

University of Groningen

Self-assembled structures and applications of DNA hybrid materials

Kwak, Min Seok

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2011

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Kwak, M. S. (2011). *Self-assembled structures and applications of DNA hybrid materials*. s.n.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Matlab source code for random particle generator (in page 65)

```

1 % Random particle generator runs on Matlab2009a
2 % as a part of a manuscript "DNA-Functionalised Blend Micelles: Mix
  and Fix Polymeric Hybrid Nanostructures"
3 % by M. Kwak, A. J. Musser, J. Lee and A. Herrmann
4 % submitted to Chem. Commun. (Royal Chemical Society) in April, 2010.
5 % sourcecode of rparticle.m
6
7 noPixels = 1024;      % image size (px)
8 noParticles = 722;   % number of particles in a square
9 sizeNP = 5.0;       % size of the used particle (nm)
10 imageLength = 915.75; % real size of the image calculated with
  magnification of microscope (nm)
11 tableD = zeros(noParticles); % empty matrix for table of distances
12 tableCount = zeros(1,11); % empty matrix for counts
13 cdnt = floor(noPixels * rand(2, noParticles)); % random x,y
  coordinates in the defined area
14 maxD = sqrt(2)* noPixels; % diagonal length of the image (px)
15
16 % calculate all distances between possible 2 (x,y) coordinates
17 for i=1:noParticles
18     for j=1:noParticles
19         tableD(i,j)=sqrt(power(cdnt(1,i)-cdnt(1,j), 2) +
  power(cdnt(2,i)-cdnt(2,j), 2));
20         % replace the distance of itself (0 px) to the maximum
  diagonal distance
21         if tableD(i,j) == 0
22             tableD(i,j) = maxD;
23         end
24     end
25 end
26
27 % find and save the d_NN (center to center, pixel) of each
  coordinates
28 tableMinD = min(tableD, [], 1);
29
30 % count particles in range
31 for i=1:noParticles
32     % unit conversion after applying size of the NP (pixel to nm)
33     l = imageLength / noPixels * tableMinD(i) - sizeNP;
34     % count the d_NN distribution
35     if l >= 20
36         tableCount(11)=tableCount(11)+1;
37     elseif l > 18
38         tableCount(10)=tableCount(10)+1;
39     elseif l > 16
40         tableCount(9)=tableCount(9)+1;
41     elseif l > 14
42         tableCount(8)=tableCount(8)+1;
43     elseif l > 12
44         tableCount(7)=tableCount(7)+1;
45     elseif l > 10
46         tableCount(6)=tableCount(6)+1;
47     elseif l > 8
48         tableCount(5)=tableCount(5)+1;
49     elseif l > 6
50         tableCount(4)=tableCount(4)+1;
51     elseif l > 4
52         tableCount(3)=tableCount(3)+1;
53     elseif l > 2
54         tableCount(2)=tableCount(2)+1;
55     else
56         tableCount(1)=tableCount(1)+1;
57     end
58 end
59
60 % draw a graph
61 bar(tableCount);

```

