**Supplementary Note**

**Mathematical background**

A linear imaging system with white additive Gaussian noise on the observed data is modeled as follows:

\[ X_i = R_{\varphi_i} V_k + G_i, \tag{1} \]

where \( X_i \in \mathbb{R}^J \) are the experimental, two-dimensional projection data, \( i = 1, \ldots, I \), of \( j = 1, \ldots, J \) pixels each; \( V_k \), with \( k = 1, \ldots, K \), is one of \( K \) underlying three-dimensional structures; \( R_{\varphi_i} \) are projection operations, where \( \varphi_i \) defines the position of the three-dimensional structure with respect to projection image \( X_i \) (as described by three Euler angles and two in-plane translations); and \( G_i \in \mathbb{R}^J \) are images containing independent zero-mean Gaussian noise with standard deviation \( \sigma \).

We optimize the logarithm of the likelihood of observing the entire data set, given a model with parameter set \( \Theta \):

\[ L(\Theta) = \sum_{i=1}^{I} \ln P(X_i \mid \Theta), \tag{2} \]

where \( P(X_i \mid \Theta) \) is the probability of observing data image \( X_i \) given all current model parameters. We treat the assignments to underlying 3D-structures \( k \) and positions \( \varphi_i \) as hidden variables. Expressing \( P(X_i \mid \Theta) \) as an integral over all possible assignments, Eq. 2 can be rewritten as:
$$L(\Theta) = \sum_{i=1}^{l} \ln \sum_{k=1}^{K} \int P(X_i \mid k, \varphi, \Theta) P(k, \varphi \mid \Theta) d\varphi$$

(3)

where $P(X_i \mid k, \varphi, \Theta)$ is the probability of observing data image $X_i$ given $k$, $\varphi$ and $\Theta$; and $P(k, \varphi \mid \Theta)$ is the probability density function of $k$ and $\varphi$.

In (3), we discretize the integral over $\varphi$ by a summation over $p = 1, \ldots, P$ distinct projection directions (described by $P$ combinations of two Euler angles) and a summation over $q = 1, \ldots, Q$ in-plane transformations (consisting of $Q_{rot}$ discrete in-plane rotations and $Q_{trans}$ translations). The log-likelihood function as defined in Eq. 3 can then be written as:

$$L(\Theta) = \sum_{i=1}^{l} \ln \sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{q=1}^{Q} P(X_i \mid k, p, q, \Theta) P(k, p, q \mid \Theta),$$

(4)

Furthermore, we model the 3D-structure $V_k$ as a summation over $L$ basis functions $b_l$ with coefficients $c_{kl}$: $V_k = \sum_{l=1}^{L} c_{kl} b_l$. Let $B_p$ denote the $J$-dimensional column vector that is the projection of $b_l$ in direction $p$. Then it is easy to see that the projection of $\sum_{l=1}^{L} c_{kl} b_l$ is $\sum_{l=1}^{L} c_{kl} B_p = B_p c_k$, where $B_p$ denotes the $J \times L$ matrix whose $l^{th}$ column is $B_{pl}$. We use the notation $X_i(q)$ to indicate that $X_i$ is mapped onto a transformed coordinate system, where $q$ defines the (in-plane) transformation.

Given $k, p$ and $q$, we assume that the probability density function of each pixel value in image $X_i(q)$ can be described by a Gaussian function, centered at the value of the corresponding pixel in the $p^{th}$ projection of the $k^{th}$ underlying 3D structure. The width of these Gaussians is equal to the estimated standard deviation $\sigma$ in the experimental noise. Assuming independence among all pixels in image $X_i$, i.e. assuming white, or independent noise, $P(X_i \mid k, p, q, \Theta)$ can be expressed as a multiplication over $J$ Gaussian functions:
\[
\begin{align*}
P(X_i | k, p, q, \Theta) &= \prod_{j=1}^{J} \frac{1}{\sqrt{2\pi}\sigma} \exp \left( - \frac{\left[ B_p c_k^j \right] - \left[ X_i(q) \right]_j^2}{2\sigma^2} \right) \\
&= \left( \frac{1}{\sqrt{2\pi}\sigma} \right)^J \exp \left( - \frac{\| B_p c_k - X_i(q) \|_2^2}{2\sigma^2} \right)
\end{align*}
\]

where \([X]_j^j\) denotes the value of the \(j^{th}\) pixel of an image \(X\), and \(\|X\|^2 = \sum_{j=1}^{J} [X]^2\)

denotes its squared norm.

We define \(P(k, p, q | \Theta)\), i.e. the probability density function of \(k, p\), and \(q\), as follows. We assume that particle picking has left a two-dimensional Gaussian distribution of
residual translations \(q_x\) and \(q_y\), centered at the origin and with a standard deviation of \(\xi\)

depixels. Furthermore, we assume an even distribution for all \(Q_{rot}\) in-plane rotations, and a

discretized distribution of estimated proportions \(\alpha_{kp}\) of the data belonging to the \(p^{th}\)

projection of the \(k^{th}\) underlying 3D-structure (with \(\alpha_{kp} \geq 0\) and \(\sum_{k=1}^{K} \sum_{p=1}^{P} \alpha_{kp} = 1\)).

Thereby, \(P(k, p, q | \Theta)\) is defined by:

\[
P(k, p, q | \Theta) = \frac{\alpha_{kp}}{\sum_{q=1}^{Q_{rot}} 2\pi\xi^2} \exp \left( - \frac{q_x^2 + q_y^2}{2\xi^2} \right)
\]

The log-likelihood target function, as defined by Eqs. 4-6, is optimized using an

Expectation-Maximization (EM) algorithm. In the \(E\)-step of this algorithm a lower

bound \(Z\) to the likelihood is built, based on the current estimates for the model parameters \(\Theta^{(n)}\), \(n\) being the iteration number:

\[
Z(\Theta, \Theta^{(n)}) = \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{q=1}^{Q} P(k, p, q | X_i, \Theta^{(n)}) \ln \left( P(X_i | k, p, q, \Theta) P(k, p, q | \Theta) \right).
\]
Here, $P(k, p, q \mid X_i, \Theta^{(n)})$ is the probability of $k$, $p$ and $q$, given image $X_i$ and the current estimates for the model. According to Bayes’ theorem this probability can be written as:

$$P(k, p, q \mid X_i, \Theta^{(n)}) = \frac{P(X_i \mid k, p, q, \Theta^{(n)})P(k, p, q \mid \Theta^{(n)})}{\sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{q=1}^{Q} P(X_i \mid k, p, q, \Theta^{(n)})P(k, p, q \mid \Theta^{(n)})}$$  \hspace{1cm} (8)

In the subsequent $M$-step of the EM-algorithm we maximize the lower bound to obtain a parameter set for the next iteration ($\Theta^{(n+1)}$). Recapitulating, $\Theta$ consists of an estimate $\sigma$ for the standard deviation in the experimental noise, an estimate $\xi$ for the standard deviation in the origin offsets, $K \times P$ estimates $\alpha_{kp}$ for the probability density functions of classes $k$ and projection directions $p$, and $K \times L$ estimates for coefficients $c_{kl}$ of the different 3D objects. Setting the partial derivative to the lower bound with respect to $\sigma$ to zero, and solving for $\sigma$ yields:

$$\sigma^{(n+1)} = \sqrt{\frac{1}{IJ} \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{q=1}^{Q} P(k, p, q \mid X_i, \Theta^{(n)}) \sum_{j=1}^{J} \left(B_p c_k \right)_j - \left[X_i(q)\right]_j}^2$$  \hspace{1cm} (9)

Similarly, the updated estimate for the standard deviation in the origin offsets is obtained by:

$$\xi^{(n+1)} = \sqrt{\frac{1}{2IJ} \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{q=1}^{Q} P(k, p, q \mid X_i, \Theta^{(n)}) \left(q_x^2 + q_y^2\right)}$$  \hspace{1cm} (10)

For maximization of the lower bound with respect to $\alpha_{kp}$, we introduce a Lagrange multiplier to constrain $\sum_{k=1}^{K} \sum_{p=1}^{P} \alpha_{kp} = 1$, yielding the following solution:

$$\alpha_{kp}^{(n+1)} = \frac{1}{IJ} \sum_{i=1}^{I} \sum_{q=1}^{Q} P(k, p, q \mid X_i, \Theta^{(n)})$$  \hspace{1cm} (11)
To obtain updated estimates for the 3D-reference structures, we set partial derivatives to coefficients $c_{kl}$ to zero. Dropping constants, we get for all $l$ and all $k$:

$$
\frac{\partial Z(\Theta, \Theta^{(n)})}{\partial c_{kl}} = \sum_{i=1}^{I} \sum_{p=1}^{P} \sum_{q=1}^{Q} P(k, p, q \mid X_i, \Theta^{(n)}) \left[ B_p c_k \right]_j - \left[ X_i(q) \right]_j \left[ B_{pl} \right]_j = 0. \quad (12)
$$

Defining:

$$
A_{kp} = \frac{\sum_{i=1}^{I} \sum_{q=1}^{Q} P(k, p, q \mid X_i, \Theta^{(n)}) X_i(q)}{\sum_{i=1}^{I} \sum_{q=1}^{Q} P(k, p, q \mid X_i, \Theta^{(n)})}, \quad (13)
$$

$$
w_{kp} = \sum_{i=1}^{I} \sum_{q=1}^{Q} P(k, p, q \mid X_i, \Theta^{(n)}), \quad (14)
$$

we rewrite Eq. 12 (for all $k$ and $l$) as:

$$
\sum_{p=1}^{P} w_{kp} \sum_{j=1}^{J} \left[ B_p c_k \right]_j - \left[ A_{kp} \right]_j \left[ B_{pl} \right]_j = 0, \quad (15)
$$

which are the normal equations of the least-squares problem:

$$
\min \sum_{p=1}^{P} w_{kp} \left\| B_p c_k - A_{kp} \right\|^2 \quad (16)
$$

since setting partial derivatives of Eq. 16 with respect to $c_{kl}$ equal to zero yields Eq. 15. Therefore, maximizing the likelihood through solving Eq. 12 for all $k$ and $l$ is equivalent to solving the set of $K$ weighted-least-squares problems in Eq. 16. For this purpose, we apply the following modified algebraic reconstruction technique (ART; based on the methodology introduced by Eggermont, Herman and Lent$^1$) to all $L$-dimensional vectors $c_k$: 


For the inner iteration $m = 0, 1, ..., M$ and $p = 1 + m \mod (P + 1)$, whenever $1 \leq p \leq P$,

$$c_k^{(m+1,n+1)} = c_k^{(m,n+1)} + \sqrt{w_{kp} B_p^T d^{(m,n+1)}},$$  \hspace{1cm} (17)

$$r_{kp'}^{(m+1,n+1)} = r_{kp'}^{(m,n+1)} \quad \text{for} \quad p' \neq p,$$  \hspace{1cm} (18)

$$r_{kp}^{(m+1,n+1)} = r_{kp}^{(m,n+1)} + d^{(m,n+1)},$$  \hspace{1cm} (19)

where the correction image $d^{(m,n+1)} \in \mathbb{R}^J$ is calculated as:

$$\left[ d^{(m,n+1)} \right]_j = \lambda \frac{\sqrt{w_{kp} \left[ A_{kp} \right]_j - \left[ B_p c_k^{(m,n+1)} \right]_j - \left[ r_{kp}^{(m,n+1)} \right]_j}}{w_{kp} B_p^T B_p + 1} \quad \text{for all} \quad j = 1, ..., J. \quad (20)$$

For the special case when $p = P + 1$,

$$c_k^{(m+1,n+1)} = c_k^{(m,n+1)},$$  \hspace{1cm} (21)

$$r_{kp'}^{(m+1,n+1)} = r_{kp'}^{(m,n+1)} - \kappa \sqrt{w_{kp'} B_p^T \sum_{p'=1}^{P} \sqrt{w_{kp'} B_p^T r_{kp'}^{(m,n+1)}}} \quad \text{for all} \quad p' = 1, ..., P. \quad (22)$$

Here, $\kappa$ and $\lambda$ are relaxation parameters, $B_{pj}$ is the $j$th row of $B_p$, and $r_k$ is a vector of $P$ auxiliary images $r_{kp} \in \mathbb{R}^J$, with $\left[ r_{kp}^{(0,n+1)} \right]_j = 0$ for all $j$, $k$ and $p$. For the first iteration of the expectation maximization algorithm ($n = 0$), we start the iterative reconstruction algorithm from all-zero maps ($c_{kl}^{(0,1)} = 0$, for all $l$ and $k$); for all subsequent iterations we start from the resulting map of the previous iteration ($c_{kl}^{(0,n+1)} = c_{kl}^{(M,n)}$, for all $l$ and $k$). After a user-provided number of inner iterations $M$ for the modified ART algorithm, new estimates for all $K$ 3D-structures are obtained as: $c_k^{(n+1)} = c_k^{(M,n+1)}$. 
Implementation Details

The proposed classification approach entails a maximum-likelihood multi-reference angular refinement algorithm that has been implemented in the MLrefine3D program of the open-source package Xmipp\(^2\), which is freely available through http://xmipp.cnb.csic.es. For each iteration, the \( K \) reference maps are projected onto a library of \( K \times P \) reference images, according to an even angular distribution with a user-provided sampling rate. Subsequently, all experimental images are presented to the entire reference library, evaluating Eq. 8 to calculate weights \( P(k, p, q \mid X, \Theta^{(n)}) \) for all \( Q \) in-plane transformations. To limit computational requirements, a previously reported reduced search-space approach is applied to the integration over all origin offsets\(^3\). In addition, once the projection orientations are known with relative confidence, the user may opt to further reduce the integrations to local searches around the optimal angular and translational assignments from the previous iteration (implicitly assuming zero prior probabilities for assignments outside the regions of limited integrations). After all relevant probability weights have been evaluated, new estimates for the standard deviation in the noise and for the probability density functions of the hidden parameters are obtained according to Eqs. 9-11, and updated estimates for the three-dimensional structures are obtained by performing \( K \) independent modified ART-reconstructions according to Eqs. 17-22. Then, the newly derived parameter set \( \Theta^{(n+1)} \) is used for the next iteration of the expectation maximization algorithm, until a user-defined maximum number of iterations \( N \) has been reached.

All likelihood calculations presented in this work were performed using the following set of default parameters. By default, the initial estimate for the standard deviation in the noise is set to 1, the initial estimate for the standard deviation in the origin offsets is set to 3 pixels, and a uniform distribution is used to initialize all \( \alpha_{kp} \). For all 3D reconstructions, relaxation parameters \( \lambda = 0.2 \) and \( \kappa = 0.5 \) are used to perform \( M = 10 \) iterations of the modified ART algorithm. For the basis functions of the reconstructed objects we use blobs as defined by Marabini \textit{et al.}\(^4\), with the Xmipp default parameters of \( a = 2, m = 2 \) and \( \alpha = 10.4 \).
Since fast two-dimensional Fourier transforms are employed for the integrations over multiple translations, the required computing times are of order $O(J \log \sqrt{J})$ for $J$ pixels. Furthermore, the computation times depend linearly on the number of experimental images, the number of images in the reference library, and the number of sampled in-plane rotations. The rate-limiting step is the evaluation of probability weights $P(k, p, q | X_i, \Theta^{(n)})$ for all images (Eq. 8), since it implies an extensive search over many in-plane transformations and reference projections. Fortunately, this step can be parallelized with relative ease, and the current implementation employs a message passing interface (MPI) to perform these calculations independently for multiple subsets of the data. We did not implement a parallel version of the modified ART algorithm, as the $K$ reconstructions are already independent and typically require two orders of magnitude less CPU time than the calculation of the probability weights.

To further limit the required computing times, the current implementation stores the Fourier transforms of multiple in-plane rotated versions of all reference projections in memory. Similarly, rotated versions of all (running) weighted sums for images $A_{kp}$ (Eq. 13) are also stored in memory. This speeds up the evaluation of the probability weights, as only a limited amount of image rotations needs to be performed for each of the experimental images. However, this also implies that the memory requirements increase rapidly with increasing angular sampling rates. While a $10^\circ$ sampling with $K = 4$ and images of $64 \times 64$ pixels require approximately 900 Mb, a $5^\circ$ sampling would take approximately 8 times more, exceeding the limits of our currently available hardware. Although obviously memory may be traded for computing or disk-access time, we as yet did not perform such computationally demanding tests.

With the current implementation, classification of the large T-antigen data took one CPU month (on an Intel Xeon 3GHz processor), performing 20 iterations of likelihood optimization with exhaustive angular searches and an angular sampling rate of $15^\circ$. For the ribosome data, 25 iterations of likelihood optimization with an angular sampling rate of $10^\circ$ took a total of 6 CPU months. In this case, the calculations were considerably speeded up by local searches. While exhaustive searches required on average 18 CPU days per iteration during the first 7 iterations, local searches of +/- $20^\circ$
in projection directions and +/- 2 pixels in origin offsets required approximately 3 CPU days per iteration during the remaining 18 iterations.

References


