

Supplementary Information

Salt-induced cluster formation of gold nanoparticles followed by stopped-flow SAXS, DLS and extinction spectroscopy

Nico Carl,^{a,b,c} Sylvain Prévost,^{b,d} Joseph P.S. Fitzgerald^e and Matthias Karg^{*a,e}

^a Physical Chemistry I, University of Bayreuth, 95447 Bayreuth, Germany.

^b Present address: Institut Laue-Langevin, 38043 Grenoble Cedex 9, France.

^c Present address: Physical Chemistry, University of Paderborn, 33098 Paderborn, Germany.

^d ESRF – The European Synchrotron, CS40220, 38043 Grenoble Cedex 9, France.

^e Present address: Physical Chemistry I, Heinrich-Heine-University Düsseldorf, 40225 Düsseldorf, Germany.

Tel: +49 211 81 12400; Fax: +49 211 81 12179; E-mail: karg@hhu.de

1 Experimental setup

Figure S1 shows the experimental setup in the hutch of the beamline ID02. The observation head of the stopped-flow device is equipped with a cuvette. Perpendicular to the X-Ray beam a light source and a spectrophotometer are used to acquire UV-Vis absorbance spectra.

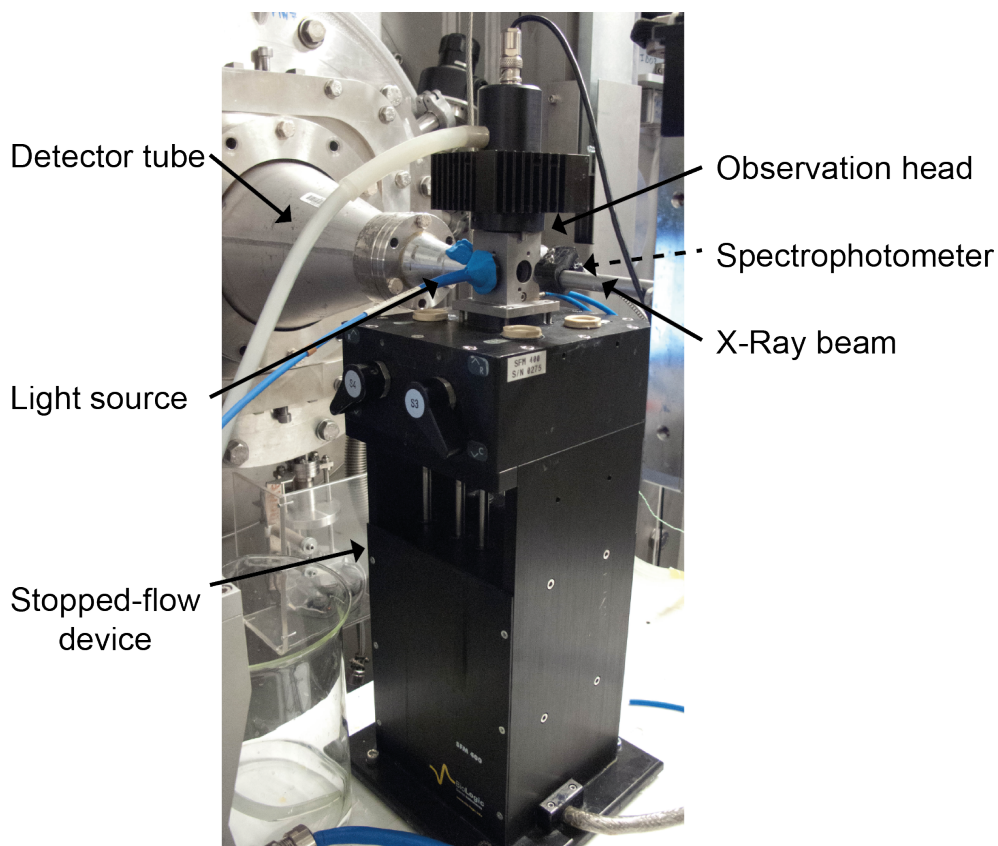


Fig. S1 Experimental setup in the experimental hutch of the beamline ID02.

2 Scattering length densities and length of the ligand

The scattering length density of the MUA shell can be estimated based on the volume v of a hydrocarbon chain which is given according to:¹

$$v[\text{\AA}^3] = 27.4 + 26.9 \cdot n_c \quad (1)$$

with n_c being the number of CH₂ groups.

The resulting scattering length density ρ of the shell as well as the one of gold and water are listed in table 1.

Table 1 Mass densities and X-Ray scattering length densities ρ at 11.5 keV of H₂O, Au and the MUA shell.

Material	Mass densities [g cm ⁻³]	ρ [cm ⁻²]	$\Delta\rho$ [cm ⁻²]
H ₂ O	1.0	9.44×10^{10}	—
Au	19.3	1.15×10^{12}	1.06×10^{12}
MUA	1.03	9.63×10^{10}	1.92×10^9

The length of the ligand l was calculated according to:¹

$$l[\text{\AA}] = 1.5 + 1.265 \cdot n_c \quad (2)$$

with n_c being the number of CH₂ groups. The length of the fully stretched ligand is roughly 1.5 nm.

3 Analysis of the low q regime of the SAXS profiles of the salt free particle dispersion

Figure S2A shows the SAXS profile of the salt free AuNP dispersion. In addition, a fit to the model of polydisperse homogeneous spheres and a sum of polydisperse spheres and a dimer of two spheres in close contact is shown. The model taking into account the dimers describes the scattering profile slightly better at low q . The number fraction of gold nanoparticles present in dimers is found to be 7%. The relatively high residuals in Figure S2B at $0.3 - 1 \text{ nm}^{-1}$ occur due to merging of scattering profiles from two different detector positions.

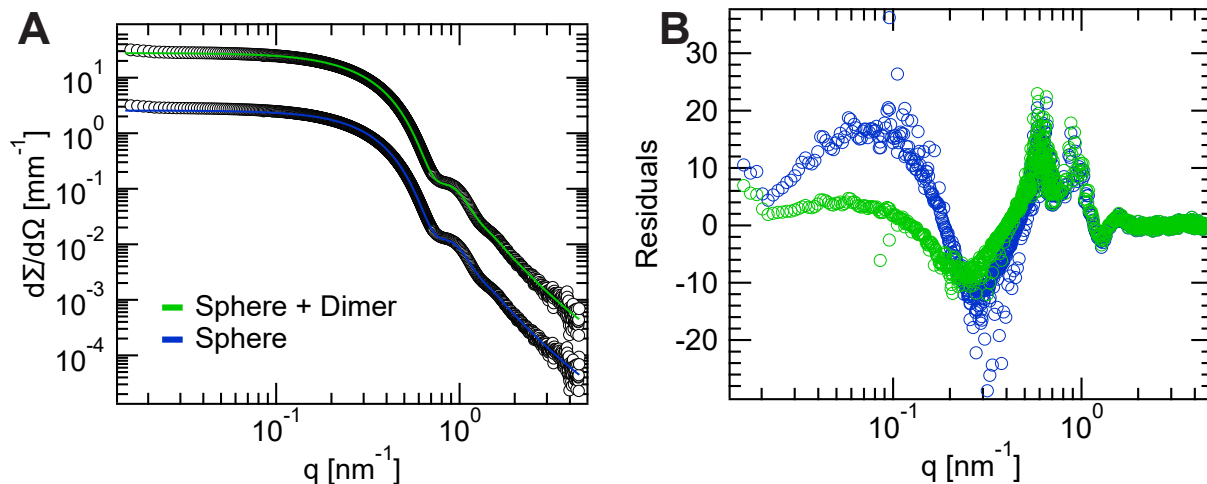


Fig. S2 SAXS profile of a dilute AuNP dispersion in H₂O. The error bars are smaller than the symbol size and are not included for clarity. The solid blue line represents a fit by the model of polydisperse homogeneous spheres, whereas the green solid line is a fit using the sum of polydisperse homogeneous spheres and a dimer of two spheres in close contact. The fit for the sum of dimers and single spheres is multiplied by a factor of 10 (A). Residuals of the shown fits (B).

4 Structure factor determination

The macroscopic scattering cross-section is given by:

$$\frac{d\Sigma}{d\Omega} = N \cdot V^2 \cdot \Delta\rho^2 \cdot P(q) \cdot S(q) \quad (3)$$

with N being the particle number density, V the volume of the scattering object, $\Delta\rho$ the scattering length density between particle and solvent, $P(q)$ the form factor of the particle and $S(q)$ the structure factor.

By measuring a dilute dispersion of AuNPs the experimental form factor $P(q)$ of the particles is available ($S(q) \rightarrow 1$). In contrast for dispersions of AuNP clusters, the structure factor affects the scattering profile and needs to be considered. The experimentally obtained

form factor $P(q)$ from measurements of a dilute sample (solid lines) and the scattering profile of the clusters (symbols) are shown in direct comparison in Fig. S3. At high q the form factor and the scattering profiles of the clusters overlap nicely, whereas a significant difference is visible at low q . This difference is related to the structure factor of the clusters. By scaling the high q region of $P(q)$ to the measured scattering profiles an effective structure factor $S(q)$ can be determined:

$$S(q) = \frac{\frac{d\Sigma}{d\Omega}}{N \cdot V^2 \Delta\rho^2 \cdot P(q)} \quad (4)$$

The resulting effective structure factor is shown in the main manuscript in Figure 4D.

The position q_{\max} of the first maximum of $S(q)$ is related to the most probable inter-particle distance d_{c-c} by:

$$d_{c-c} = \frac{2\pi}{q_{\max}} \quad (5)$$

The structure factors $S(q)$ in Figure 4D show an upturn at low q as well as two oscillations at high q . We empirically explain these features by a superposition of a hard-sphere form factor as well as the form factor of a fractal object. Figure S4A shows the theoretical hard sphere and fractal structure factors. Figure S4B shows a superposition of both structure factors, showing the same trend as the experimental structure factors.

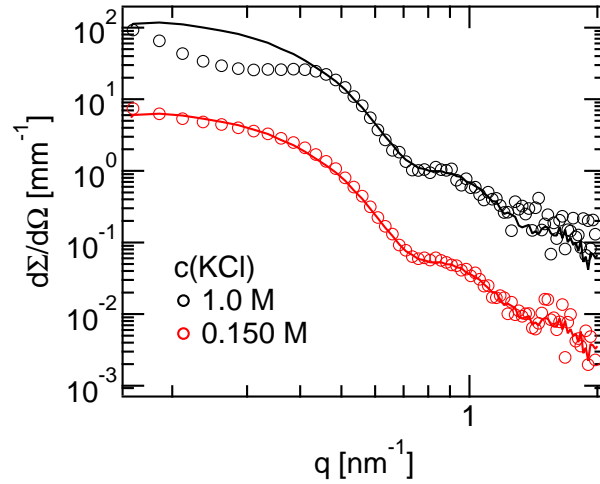


Fig. S3 Scattering profiles of AuNP clusters at low q for $c(\text{KCl}) = 0.150$ and 1.0 M. The drawn through lines represent the measurement of the dilute, non-aggregated particles scaled to the scattering profiles of the clusters. The profile at a salt concentration of 1.0 M is multiplied by a factor of 20 and the error bars are neglected for clearer representation.

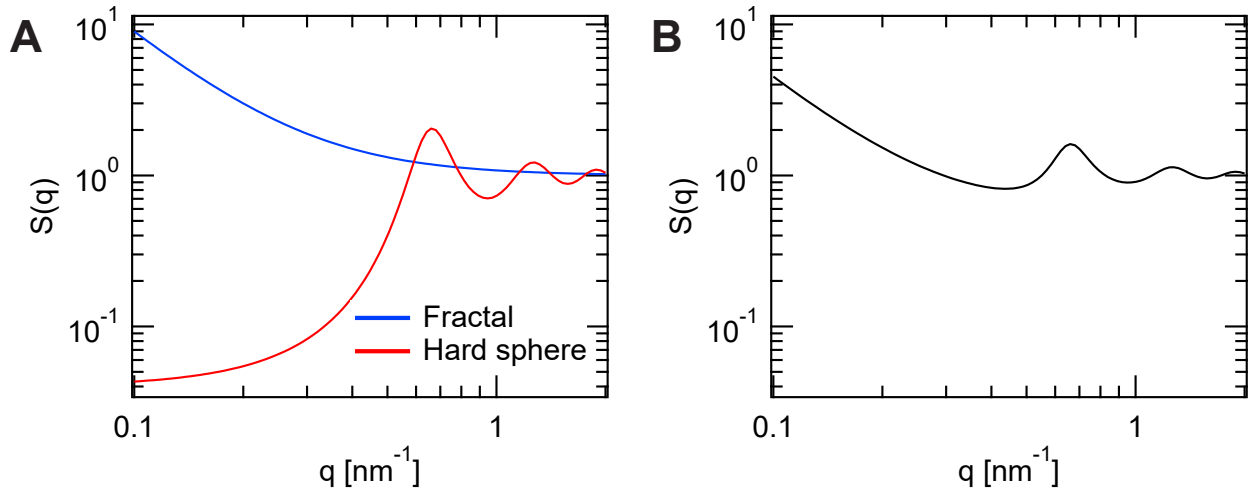


Fig. S4 Theoretical structure factor $S(q)$ for hard spheres (hard sphere repulsion radius of 5.0 nm and volume fraction of 0.4) and mass fractals (characteristic size of the individual scatters 5.0 nm, size of the fractal aggregate 200 nm and a fractal dimension of 2.0) (A). Superposition of the structure factors shown in (A) (B).

5 3D plot of TR-SAXS data

Figure S5 shows an exemplary 3D plot of one SAXS dataset acquired at a KCl concentration of 0.325 mol L^{-1} .

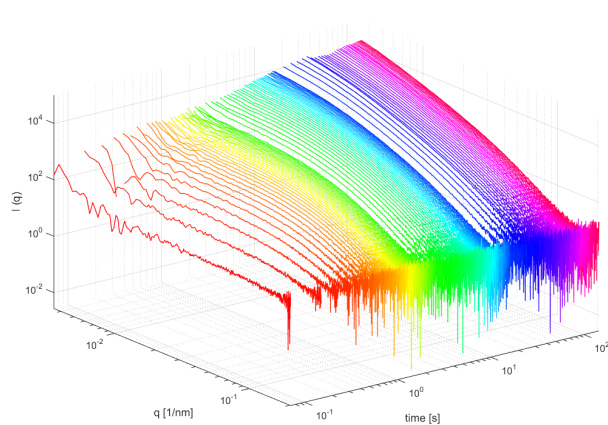


Fig. S5 3D plot of one SAXS dataset at 0.325 mol L^{-1} KCl.

6 Random walk simulation

The clusters were built using a self-avoiding random walk using a Python script. The first particle was set to the cartesian coordinate (0,0,0). For a given inter-particle spacing d_{spacing} the center-to-center distance d_{c-c} for two particles was calculated:

$$d_{c-c} = 2 \cdot R + d_{\text{spacing}} \quad (6)$$

With R being the particle radius.

Consequently, a random number between $[0, \pi]$ and $[0, 2\pi]$ was generated for the two angles ϕ and ϑ . R , ϕ and ϑ define the position of the second particle in the polar coordinate system. The cartesian coordinates (x, y, z) for the new particle are then given by:

$$x = d \cdot \cos(\phi) \cdot \sin(\vartheta) \quad (7)$$

$$y = d \cdot \sin(\phi) \cdot \sin(\vartheta) \quad (8)$$

$$z = d \cdot \sin(\vartheta) \quad (9)$$

This is consequently repeated N times, with N being the total particle number in the cluster.

For self-avoidance the center-to-center distance between a new particle and all old ones must be $\geq d_{c-c}$.

Listing 1 shows the used Python code for the self-avoiding random walk simulation.

```

1 import numpy as np
2 import numpy.ma as ma
3 import random
4 import math
5
6 rvec = np.zeros([1,3])
7 while((np.size(rvec)/3) < Nparticles):
8     rvec_new = np.zeros([3,])
9     theta = math.radians(random.randrange(0,180))
10    sigma = math.radians(random.randrange(0,360))
11    rvec_new[0] = np.round(rvec[-1][0] + dist*math.sin(theta)*math.cos(sigma),2)
12    rvec_new[1] = np.round(rvec[-1][1] + dist*math.sin(theta)*math.sin(sigma),2)
13    rvec_new[2] = np.round(rvec[-1][2] + dist*math.cos(theta),2)
14    rvec_dist = np.zeros([int(np.size(rvec)/3)])
15    for i in range(int(np.size(rvec_dist))):
16        rvec_dist[i] = np.linalg.norm(rvec[i]-rvec_new)
17
18
19    while(np.amin(rvec_dist) < dist):
20
21        theta = math.radians(random.randrange(0,180))
22        sigma = math.radians(random.randrange(0,360))
23        rvec_new[0] = np.round(rvec[-1][0] + dist*math.sin(theta)*math.cos(sigma),2)
24        rvec_new[1] = np.round(rvec[-1][1] + dist*math.sin(theta)*math.sin(sigma),2)
25        rvec_new[2] = np.round(rvec[-1][2] + dist*math.cos(theta),2)
26
27        rvec_dist = np.zeros([int(np.size(rvec)/3)])
28        for i in range(int(np.size(rvec_dist))):
29            rvec_dist[i] = np.linalg.norm(rvec[i]-rvec_new)
30    rvec = np.vstack((rvec, rvec_new))

```

Listing 1 Python code used for the self-avoiding random walk.

References

- 1 C. Tanford, *The Journal of Physical Chemistry*, 1972, **76**, 3020–3024.