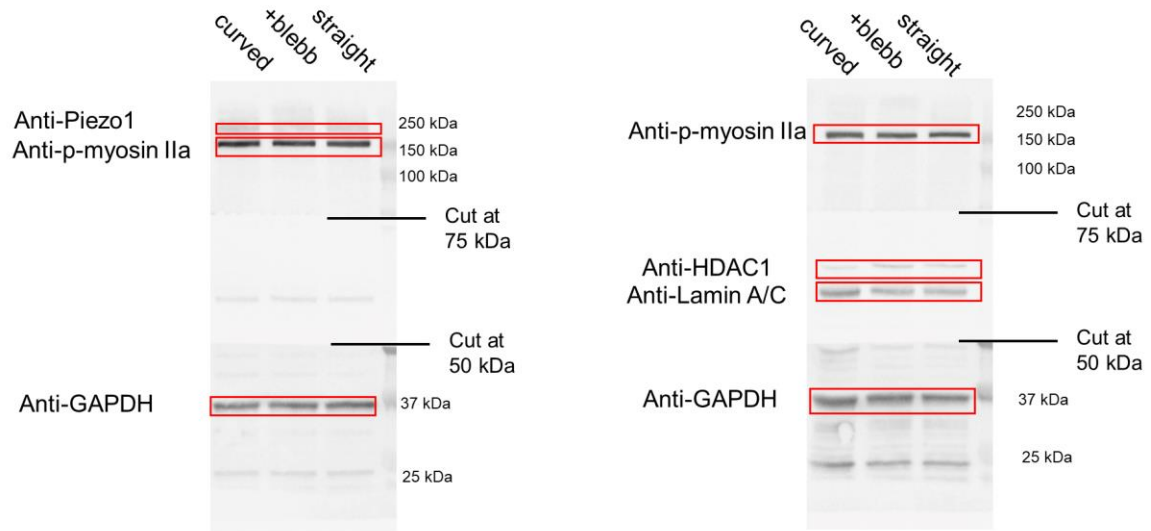


Figure S8. Full scan of uncropped membrane in the main figure. The membranes were cut at the indicated position before staining.

MATLAB Program 1. Representation of cell edge and curvature:

The main program:

```
1. clear
2. clc
3. x = xlsread('X-Coordinate (um)') % Each x coordinate of the scan point%
4. y = xlsread('Y-Coordinate (um)') % Each y coordinate of the scan point%
5. z = xlsread('Point Curvature (um-1)') % Each curvature coordinate of the scan point%
6.
7. a = mean(x)
8. b = mean(y)
9. plot3(x, y, z) % Draw 3D images%
10.
11. axis equal
12.
13. hold on
14. scatter(a,b)% Mark the center of the cell%
15. hold on
16. patch(x,y,z,z, 'edgecolor','flat', 'facecolor','none')% Give a gradient color to the curvature%
17. view(3); grid on; colorbar
18.
19. grid off
```

MATLAB Program 2. Representation of cell edge vertex:

The main program:

```
1. clear
2. clc
3.
4. z = xlsread ('Point Curvature (um-1)') % Each curvature coordinate o
   f the scan point%
5.
6. hold on
```

MATLAB Program 3. Laplace model:

The main program:

```
1. clear
2. clc
3. k = 5;%one actomyosin motor force,free parameter%
4. i = 1;%free.parameter,L0,fraction%
5. x = 2:0.1:75; %range of distance between one cell bridge,l%
6. n = numel(x)-1;
7. y = 2:98/n:100; %range of curvature radius,R%
8. y1 = 1./y;%count backwards %
9. y2 = transpose(y1);%transposition %
10.
11. %myosin density caculation%
12. m1 = 5;%myosin density free parameter 1%
13. m2 =1.2;%myosin density free parameter 2%
14. m3 = m1 * y.^m2;%myosin density, is a function of myosin density%
15. m4 = transpose(m3);
16. m = repmat( m4 ,1,n+1 );
17.
18. % Calculation process : R-d-F%
19. a1 = y2 * x;%1/R%
20. angle1 = 2 * asin(a1./2);
21. y3 = diag(y);%Create diagonal matrix%
22. c = y3 * angle1;
23. d = i./c;
24. F = 1./(1 + d);% ignore myosin density change,R-d-F%
25. F1 = F .* m;% include myosin density,R-d-F1%
26. F(imag(F)~=0) = nan;
27. F1(imag(F1)~=0) = nan;
28. %F Results: Horizontal direction, from left to right, R is constant, d increases gradually;%
29. %In the vertical direction, from top to bottom, d stays the same, and R increases%
30.
31. % Analysis : P :memberane contractile pressure%%
32. c1 = i./(angle1);
33. y4 = transpose(y);
34. y5 = repmat( y4 ,1,n+1 );
35. d1 = y5 + c1;
36. P = 1./d1;% ignore myosin density change,R-d-F%
37. P1 = P .* m;% include myosin density,R-d-F1%
38. P2 = F1 ./ y5;
39. P(imag(P)~=0) = nan;
40. P1(imag(P1)~=0) = nan;
41. P2(imag(P2)~=0) = nan;
```

```

42. %P Results: Horizontal direction, from left to right, R is constant
    , L increases gradually;%
43. %In the vertical direction, from top to bottom, l stays the same, a
    nd R increases%
44.
45. %Calculation process : R-angle-F,R-angle-P %
46. angle = 0.1 :(pi-0.1)/n :pi;
47. y6 = y.';
48. angle2 = y6 * angle;%x axi -angle, y axi - R%
49. angle3 = 1 + i./angle2;
50. angle4 = 1 ./ angle3;
51. F2 = angle4 .* m;
52. P3 = F2 ./ y5;
53. F2(imag(F2)~=0) = nan;
54. P3(imag(P3)~=0) = nan;
55.
56.
57. %plot%
58. %%plot%%
59. figure(1);
60. surf(x,y,F1,'FaceAlpha',0.7,'EdgeColor','none');
61. xlabel('d');
62. ylabel('R');
63. zlabel('F1');
64. %hold on
65. %image(x,y,F,'CDataMapping','scaled');
66. set(gca,'linewidth',2)
67.
68. figure(2);
69. surf(x,y,P2,'FaceAlpha',0.7,'EdgeColor','none');
70. xlabel('d');
71. ylabel('R');
72. zlabel('P2');
73. %hold on
74. %image(x,y,P,'CDataMapping','scaled');
75. set(gca,'linewidth',2)
76.
77. figure(3);
78. surf(x,y,F1,'FaceAlpha',0.7,'EdgeColor','none');
79. xlabel('d');
80. ylabel('R');
81. zlabel('F1')
82. %hold on
83. %image(x,y,F1,'CDataMapping','scaled');
84. hold on;
85. set(gca,'linewidth',2)
86.

```

```

87. % define R/d=0.7 contour %
88. v=[0.7,0.7];
89. vx = repmat( x ,n+1,1 );
90. v1=y5./vx
91. [~,c]=contour(x,y,v1,v);
92. c.LineWidth = 4;
93. c.LineColor = [0 0.4492 0.4180];
94. hold on;
95.
96. R_curved = xlsread('curved_straight.xlsx', 'Sheet1', 'A3:A55')
97. d_curved = xlsread('curved_straight.xlsx', 'Sheet1', 'B3:B55')
98. curved = scatter(d_curved,R_curved,'filled','MarkerFaceColor',[.8945
    .3555 .3555]);
99. curved.SizeData = 120;
100. uistack(c);
101. hold on
102. R_straight = xlsread('curved_straight.xlsx', 'Sheet1', 'D3:D55')
103. d_straight = xlsread('curved_straight.xlsx', 'Sheet1', 'E3:E55')
104. straight = scatter(d_straight,R_straight,'filled','MarkerFaceColor',
    [.2578 .7109 .9297]);
105. straight.SizeData = 120;
106. uistack(c);
107. hold on;
108. R_blebb = xlsread('curved_straight.xlsx', 'Sheet1', 'G3:G55')
109. d_blebb = xlsread('curved_straight.xlsx', 'Sheet1', 'H3:H55')
110. blebb = scatter(d_blebb,R_blebb,'filled','MarkerFaceColor',[.5742 .8
    438 .2266]);
111. blebb.SizeData = 120;
112. uistack(c);
113. hold on;
114.
115.
116. figure(4);
117. surf(x,y,P2,'FaceAlpha',0.7,'EdgeColor','none');
118. xlabel('d');
119. ylabel('R');
120. zlabel('P2');
121. %hold on
122. %image(x,y,P1,'CDataMapping','scaled');
123. set(gca,'linewidth',2);
124. hold on;
125.
126. [~,c]=contour(x,y,v1,v);
127. c.LineWidth = 4;
128. c.LineColor = [0 0.4492 0.4180];
129. hold on;
130. R_curved = xlsread('curved_straight.xlsx', 'Sheet1', 'A3:A55')
131. d_curved = xlsread('curved_straight.xlsx', 'Sheet1', 'B3:B55')

```

```
132. curved = scatter(d_curved,R_curved,'filled','MarkerFaceColor',[.8945
    .3555 .3555]);
133. curved.SizeData = 120;
134. uistack(c);
135. hold on
136. R_straight = xlsread('curved_straight.xlsx', 'Sheet1', 'D3:D55')
137. d_straight = xlsread('curved_straight.xlsx', 'Sheet1', 'E3:E55')
138. straight = scatter(d_straight,R_straight,'filled','MarkerFaceColor',
    [.2578 .7109 .9297]);
139. straight.SizeData = 120;
140. uistack(c);
141. hold on;
142. R_blebb = xlsread('curved_straight.xlsx', 'Sheet1', 'G3:G55')
143. d_blebb = xlsread('curved_straight.xlsx', 'Sheet1', 'H3:H55')
144. blebb = scatter(d_blebb,R_blebb,'filled','MarkerFaceColor',[.5742 .8
    438 .2266]);
145. blebb.SizeData = 120;
146. uistack(c);
147. hold on;
148.
149. figure(5);%merge F1 and P1 %
150. surf(angle,m,P3,'FaceAlpha',0.7,'EdgeColor','none');
151. xlabel('angle');
152. ylabel('m (myo)');
153. zlabel('P3')
154. set(gca,'linewidth',2);
155.
156. figure(6);
157. surf(angle,y,F2,'FaceAlpha',0.7,'EdgeColor','none');
158. xlabel('angle');
159. ylabel('R');
160. zlabel('F2');
161. %hold on
162. %image(x,y,P1,'CDataMapping','scaled');
163. set(gca,'linewidth',2)
```


MATLAB Program4. Canny cell edge test:

The main program:

```
1. clc
2. clear
3. actin = imread('actin.tif');
4. actin = rgb2gray(actin);
5. actin_BW1= edge(actin,'canny');
6. acti_gray= double(actin_BW1);
7. collagen = imread('collagen.jpg');
8. collagen = rgb2gray(collagen);
9. collagen_BW1= edge(collagen,'canny');
10. coll_gray= double(collagen_BW1);
11.
12. %actin edge%
13. I=acti_gray;
14. I(I<0) = 0;
15. I(I==0)=NaN;
16. figure(1);
17. x=[1:1:2048];
18. y=[1:1:2048];
19. surf(x,y,I);
20. view(2);
21.
22. % collagen edge%
23. figure(2);
24. P = coll_gray;
25. P(P<1) = 0;
26. P(P==0)=NaN;
27. figure(2);
28. surf(x,y,P);
29. view(2);
```

Table S1. Primer details for qRT-PCR.

H-GAPDH	H-GAPDH-F	GGAGCGAGATCCCTCCAAAAT
	H-GAPDH-R	GGCTGTTGTCATACTTCTCATGG
H-ALP	ALP-F	GTGAACCGCAACTGGTACTC
	ALP-R	GAGCTGCGTAGCGATGTCC
H-RUNX2	RUNX2-F	TGGTTACTGTCATGGCGGGTA
	RUNX2-R	TCTCAGATCGTTGAACCTTGCTA
OCN	OCN-F	CACTCCTCGCCCTATTGGC
	OCN-R	CCCTCCTGCTTGGACACAAAG