Supporting information for

Bayesian estimation to deconvolute single-particle ICP-MS data with a mixed Poisson distribution

Yoshinari Suzuki, *[a] Midori Kondo [a], Masae Harimoto [a], Yusuke Okamoto [a], Yu-ki Tanaka [b], Yasumitsu Ogra [b] and Hiroshi Akiyama [a,c]

[a] Division of Foods, National Institute of Health Sciences, 3-25-26 Tonomachi, Kawasaki-ku, Kawasaki-shi, Kanagawa 210-9501, Japan.

[b] Graduate School of Pharmaceutical Sciences, Chiba University, 1-8-1 Inohana, Chuo-ku, Chiba-shi, Chiba 260-8675, Japan.

[c] Department of Analytical Chemistry, Hoshi University, 2-4-41 Ebara, Shinagawa-ku, Tokyo 142-8501, Japan.

* Corresponding author:

Yoshinari Suzuki, e-mail address: szk-yoshi@nihs.go.jp

S1 Analytical examples with R codes

Many algorithms for spICP-MS analysis have been reported so far. However, most of their details are in a black box and cannot be easily reproduced by anyone other than those involved. In this supporting information, we show how the Bayesian estimation processes presented in the main text can be simulated numerically and give an example for Ag-NPs with 60-nm diameters as separate zip files ('mixed Poisson signal split model sp-ICP-NS (alpha=0.01, NP=100, blk=1).zip' and '200803 20 AgNP 60nm 10ppt-1 without data.zip'). The codes included in this paper is available via GitHub (https://github.com/Yoshinari-Suzuki/Bayesian-spICP-MS-analysis.git).

As noted in the main text, analytical processes were performed using the statistical software R. It is free software and runs on a wide variety of UNIX platforms, Windows, and MacOS. Thus, using the code published here, anyone can reproduce the same analytical result, simulate with different parameter(s), and import and analyse their own measurement data. This ability is a big advantage compared with software that must be purchased. The unzipped folder corresponds to the project folder of Rstudio. Rstudio is an integrated development environment for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging, and workspace management. The analytical code is described in the Rmd file, and the result of a series of analyses is output as an html file. Open the Rroj file, 'mixed Poisson.Rporj', and then click '*.Rmd' in the 'Files' tab (the Files tab is located in the lower right panel by default). The Rmd file will be opened in the upper-left panel. All analytical data were saved as '.Rdata' in the 'Environment' tab (the Environment tab is located in the upper-right panel by default), ('.Rdata' is available for only 'mixed Poisson signal split model sp-ICP-NS (alpha=0.01, NP=100, blk=1).zip' because of the large file size). To load our analytical results onto R environment, click '.Rdata' in the 'Environment' tab.

To install rstan or cmdstanr, follow the official instructions (https://mc-stan.org/users/interfaces/rstan for rstan and https://mc-stan.org/cmdstanr/cmdstanr for cmdstanr). Install the other R packages if necessary. It seems safer to execute Stan after confirming that it works with a simpler model. When the environment is ready, please execute the code in the Rmd file. Under our environment [Processor: Intel® Core(TM) i7-8700 CPU @ 3.2GHz; RAM: 16.GB], it took about 5 min to estimate the Bayesian model for the simulated data, and it took about 1 h to estimate for Ag-NPs with particle sizes of 60 nm. It may be useful to check the flow using a smaller amount of data.

```
data {
  int N;
  int<lower=0> Y[N];
  real<lower=0> mu_blk;
  real<lower=0> sd_blk;
  real<lower=0> alpha_int;
  real<lower=0> lambda_NP_int;
  real<lower=0> lambda_NP_max;
}

transformed data {
  real max_Y;
  int C_max;

  max_Y = max(Y) - mu_blk;
  C_max = max(Y);
}

parameters {
  real<lower=0, upper=1> alpha;
  real<lower=0> lambda_blk;
  real<lower=0, upper=lambda_NP_max > lambda_NP;
  real<lower=0, upper=1> delta_r;
}

transformed parameters {
  real<lower=0.5, upper=1> delta;

  delta = 0.25*delta_r*(3-delta_r^2) + 0.5;
}

model {
  delta_r ~ beta(1, 1);
  alpha ~ normal(alpha_int, alpha_int/2);
  lambda_blk ~ normal(mu_blk, sd_blk);
  lambda_NP ~ normal(lambda_NP_int, lambda_NP_int/2);

  for (i in 2:N-1) {
    vector lp[9];
    lp[1] = 3*log1m(theta) + poisson_lpmf( Y[i] | lambda_blk);
    lp[2] = 2*log1m(theta) + log(theta) + poisson_lpmf( Y[i] | lambda_blk + (1-ssf)*lambda_NP );
    lp[3] = 2*log1m(theta) + log(theta) + poisson_lpmf( Y[i] | lambda_blk );
    lp[4] = log(0.5) + log1m(theta) + 2*log(theta) + poisson_lpmf( Y[i] | lambda_blk + (1-ssf)*lambda_NP );
    lp[5] = log(0.25) + log1m(theta) + 2*log(theta) + poisson_lpmf( Y[i] | lambda_blk + 2*(1-ssf)*lambda_NP );
    lp[6] = log(0.25) + log1m(theta) + 2*log(theta) + poisson_lpmf( Y[i] | lambda_blk );
    lp[7] = 2*log1m(theta) + log(theta) + poisson_lpmf( Y[i] | lambda_blk + ssf*lambda_NP );
    lp[8] = log1m(theta) + 2*log(theta) + poisson_lpmf( Y[i] | lambda_blk + ssf*lambda_NP );
    lp[9] = log1m(theta) + 2*log(theta) + poisson_lpmf( Y[i] | lambda_blk + lambda_NP );  //

    target += log_sum_exp(lp);
  }
}

generated quantities {
  real pmf_blk[C_max + 1];
  real pmf_NP[C_max + 1];
  real pos_NP[C_max + 1];

  for (i in 1:C_max+1) {
    vector[6] lp_blk;
    vector[3] lp_np;
    lp_blk[1] = 3*log1m(theta) + poisson_lpmf( i-1 | lambda_blk);
    lp_blk[2] = 2*log1m(theta) + log(theta) + poisson_lpmf( i-1 | lambda_blk + (1-ssf)*lambda_NP);
    lp_blk[3] = 2*log1m(theta) + log(theta) + poisson_lpmf( i-1 | lambda_blk);
    lp_blk[4] = log(0.5) + log1m(theta) + 2*log(theta) + poisson_lpmf( i-1 | lambda_blk + (1-ssf)*lambda_NP);
    lp_blk[5] = log(0.25) + log1m(theta) + 2*log(theta) + poisson_lpmf( i-1 | lambda_blk + 2*(1-ssf)*lambda_NP);
    lp_blk[6] = log(0.25) + log1m(theta) + 2*log(theta) + poisson_lpmf( i-1 | lambda_blk );
    lp_np[1] = 2*log1m(theta) + log(theta) + poisson_lpmf( i-1 | lambda_blk + ssf * lambda_NP);
    lp_np[2] = log1m(theta) + 2*log(theta) + poisson_lpmf( i-1 | lambda_blk + lambda_NP);
    lp_np[3] = log1m(theta) + 2*log(theta) + poisson_lpmf( i-1 | lambda_blk + ssf*lambda_NP);

    pmf_blk[i] = exp( log_sum_exp(lp_blk) );
    pmf_NP[i]  = exp( log_sum_exp(lp_np) );
    if (pmf_blk[i] + pmf_NP[i] == 0) {
      pos_NP[i] = 0;
    } else {
      pos_NP[i] = pmf_NP[i] / (pmf_blk[i] + pmf_NP[i]);
    }
  }
}
```

Fig. S1 Stan code for estimating parameters from sp-ICPMS data assuming a mixed Poisson distribution

S2 Description of the Stan Program

The Stan code, which assumes a mixed Poisson distribution, is shown in Fig. S1. This code contains 6 blocks (data, transformed data, parameters, transformed parameters, model, and generated quantities). Simple explanations for the code are described after the "//" notation in some rows.

In the data block (Fig. S1, lines 1–9), we specified two data dimensions (sample number of observed data [N] and value of observed data [Y]) and four values (mu_blk, sd_blk, alpha_int, lambda_NP_int, and lambda_NP_max) used in the prior distribution. "mu_blk" and "sd_blk" are the mean and standard deviation of the signals of the blank solution, respectively, and "alpha_int" and "lambda_NP_int" are the expected mean of alpha and lambda_NP, respectively. Basically, we used an "alpha_int" of 0.00067 and "lambda_NP_int" of 200 for 25 ng mL^{-1} of Ag-NPs with 60-nm diameter. In the transformed data block (Fig. S1, lines 11–17), we calculated the maximum value of Y as a real, integer number. "lambda_NP_max" was used for upper limit for "lambda_NP".

In the parameters block (Fig. S1, lines 19–24), we declared four parameters [alpha (α), lambda_bkg (λ_{bkg}), lambda_NP (λ_{NP}), and delta_r (δ_r)]. Upper and/or lower limit(s) were set in some cases. 'alpha' is the frequency parameter of the particle event. 'lambda_bkg' and 'lambda_NP' are the expected signal of the background and particle-event intensity, respectively. Let the radius of the sphere be 1, and consider it in the coordinate space, as shown in Fig. 2S. When a sphere is cut by one plane, 'delta_r' (δ_R, $0 \le \delta_R \le 1$) corresponds to the shortest distance between the plane and the centre of the sphere.

In the transformed parameters block (Fig. S1, lines 26–30), we transformed the ratio of the spherical segment volume to spherical volume (δ, $0.5 \le \delta \le 1$) from δ_R. Let V be a spherical volume with a radius of 1 and V_1 be the spherical segment volume containing the centre of the sphere. Then δ can be described as follows:

$$\delta = \frac{V_1}{V} = \frac{1/6\pi(1+\delta_R)\{3(1-{\delta_R}^2)+(1+\delta_R)^2\}}{4/3\pi} = \frac{1}{4}\delta_R(3-{\delta_R}^2)+0.5. \ (S1)$$

In the model block (Fig. S1, lines 32–52), Stan can estimate the declared parameter(s). In the Bayesian framework, all parameters follow a probability distribution. The prior distribution(s) must therefore be specified. In Stan, when a prior distribution(s) is not specified in the model block, a uniform distribution [$f_{pri}(\theta) \sim Uniform(-\infty,\infty)$] is automatically applied. In lines 33–36 of Fig. S1, the informative prior distributions for all parameters are specified. The prior distributions for 'alpha' and 'lambda_NP' were specified as normal distributions with a mean value of the specified value in the data block and a standard deviation of the half-specified value in the data block. The prior distribution of 'delta_r' was specified as uniform distribution because there was no informative prior distribution. A Beta distribution [Beta(1,1)], which is equivalent to the uniform distribution with a

range of 0–1, is applied. From lines 38– 50 Fig. S1, the values of 'Y' are specified as stochastically generated from a mixed Poisson distribution with certain parameters that follow the patterns of Fig. 1. The log-probability for 9 patterns were calculated in the lines 40–48, then the probability of each pattern is summed to calculate the logarithmic probability of the entire model in the line 50. Stan seeks to find the optimum parameter values from the data. In the generated quantities block (Fig. S1, lines 54–80), 'pos_NP' [P(k)], 'pmf_NP' [$PMF_{NP}(k)$], and 'pmf_bkg' [$PMF_{bkg}(k)$] were calculated by using the estimated parameters.

In the numerical simulation study, the value of δ_R was first generated as a random number drawn from a uniform distribution with a range of 0–1 [$\delta_R \sim Uniform(0,1)$], and then δ was calculated according to Eqn. S1.

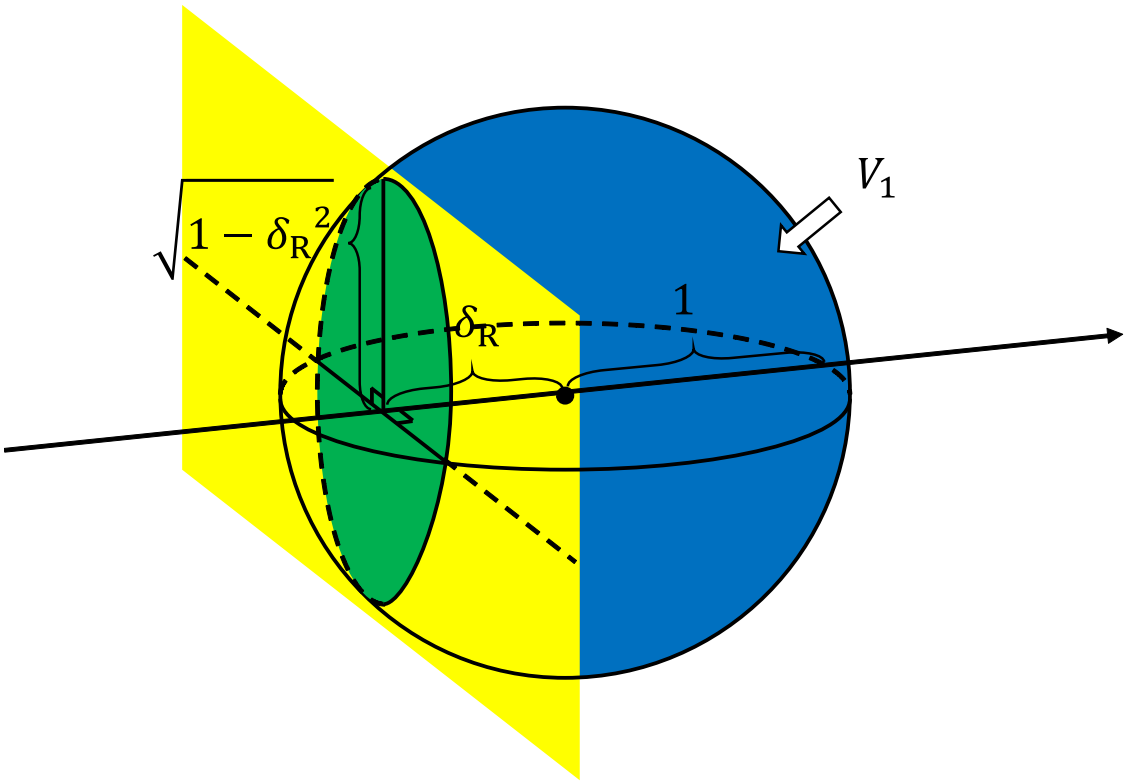


Fig. S2 Coordinate space associated with a sphere of radius 1 and a sphere segment. δ_R: the shortest distance between the plane and the centre of the sphere, V_1: the spherical segment volume containing the centre of the sphere.

Table S1 Prepared concentrations of nanoparticle standards

Nanoparticle	Diameter (nm)	Mass concentration	Particle number concentration (particles mL^{-1})
Ag	10	0.10 pg mL^{-1}	1.8×10^4
Ag	20	0.75 pg mL^{-1}	1.7×10^4
Ag	20	1.0 pg mL^{-1}	2.3×10^4
Ag	40	5.0 pg mL^{-1}	1.4×10^4
Ag	60	25 pg mL^{-1}	2.1×10^4
SiO_2	500	0.50 ng mL^{-1}	6.5×10^3 *
SiO_2	500	1.0 ng mL^{-1}	1.3×10^4 *
SiO_2	500	1.5 ng mL^{-1}	2.0×10^4 *
SiO_2	500	2.0 ng mL^{-1}	2.6×10^4 *
SiO_2	500	5.0 ng mL^{-1}	6.5×10^4 *
SiO_2	1000	15 ng mL^{-1}	2.5×10^4 *

* The particle number concentrations were calculated based on the nominal particle size of 492 or 976 nm.

Table S2 Typical operating conditions of ICP-MS

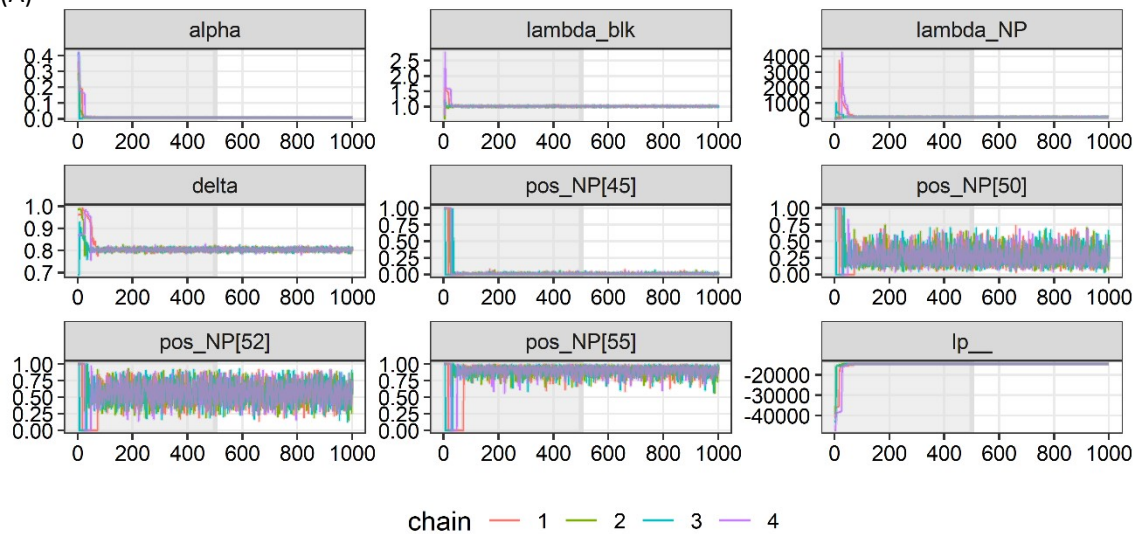
Plasma condition	
Rf power	1550 W
Plasma gas flow rate	13.8 L min^{-1}
Auxiliary gas flow rate	0.79 L min^{-1}
Nebulizer gas flow rate	1.1 L min^{-1}
Sampling depth	5.0 mm
Cell gas	no-gas for Ag, H_2 gas (5.5 mL min^{-1}) for Si
Data acquisition	
Data point per mass	1
Integration time	0.5 ms
Data acquisition time	60 s

S3 Convergence of Markov chain Monte Carlo

After the Markov chain Monte Carlo (MCMC) iteration had finished, we confirmed the convergence of the MCMC for all parameters. Fig. S3 illustrates the result for the simulated data as an example. The declared parameters (α, λ_{bkg}, and λ_{NP}), the transformed parameter (δ), and the sum of the log(arithmetic) posterior probabilities (lp__) had converged within 1000 iterations (Figs. S3A and B). Moreover, $\hat{R}$, which is the ratio of inter-chain variance to intra-chain variance, was 1.1 or less. This condition is a general criterion of convergence. In addition, the relative effective sample number (n_{eff}/N) and the relative Monte Carlo standard deviation (mcse/sd) also satisfied general criteria (0.1 or more and 0.1 or less, respectively) (Fig. S3C). Actually, one n_{eff}/N for pos_NP[107] was less than 0.1. Because the posterior distribution of pos_NP[107] equalled 1 for 2000 Monte Carlo samples, the calculations indicated that the posterior distribution of pos_NP[107] was autocorrelated. These results imply that the dissociation of the estimated values among the chains and the influence of

autocorrelation were small. Although each chain started from different initial values, they all finally arrived at a similar value. We conclude that all the calculated values converged.

(A)



Parameter	Mean	SD	$\hat{R}$	n_{eff}/N	$mcse/sd$
α	8.71×10^{-3}	0.67×10^{-3}	0.9997	0.885	0.024
λ_{bkg}	1.01	0.099	1.000	1.015	0.022
λ_{NP}	103	1.7	0.9994	0.845	0.024
δ	0.804	0.006	0.9993	0.807	0.025
lp__	-1.463×10^{3}	4.75×10^{-2}	0.9999	0.438	0.034

(B) Convergence indices for MCMC parameters (iteration=1000, warmup=500, chain=4, thinning=1)

(C)

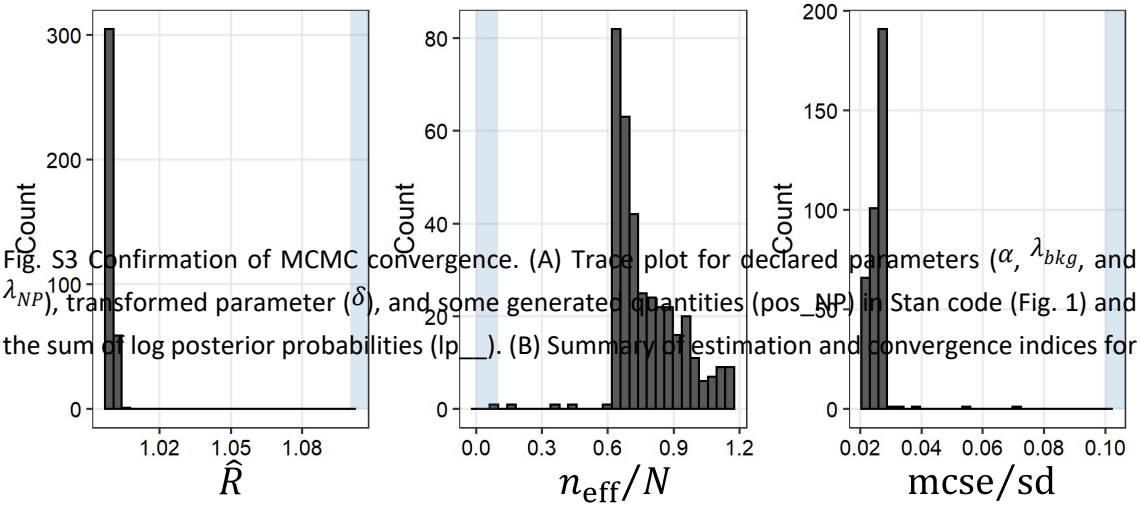


Fig. S3 Confirmation of MCMC convergence. (A) Trace plot for declared parameters (α, λ_{bkg}, and λ_{NP}), transformed parameter (δ), and some generated quantities (pos_NP) in Stan code (Fig. 1) and the sum of log posterior probabilities (lp__). (B) Summary of estimation and convergence indices for

parameters and lp__. (C) Histograms of three convergence indices ($\hat{R}$, n_{eff}/N, and $mcse/sd$) for all calculated values, where $\hat{R}$ is the ratio of intra-chain variation to inter-chain variance, n_{eff}/N is the ratio of effective sample number to MCMC sample, and $mcse/sd$ is the ratio of Monte Carlo standard error to standard deviation. If calculated data are not distributed in the light blue area, it can be concluded that MCMC calculations have converged.

S4 Point estimator of posterior distribution

When we want to estimate an unknown population parameter θ on the basis of observations Y, we can calculate the posterior distribution of θ [$f_{post}(\theta \mid Y)$] using Bayes' theorem:

$$f_{post}(\theta \mid Y) = \frac{L(Y \mid \theta) f_{pri}(\theta)}{f_{obs}(Y)}, (S2)$$

where $L(Y \mid \theta)$ is a likelihood function, $f_{pri}(\theta)$ is a prior distribution, and $f_{obs}(Y)$ is a distribution of observations Y. In the Bayesian framework, estimated results are obtained as distributions. The three-point estimators for posterior distributions are known: expected a posteriori (EAP), median of posterior distribution (MED), and maximum a posteriori (MAP).

Using the mean squared error (MSE) as risk, where MSE is defined by $E[(\hat{\theta} - \theta)^2]$, the point estimate of the certain unknown parameter is simply the EAP. If the "linear" loss function (LLF) is used as the risk, where LLF is defined by $a|\hat{\theta} - \theta|$ with $a > 0$, the point estimate of the certain unknown parameter is simply the MED. The MAP is closely related to the method of maximum likelihood estimation (MLE), but it employs an augmented optimization objective that incorporates a prior distribution. MAP estimation can therefore be seen as a regularization of MLE. The method of MAP estimates the mode of the posterior distribution of this random variable:

$$\hat{\theta}^{MAP} = arg\max_{\theta} f_{post}(\theta \mid Y). (S3)$$

Although the EAP of the skewed posterior distribution is different from MED and MAP, EAP is the most widely used and validated estimator.

S5 Confidence Interval and Credible Interval

A credible interval (CrI) is an important concept in Bayesian statistics to describe and summarize uncertainty. In this regard, CrI is quite similar to the frequentist "confidence Intervals (CI)". However, whereas their goals are similar, their statistical meanings are different.

● 95% CI: with a large number of repeated samples, 95% CI represents 95% frequency (i.e., 95% proportion) of possible confidence intervals that contain the true estimate of the unknown parameter.

● 95% CrI: given the observed data, there is a 95% probability that the true estimate of an unknown parameter would lie within the 95% CrI.

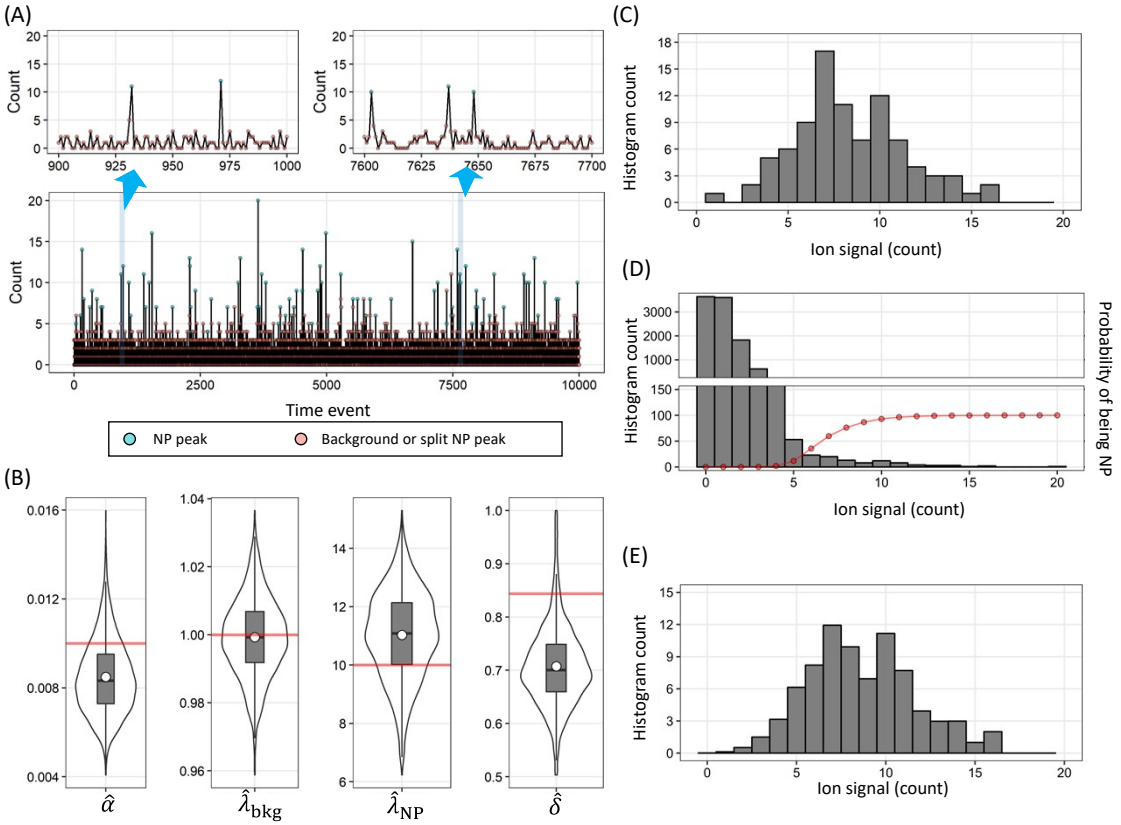


Fig. S4 Analytical example for simulation study under the condition of signal-to-background ratio of 10. (A) Simulated time-resolved data assuming spICP-MS. (B) Posterior distributions for $\hat{\alpha}$, $\hat{\lambda}_{bkg}$, $\hat{\lambda}_{NP}$, and $\hat{\delta}$. Horizontal red lines indicate the true values. (C) Histogram of particle-event height, which corresponds to the green points in panel A. (D) Histogram of ion signal for all readings and probability of being NP (red line). (E) Histogram of restored particle-event height. Symbols are explained in the list of symbols and throughout the text.

S6 Simulation study under the following condition: α**=0.01,** λ_{NP}**=10 and** λ_{bkg}**=1**

Figure S4(A) shows the simulated counts for the time-resolved analysis obtained by spICP-MS, and Fig. S4(B) shows the estimated ion signal obtained with the BE method for those data. The true values were distributed within the 95%CrI of the posterior distributions for all parameters.

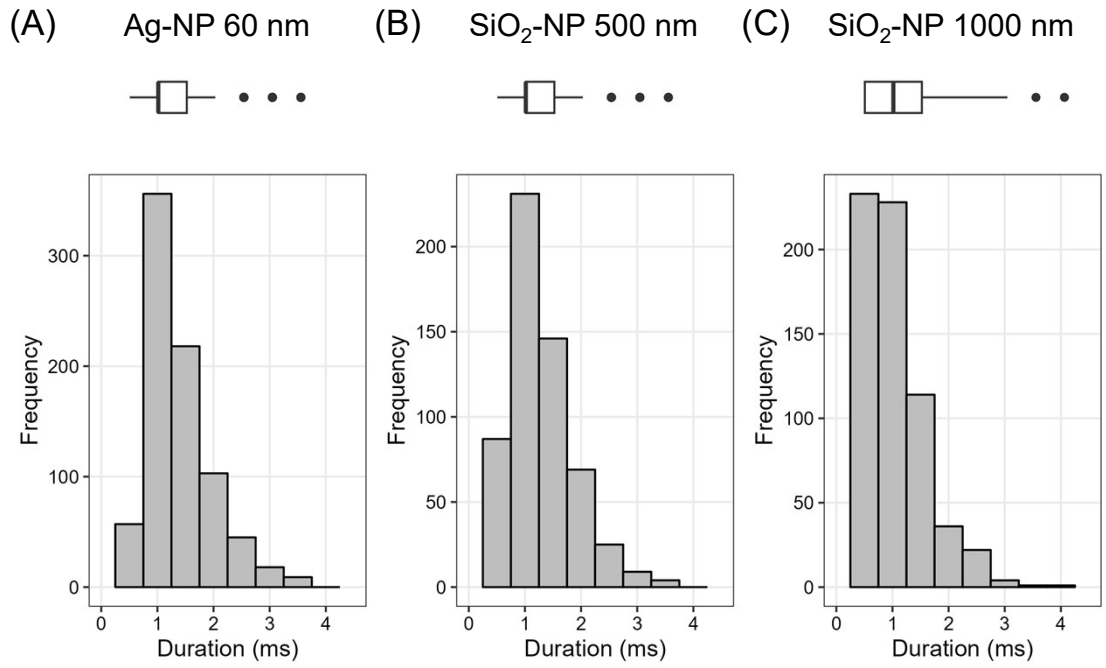


Fig. S5 Boxplots and histograms of particle-event duration

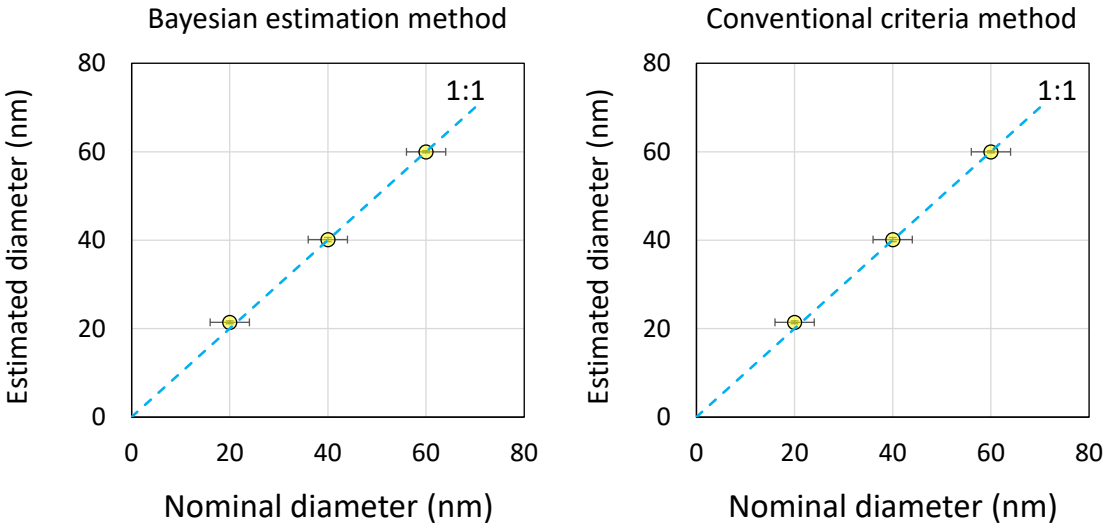


Fig. S6 Relationships between estimated diameter and nominal diameter of Ag-NP with diameters of 20–60 nm using Bayesian estimation (left) and conventional criteria method (right)

S7 Analytical example of 1000-nm SiO_2 nanoparticles

Figure S6 shows the analytical example for SiO_2-NPs with 1000-nm diameters with a DT of 0.5 ms. The both mean $\hat{\phi}_{NP}^{CC}$ (952 nm) and $\hat{\phi}_{NP}^{BE}$ (979 nm) show good agreements with the nominal diameter (976 ± 30 nm). Moreover, the both $\hat{C}_{PN}^{CC}$ (2.3 × 10^4 particles mL^{-1}) and $\hat{C}_{PN}^{BE}$ (2.5 × 10^4 particles mL^{-1}) distributed within the uncertainly of the prepared concentration [(2.3–2.8) × 10^4 particles mL^{-1})

derived from the uncertainty of the nominal diameter by the manufacture.

In the case of large particles with a high melting point, there is a concern that accurate particle mass calculation may not be possible due to incomplete vaporization in the plasma. In this study, reasonable particle size estimation results are obtained for SiO_2-NPs with 1000-nm diameter. This result indicates that the influence of incomplete vaporization can be ignored under current condition.

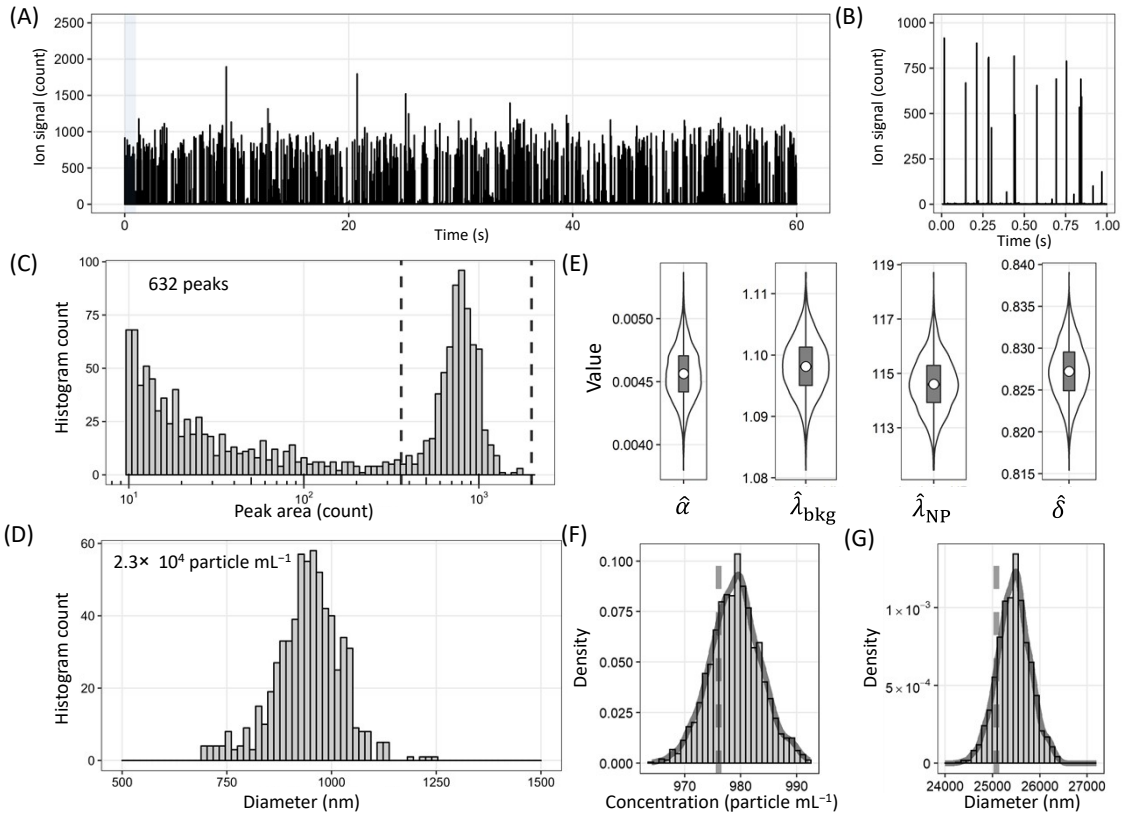


Fig. S7 Analytical example of spICP-MS for SiO_2-NP with 1000-nm diameter. (A) Time-resolved analysis. (B) Time-resolved analysis for shaded region in panel A. (C) Histogram of particle-event intensity using conventional criteria method. (D) Histogram of particle diameter using conventional criteria method. (E) Posterior distributions of parameters of mixed Poisson distribution. (F) Posterior distributions of particle number concentration and (G) particle diameter. Vertical dashed line indicates prepared (F) or nominal value (G). Symbols are explained in the list of symbols and throughout the text.

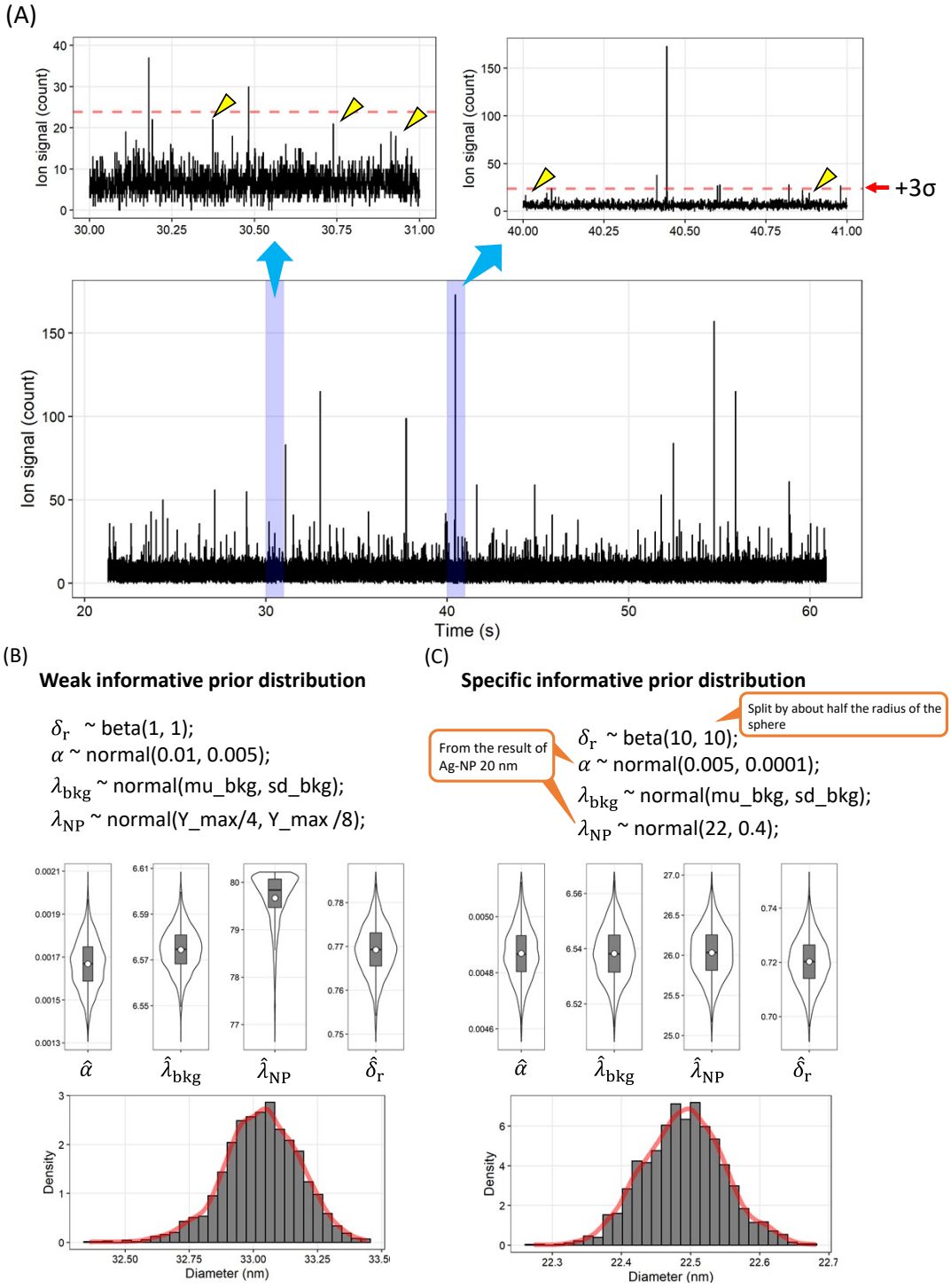


Fig. S8 Examples of Bayesian estimation for spICP-MS data of Ag-NPs with 20-nm diameters plus dissolved Ag ions at a concentration of 0.1 μg L^{-1} using different prior distributions. (A) Time resolved analysis data. Red dashed lines indicate criterion of NP signal defined as mean plus 3σ, and yellow triangles indicate transient signals, which appear to be derived from NP but are lower than the criterion value. (B) Bayesian estimation results using the weak informative prior distribution. (C) Bayesian estimation results using the specific informative prior distribution.

Table S3 The prior distributions used in the simulation study for various signal-to-background ratios[a,b]

Background parameter	α	λ_{blk}	λ_{NP}	δ_R
0.3	*N*(0.01, 0.01)	*N*(0.3, 0.5)	*N*(30,30)	Beta(300, 300)
1	*N*(0.01, 0.01)	*N*(1, 1)	*N*(30,30)	Beta(300, 300)
3	*N*(0.01, 0.01)	*N*(3, 1.7)	*N*(30,30)	Beta(300, 300)
10	*N*(0.01, 0.01)	*N*(10, 3.2)	*N*(30,30)	Beta(300, 300)
30	*N*(0.01, 0.01/10)	*N*(30, 5.5)	*N*(30,30/10)	Beta(300, 300)

a: $N(\mu,\sigma)$ denotes the normal distribution with mean μ and standard deviation σ.

b: $Beta(a,b)$ denotes the beta distribution with a and b, when $\frac{Y^{1-a}(1-Y)^{1-b}}{B(a,b)}$ is the probability density function of beta distribution. Note that, $B(x,y)$ means beta function

S8 The prior distributions used in the simulation study for various signal-to-background ratios

The weak informative prior distributions were adopted for α, λ_{blk}, and λ_{NP} under the S/B ratio of more than 3. On the other hand, the specific informative prior distributions were adopted under the S/B of 1. For δ_R parameter, the specific informative prior distributions, intended to be almost constant to the expected value of 0.5, were adopted for all S/B ratio cases.

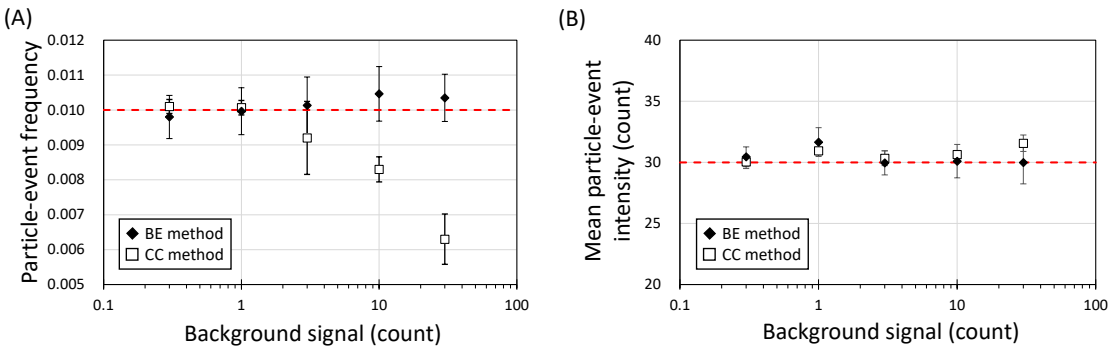


Fig. S9 Analytical results of particle-event frequency (A) and mean particle-event intensity (B) for simulation study under the various signal-to-background ratios: $\alpha = 0.01$, $\lambda_{blk} = (0.3, 1, 3, 10, 30)$, and $\lambda_{NP} = 30$. The error bars in each panel indicate the standard deviation of the triplicated analysis. Horizontal red dashed line indicates true values.