# Supplementary Information: 3D printing with nucleic acid adhesives

Peter B. Allen, Zin Khaing, Christine E. Schmidt, and Andrew D. Ellington*

**Contents:**

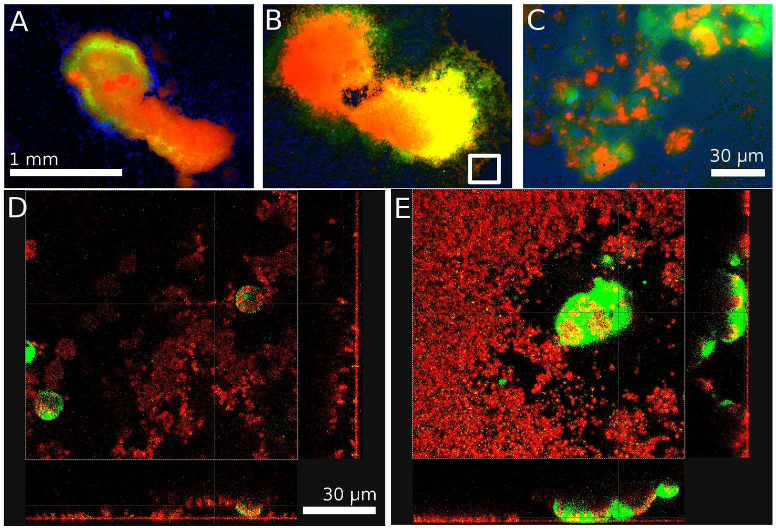

**Figure S1.** HEK cells grow within the DNA-assembled polystyrene colloidal gel. (A) A false colored epifluorescence micrograph shows brightfield (blue), GFP fluorescence (green), and Cy5 fluorescence (red). Self-assembled colloidal gel generated from Cy5-labeled polystyrene microparticles incorporates HEK cells expressing GFP. Scale bar indicates 1 mm. (B) False colored micrograph with the same scale shows the same mass after 4 days. The margin that was imaged at higher magnification is indicated with a white box. (C) The magnified region at the margin of the mass shows healthy, proliferating green cells coated in some cases with the red colloidal gel. Scale bar is 30 µm. (D) A scanning confocal microscope image of the same cell and bead types. Scale bar is 30 µm. Cells (green) are situated within the colloidal mass (red). (E) An image of non-complementary microparticles with occasional cells on the glass surface.

**Results:** *GFP-labeled cells proliferate within the colloidal gel:* HEK 293T cells expressing GFP grow and proliferate within the colloidal gel. These cells were trypsinized from a culture flask then mixed with a X type particles and then subsequently mixed with X* particles. The whole suspension formed a colloidal gel that was extruded with a pipette tip into a ~1 mm agglomerate within tissue culture media (see **Figure S1A**). This agglomerate was allowed to incubate for several days. The increased green signal (relative to red) in **Figure S1B** indicates growth and proliferation of GFP expressing cells within the mass. At the margin (seen under high magnification in **Figure S1C**) healthy cells can be seen to have grown out of the mass; some are coated with red polystyrene particles. A second sample of the cells in the colloidal gel was extruded into a chamber slide and imaged on a confocal microscope (**Figure S1D**). Only the margin could be successfully imaged with the confocal microscope due to the high scattering from the colloidal particles. Nonetheless, the variegated pattern of the self-assembled internal gel structure (red) can be seen in 2D and the cross sections. The cells (green) are embedded within that structure. The non-complementary particles simply settle to the surface (**Figure S1E**); cells were visible in the relatively particle-free areas.

**Methods:** *Tissue culture and incorporation of GFP-labeled cells into the colloidal gel:* complementary bead types were generated as described in **Materials and Methods**. Human embryonic kidney (HEK-393T) cells expressing GFP were cultured according to standard protocols. Briefly, cells were cultured on Corning T75 flasks in Dulbecco's modified Eagle's medium (DMEM, Thermo Fisher) supplemented with FBS (fetal bovine serum) to 5%, 100 u/ml penicillin and 100 µg/ml streptomycin. Cells were passaged by detaching with .25% trypsin-EDTA. Freshly detached cells were rinsed with PBS then mixed with bead type X. Bead type X* was then added to the suspension. Cell/bead slurry was then slowly pipetted into a micro-well pre-filled with media and allowed to incubate at 37°C with 5% carbon dioxide. Cells within the colloidal aggregate were examined under a confocal microscope (Leica SP2 AOBS Confocal, Solms, Germany) on day 2 or epifluorescence microscope (Olympus IX, Shinjuku, Tokyo) on day five.

```
function totalNucleations = assemblySimulation(proportion)
%%This MATLAB function is called with an argument [proportion]
% and runs an assemlby simulation with taking the argument
% as the fraciton of (0-1, default .32) small particles
% and return totalNucleation, a count of the nucleation events.

%%The simulation should generate particles in the proper proportion
%%then dock those particles onto each other in non-colliding
%%configurations.

%initialize
%set some proportion of the particles to a different identity
if nargin==0
    proportion=.32; %from experiment, .32, .24, .19, .16
end

%initialize geometry parameters
R2=2.3/2;
R1=1/2;
SizeOfArea=40;

%calculate number to get to 40% coverage
averageAreaPerParticle=((proportion*(3.14*((R2)^2)))+((1-proportion)*(3.14*((R1)^2))));
ParticleCount=int64(SizeOfArea^2/averageAreaPerParticle)*.4;

%initialize particle parameters
ParticlesX=zeros(ParticleCount,1);
ParticlesY=zeros(ParticleCount,1);
ParticlesI=ones(ParticleCount,1);


finalpoint=round(ParticleCount*proportion);
ParticlesI(1:finalpoint)=2;
%randomize the order
ParticlesI=ParticlesI(randperm(ParticleCount,ParticleCount));

%use this identity order to generate the two sizes
for i=1:ParticleCount
    if ParticlesI(i)==1
        ParticlesR(i)=R1;
    else
        ParticlesR(i)=R2;
    end
end

% Initialize the first particle manually
ParticlesX(1)=round(SizeOfArea/2);
ParticlesY(1)=round(SizeOfArea/2);
totalNucleations=1;

%dock each successive particle
for i=2:ParticleCount

    collision=true;
    attemptCount=0; %reset attempt counter for max tries
    while (collision==true && attemptCount<100)
        %if it's colliding and not over a max, then try again and again to
        %connect the next circle to a random existing circle

        atteptConncetionTo=randi([1,i-1]);
```

```
[offsetx,offsety]=pol2cart(6.282*rand(),(ParticlesR(i)+ParticlesR(atteptConncetionTo)));%(angle, radius)
          ParticlesX(i)=ParticlesX(atteptConncetionTo)+offsetx;
          ParticlesY(i)=ParticlesY(atteptConncetionTo)+offsety;
          %start by assuming no collision
          collision=false;
          if ParticlesI(i)==ParticlesI(atteptConncetionTo)
              collision=true; %particles only join to non-same particles
          else
              %check for collision with each previous particle
              for j=1:i-1
                  if (abs(ParticlesX(i)-ParticlesX(j)) < ParticlesR(i)+ParticlesR(j) &&...
                      abs(ParticlesY(i)-ParticlesY(j)) < ParticlesR(i)+ParticlesR(j) )

                  %If center-to-center distance is < 2 radius...
                      if hypot(abs(ParticlesX(i)-ParticlesX(j)),...
                              abs(ParticlesY(i)-ParticlesY(j)))<=(ParticlesR(i)+ParticlesR(j))

                          collision=true; %any collision is sufficient to flag

                      end
                  end
              end
          end
          %it is possible (but unlikely) to get stuck; set a counter so that
          %if the random offset and conneciton routine will run a max number
          attemptCount=attemptCount+1;
      end
      if(collision==true)
          ParticlesX(i)=SizeOfArea*rand();
          ParticlesY(i)=SizeOfArea*rand();
          totalNucleations=totalNucleations+1;
      end

  end

  colors=[[0 1 0];[1 0 0]];

  %fig1 = figure();
  %hold on;
  clf;
  for i=1:int64(ParticleCount)
      circle(ParticlesX(i),ParticlesY(i),ParticlesR(i),...
          (double(i)/(double(ParticleCount)+5))*colors(ParticlesI(i),:) );
  end
  title(['Proportion Large: ', num2str(round(proportion*100)), '% Nucleii: ',
num2str(round(totalNucleations))]);
  axis equal;
  axis([0,SizeOfArea,0,SizeOfArea]);
  %set(fig1, 'Color', [1 1 1]);
  print('-dtiffn','-r150',['result_image_', num2str(round(proportion*100)), '_',
num2str(round(SizeOfArea))]);
  %hold off;

  %
  % orig_mode = get(fig1, 'PaperPositionMode');
  % set(fig1, 'PaperPositionMode', 'auto');
  % cdata = hardcopy(fig1, '-Dzbuffer', '-r0');
  % % Restore figure to original state
  % set(fig1, 'PaperPositionMode', orig_mode); % end

  end
```

```
function h=circle(x,y,r,c)
    x=x-r;
    y=y-r;
   h=rectangle('Position',[x,y,r*2,r*2],...
  'Curvature',[1,1], 'FaceColor',c);
end
```