Biophysical Journal, Volume 99

Supporting Material

Two-dimensional continuum percolation threshold for diffusing particles of nonzero radius

Michael J. Saxton

June 13, 2010

SUPPORTING MATERIAL

Two-dimensional continuum percolation threshold for diffusing particles of nonzero radius

Biophysical Journal, September 2010

Michael J. Saxton

PROGRAMMING METHODS

Here we discuss the specifics of the methods described in the text. For a general review of simulation techniques for diffusion, see Saxton (1).

The overall structure of the program is as follows. A bash script executes a loop over obstacle configurations, in which

1. a Fortran program generates equilibrium obstacle positions by Brownian dynamics or Monte Carlo methods;

2. a qhull and Mathematica program does the Voronoi analysis of the obstacles;

3. a Fortran program finds the percolation threshold.

At the end of the loop, a Fortran program finds the cumulative distribution function (CDF) of the thresholds and analyzes it. Programs communicate by text files. This approach is simple and modular; it generates coordinate files of equilibrated positions so that the data can be reanalyzed if necessary; and it avoids having to call Mathematica from Fortran or vice versa. The goal is to minimize hassle, not to maximize elegance.

Obstacle configurations

To generate obstacle configurations by Brownian dynamics, obstacles are placed on the smallest square grid that contains at least the prescribed number of obstacles, and obstacles are randomly deleted until the prescribed number is reached. For soft repulsive disks, the system is then annealed by standard Brownian dynamics (2) until the energy is constant. For hard disks, the system is annealed by standard Metropolis Monte Carlo moves (2) and equilibration is measured by the radial distribution function. For each obstacle configuration, a fresh set of obstacles is generated from the beginning to ensure the independence of the configurations. Periodic boundary conditions are used. The ran2 random number generator (3) is used, modified for inline use.

The calculations are done in physical units, not dimensionless units. The diffusion coefficient of the obstacles is 0.1 $\mu\text{m}^2/\text{s} = 0.0001\ \text{nm}^2/\text{ns}$. The obstacle diameter is $\sigma = 4$ nm, and the time increment is 0.5 ns. The Verlet neighbor list is updated every 32 time steps. Equilibration times range between 16×1024^2 time steps for the lowest concentrations to $\frac{1}{2} \times 1024^2$ for the highest.

System sizes

Percolation thresholds are defined in the limit of an infinite system, so it is essential to vary the system size. Here the calculations were carried out for a range of relative system sizes L_r for each obstacle concentration. The starting system, designated $L_r = 1.0$, had number density $24\text{k}/\mu\text{m}^2$ (1k = 1000), physical system size $L = 100$ nm, and 240 obstacles. From this we generated the base systems at each number density 4k, 8k, 16k, 24k, 32k, 40k, 48k by changing the physical system size at an approximately constant number of obstacles. At each number density, we generated a size series $L_r = 0.5$, 0.75, 1.0, 1.5, 2.0, 2.5, and 3.0 (and sometimes larger) by multiplying the base system size by L_r and the base number of obstacles by L_r^2, and rounding as required. The ratio of

the actual number density to the nominal was at worst 1.008; typically it was much closer (mean 1.0002, SD 0.0022). For the 24k series, for the $L_r = 0.5$ and 3.0 runs, physical system sizes were 50 and 300 nm respectively, and the numbers of obstacles were 60 and 2160. All the runs at a given value of L_r had approximately the same number of obstacles, regardless of number density. Quantities related to the system size for $L_r = 1$ are given in Supporting Table S1.

Most runs used 100 independent obstacle configurations. For the runs with $L_r = 1.0$ and below, and a few larger systems, 500 or 1000 configurations were used to get a smoother CDF. In all cases percolation thresholds were found in each direction for each obstacle configuration.

Table S1: System sizes for the $L_r = 1$ series

Number density $1/\mu\text{m}^2$	Area fraction	System size nm	Number of obstacles
4000	0.05026	244	238
8000	0.10053	174	242
16000	0.20106	122	238
24000	0.30159	100	240
32000	0.40212	86	237
40000	0.50266	78	243
48000	0.60319	70	235

Voronoi analysis

The Voronoi analysis of obstacle positions was first done in Mathematica V5.0 to take advantage of its graphics and analysis capabilities (4). This version was too slow for production runs so the program was revised to use qhull (5, 6) via the mPower (7) Mathematica interface. These programs are much faster, and retain the capabilities of Mathematica. The mPower output is parsed using Mathematica.

To test for percolation one must define north-south (NS) and east-west (EW) conduction precisely. The obstacles are within a square, so the diagonals of the square are used to define the N,S,E,W quadrants and the convex hull of the obstacles is used to define in and out. Using the convex hull is physically reasonable because it is where obstruction begins to have an effect. Specifically, for each Voronoi bond crossing the convex hull, the intersection of the bond with the convex hull is found, and the bond is classified as N,S,E,W depending on the quadrant in which the intersection occurred. This approach complicates the program but eliminates arbitrariness in assignments when a Voronoi bond crosses a diagonal far outside the convex hull.

For simplicity, periodic boundary conditions were not used in the Voronoi analysis, except in one series of test runs.

To avoid an occasional problem of having a few Voronoi vertices at extremely large distances, boundary points are generated using a program option in mPower. This option generates a set of extra points outside the convex hull of the original points on a outer polygon geometrically similar

to the convex hull and approximately 1.4× its radius. The extra points are roughly equally spaced along the outer polygon, with a small random component. The Voronoi diagram is constructed for the set of obstacles plus extra points, and then the extra points and the Voronoi bonds outside the outer polygon are dropped. This program feature was useful but it sometimes gives Voronoi bonds that extend from one edge across the entire figure. It is necessary to test for this and try a new set of random extra points on the outer polygon when this problem occurs.

Percolation thresholds

The percolation threshold is found using a bisection routine (8, 9) adapted from Press et al. (3) using the standard Hoshen-Kopelman algorithm (10) to test for bond percolation. For each obstacle configuration this algorithm gives upper and lower limits for percolation, and the mean is used. Thresholds for both NS and EW percolation are used. Clearly the thresholds are not independent – sometimes a single bond controls both NS and EW percolation and the thresholds coincide – but using both is the most efficient use of relatively expensive equilibrated configurations. The output is a list of mean percolation thresholds, typically 100 NS and 100 EW. These are sorted and the CDF is obtained directly from the sorted list. This method has the advantage over standard histograms that there is no need to impose a bin size and no information is discarded in the binning (11–14).

Determination of thresholds from CDFs

To obtain the percolation threshold, the midpoint of the CDFs must be extrapolated to infinite system size. Several methods to determine the midpoint of the CDFs gave very similar results. (a) Calculate the mean value of the distribution directly. (b) Read the 50% point directly from the CDF. (c) To provide some smoothing, fit a straight line to the linear region of the CDF around the 50% point (typicaly > 50) points) and calculate the midpoint from the best-fit line. (d) Fit an incomplete beta function to the entire CDF and determine the 50% point, as done in earlier work (15). (A potential problem with the last method is that one must assume some functional form, and a least-squares fit to a particular function might distort the relevant behavior near the 50% point in an attempt to fit the less relevant noisy ends to an approximate function. Becker and Ziff (16), however, had good results using various assumed functional forms to analyze their highly accurate data.) The mean of the distribution was used routinely. In all these methods, the midpoint is extrapolated to infinite L by plotting the midpoint versus $1/L^{\nu}$, where ν is a scaling exponent discussed later. Alternatively one might use the intersection points of the CDFs as estimates of the threshold (see Saxton (15) for discussion and references). On the scale of text Fig. 3, this approach looks plausible, but at higher magnification, the intersections are more scattered than one would like.

Error estimates

The percolation diameter for a given set of conditions is highly reproducible. For six independent calculations at a number density of $24\text{k}/\mu\text{m}^2$, system size 150 nm, and 540 obstacles, the percolation

diameter was 6.495 ± 0.012 (mean $\pm$ SD), and the width of the distribution was 0.205 ± 0.008.

To choose the method of extrapolation to infinite system size, we fitted the data for random overlapping disks in various ways, and chose the one that gave the best agreement with the known value. The best agreement was from a least-squares fit to data for $L_r \geq 1.5$. Values from fitting $L_r \geq 1$ and $L_r \geq 2$ were used to estimate the error bars. We take the standard deviation of the percolation diameters to be 0.8%, a value that puts 15 out of 21 test runs within 1 standard deviation of the theoretical value.

Another test – an informal runs test – compared the percolation diameters in systems of various sizes. In the calculations varying V_0, the data points for each size were consistently in the order $V_0 =$ 1, 2, and 4, as were the extrapolated values, consistent with a real difference. In the calculations varying n, the data points for $n = 6$ were consistently different from those for $n = 12$ and $n =$ 18, but the latter two sets were intertwined so not significantly different. For periodic versus nonperiodic boundary conditions, the thresholds for the periodic case were always higher than those for nonperiodic. The extrapolated values were likewise different but by a very small amount.

Excluded areas

In the physics literature, percolation thresholds are usually expressed in terms of area fractions. We evaluated the excluded area numerically for the same obstacle configurations used in the percolation calculations. From this we obtained CDFs of excluded area as a function of tracer diameter, and we used these excluded area CDFs to express the CDFs of the percolation thresholds in terms of the excluded area. The area percolation thresholds of text Table 1 were obtained similarly.

The excluded area was determined numerically using quasirandom points (3), a method that proved very useful in determining the area covered by a random walk of a particle of nonzero size, the so-called "Wiener sausage" (17). Quasirandom points are more regularly spaced than random points but less regularly spaced than a grid.

The most fundamental method of calculating the excluded area is to draw circles of prescribed radius R around the obstacle centers, place a set of quasirandom points over the entire system, and count the quasirandom points inside the circles. The fraction of points inside is the excluded area fraction for that R. But this calculation must be repeated for each value of R. A better way is to find the nearest-neighbor obstacle for each quasirandom point, and calculate their separation r_{NN}. Now r_{NN} is the obstacle radius at which that quasirandom point moves from outside the obstacle to inside, so one can simply compile the CDF of the r_{NN}, and the CDF is equal to the excluded area fraction. For simplicity of programming, nearest neighbors were found using the cell index method (2, pp. 149–151) but not using linked lists because they have poor locality of reference in memory.

To express the percolation CDF $f(d)$ in terms of the area fraction, we define

$$f(d) = CDF_{\text{perc}}(d) \tag{S1}$$

$$g(d) = CDF_{\text{area\_fr}}(d) = area\_fr, \tag{S2}$$

where d is the diameter. We invert $g(d)$ numerically using the interpolation function in Mathematica

$$d = g^{-1}(area\_fr) \tag{S3}$$

and then use this function to find the final result

$$CDF_{\rm perc}(area\_fr) = f[g^{-1}(area\_fr)]. \tag{S4}$$

SCALING ANALYSIS

We present a scaling analysis of the percolation data to relate our work to the physics literature and to show that the percolation problem in terms of the percolation diameter d_c can be treated like the conventional problem in terms of the number or area fraction p_c. [An alternative approach, well suited to the continuum problem, is to use the number of obstacles as the fundamental variable (9).] The analysis largely follows the procedures of Rintoul and Torquato (18), an admirably clear description of the analysis of the conventional percolation problem for overlapping 3D spheres.

For a description of highly accurate techniques for the conventional problem, see Becker and Ziff (16), who obtained the thresholds for bond and site percolation on the Voronoi network and Delaunay triangulation to six significant figures. Their techniques require very accurate data, impractical to obtain here because a series of obstacle number densities is needed and the obstacles must be equilibrated.

First, we find the scaling exponent ν for the correlation length. A standard result of percolation theory is that the width of the CDFs varies as $L^{1/\nu}$, where L is the system size. The log-log plot for the soft-disk data is shown in Fig. S1*a*. The mean slope for the seven different concentrations is 0.766 ± 0.032, so we assume that ν has its usual value of 4/3 in 2D (19, p. 52). Similar plots for the other interactions gave slopes in the range 0.67 to 0.90. Pooling all the slopes gave a mean slope of 0.764 ± 0.066. For random overlapping obstacles the slopes were all below 0.75 but more extensive calculations would be needed in order to exclude the standard scaling exponent.

Fig. S1*b* compares the corresponding plots for overlapping, hard, and soft disks at a number density of $24\text{k}/\mu\text{m}^2$. The systems of soft and hard disks have the same number densities and the free area available to the obstacles is similar, so it is not surprising that the CDFs of the percolation thresholds have similar widths. In constrast, the free area available in a system of overlapping disks is much greater, so the fluctuations in percolation thresholds and the widths of the CDFs are greater.

Next, we use this exponent to extrapolate the Monte Carlo percolation thresholds to an infinite system. The standard scaling variable from the percolation literature is $x_{sc} = (p - p_c)L^{1/\nu}$, where p is the observed percolation threshold for a finite system of size L, and p_c is the percolation threshold for an infinite system. (We shall see the rationale for this scaling variable in Fig. S3.) For large systems, if the CDFs are plotted as a function of the scaling variable, they collapse to the same curve. In the standard scaling problem, p_c is obtained by plotting p versus $1/L^{1/\nu}$ and extrapolating the best-fit line to infinite L. Here we use the same procedure but in terms of the diameter, as shown in Fig. S2.

Fig. S2 also shows the consistency tests used. The black lines show the fits to different numbers of points, used in the error estimates. In addition, for each obstacle configuration one could use

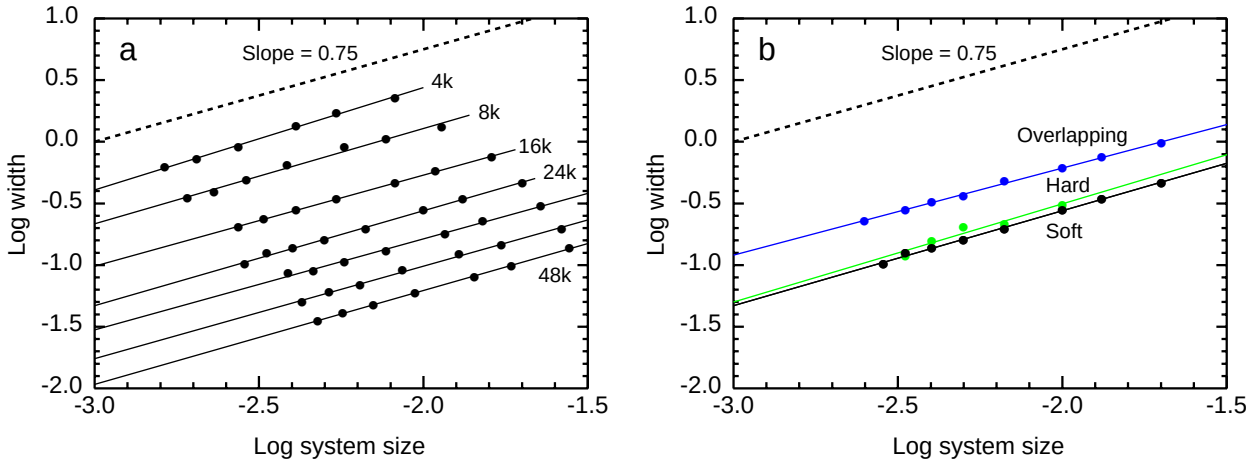

Figure S1: Log-log plot of the width of the CDFs versus system size L, the physical size in nm. *(a)* Values for soft disks at number densities 4k, 8k, 16k, 24k, 32k, 40k, and 48k/μm^2. *(b)* Values for soft disks, hard disks, and random overlapping disks at a number density of 24k/μm^2. *(Solid lines)* Least-squares fits to the various data sets. *(Dashed line)* Theoretical slope.

the threshold for percolation in either direction or the threshold for percolation in both directions. We use the two thresholds independently, but as shown in Fig. S2, the other choices extrapolate to the same threshold within error bars.

Finally we examine data collapse. Consider the family of CDFs in Fig. 3 of the text. They differ in position and steepness, so the intuitively obvious way to compare the curves is to adjust them to the same center by subtracting the observed mean, and adjust them to the same width by dividing by the observed width, defined as the standard deviation of the distribution. As Fig. S3a shows, if we plot the CDFs versus $x(expt) = (d - mean)/width$, all the different CDFs reduce to approximately the same curve for all the different model interactions, number densities, and system sizes, so in that sense they are the same curves. In particular, all the CDFs of Fig. 3 reduce to the same curve as in Fig. S3a (data not shown).

The scatter in Fig. S3a appears to be predominantly the result of noise in the CDFs, not differences among interactions, system sizes, or number densities. Fig. S3b shows the scaling plot for 6 runs under identical conditions, differing only in the random number sequence in the Brownian dynamics. The scatter in Fig. S3b for these runs is similar to that in Fig. S3a for 39 runs for 39 different sets of conditions. More quantitatively, if we measure the overall scatter by the scatter in $x(expt)$ at the 50% points of the CDFs, Figs. S3a and S3b are very similar in the range and standard deviation of $x(expt)$.

This empirical scaling is closely related to the standard scaling analysis. For large systems the mean is just the percolation threshold d_c, and as shown in Fig. S1, $width \propto 1/L^{1/\nu}$. So $x(expt) \propto (d - d_c)L^{1/\nu}$, which is the standard scaling variable x_{sc} but in terms of d instead of p. This scaling plot is shown in Figs. S3c,d. There is clear data collapse, but not to the same extent

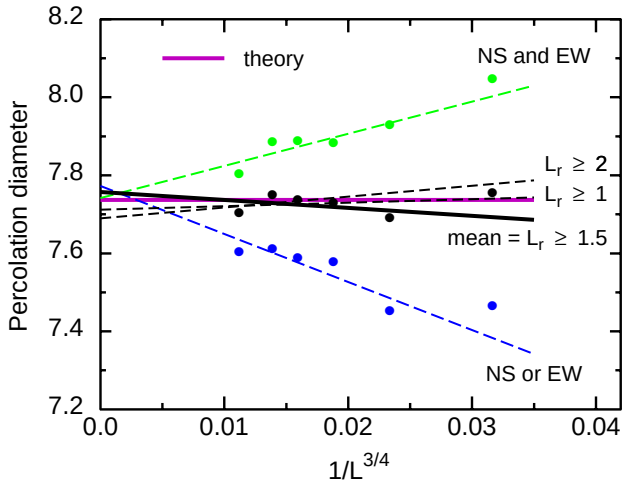

Figure S2: Extrapolation of the observed thresholds to an infinite system for random overlapping disks at a number density of $24\text{k}/\mu\text{m}^2$. *(Purple horizontal line)* The theoretical value from Eq. 3 of the text. In the standard calculation, the mean values *(black points)* are used in the least-squares fit. *(Solid black line)* The standard fit is to the set of points with $L_r \geq 1.5$. *(Dashed black lines)* Fits to the sets of points with $L_r \geq 1$ and $L_r \geq 2$ are used to estimate the error. Extrapolations based on percolation in one direction (or) and percolation in both directions (and) are shown, and give the same threshold within error bars.

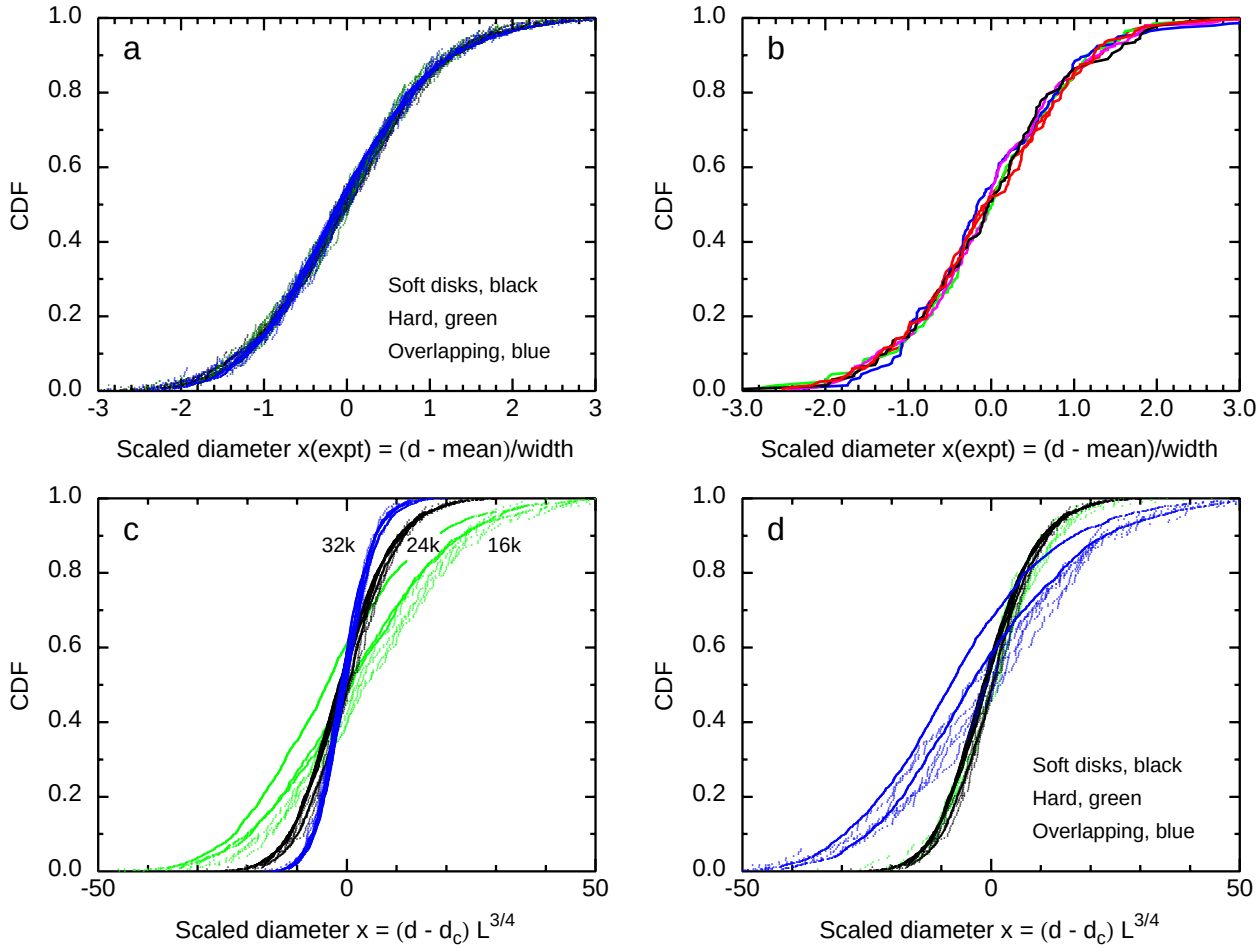

Figure S3: Data collapse for CDFs. *(a)* Empirical scaling plot of CDFs versus the scaling variable $x(expt) = (d - mean)/width$, where *mean* is the observed value for each run, and *width* is the observed standard deviation of the CDF for that run. The plot shows approximately 20,000 data points, for soft disks *(black)*, hard disks *(green)*, and random overlapping points *(blue)*. For each model interaction, a series of 5 to 7 system sizes is included at a number density of 24k/μm^2, and a series of 7 number densities is included at an approximately constant number of particles ($L_r = 2$ series). The plot was done in PostScript, where the points are opaque, so green overlaps black and blue overlaps both. *(b)* A similar plot for 6 independent runs for soft repulsive disks at 24k/μm^2, system size 150 nm. *(c)* Theoretical scaling plot of CDFs versus the scaling variable $x_{sc} = (d - d_c)L^{3/4}$ for soft disks at number densities of 16k, 24k, and 32k/μm^2. System sizes were between 44 and 366 nm. *(d)* A similar plot for soft, hard, and random overlapping disks at 24k/μm^2. System sizes were between 50 and 350 nm.

as in Fig. S3*a*. For each model interaction and number density, the CDFs for various system sizes reduce to a common curve, but there are distinct curves for each interaction and number density. The reason is that each interaction and number density has a different dependence of width on system size, even though the exponents are approximately the same, as shown in Fig. S1.

References

1. Saxton, M. J., 2007. Modeling 2D and 3D diffusion. *In* A. M. Dopico, editor, Methods in Membrane Lipids, Humana Press, Totawa, NJ, volume 400 of *Methods in Molecular Biology*, 295–321.

2. Allen, M. P., and D. J. Tildesley, 1989. Computer Simulation of Liquids. Clarendon Press, Oxford.

3. Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992. Numerical Recipes in FORTRAN: The Art of Scientific Computing. Cambridge University Press, Cambridge, 2nd edition.

4. Wolfram, S., 2003. The Mathematica Book. Wolfram Media, 5th edition.

5. Barber, C. B., D. P. Dobkin, and H. Huhdanpaa, 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Software* 22:469 – 483.

6. www.qhull.org, 2010.

7. xlr8r.info/mPower/index.html, 2010.

8. Hovi, J.-P., and A. Aharony, 1996. Scaling and universality in the spanning probability for percolation. *Phys. Rev. E* 53:235 – 253.

9. Rintoul, M. D., 2000. Precise determination of the void percolation threshold for two distributions of overlapping spheres. *Phys. Rev. E* 62:68 – 72.

10. Hoshen, J., and R. Kopelman, 1976. Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. *Phys. Rev. B* 14:3438 – 3445.

11. Hellriegel, C., J. Kirstein, C. Bräuchle, V. Latour, T. Pigot, R. Olivier, S. Lacombe, R. Brown, V. Guieu, C. Payrastre, A. Izquierdo, and P. Mocho, 2004. Diffusion of single streptocyanine molecules in the nanoporous network of sol-gel glasses. *J. Phys. Chem. B* 108:14699 – 14709.

12. Newman, M. E. J., 2005. Power laws, Pareto distributions and Zipf's law. *Contemp. Phys.* 46:323 – 351.

13. Okamoto, K., and M. Terazima, 2008. Distribution analysis for single molecule FRET measurement. *J. Phys. Chem. B* 112:7308 – 7314.

14. Berg, B. A., and R. C. Harris, 2008. From data to probability densities without histograms. *Comput. Phys. Commun.* 179:443 – 448.

15. Saxton, M. J., 1993. Lateral diffusion in an archipelago. Dependence on tracer size. *Biophys. J.* 64:1053 – 1062.

16. Becker, A. M., and R. M. Ziff, 2009. Percolation thresholds on two-dimensional Voronoi networks and Delaunay triangulations. *Phys. Rev. E* 80:041101.

17. Saxton, M. J., 1995. Single-particle tracking: effects of corrals. *Biophys. J.* 69:389 – 398.

18. Rintoul, M. D., and S. Torquato, 1997. Precise determination of the critical threshold and exponents in a three-dimensional continuum percolation model. *J. Phys. A* 30:L585 – L592.

19. Stauffer, D., and A. Aharony, 1992. Introduction to Percolation Theory. Taylor & Francis, London, 2nd edition.