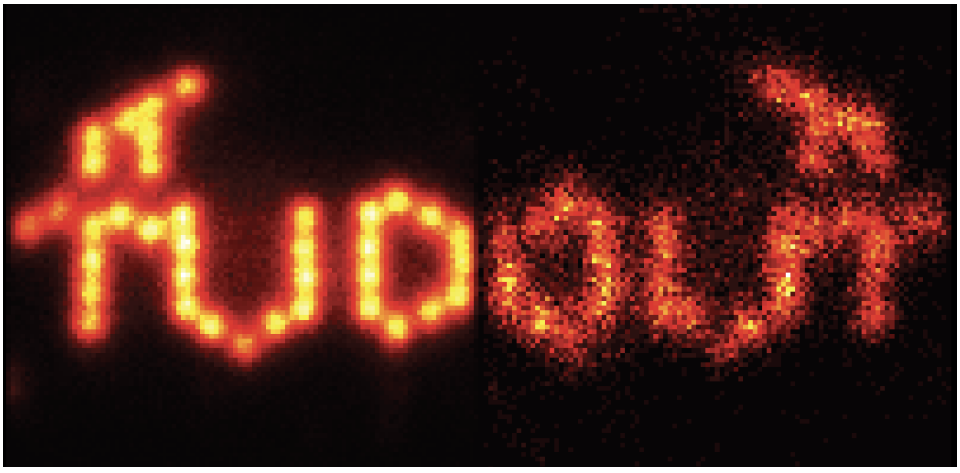
Detecting structural heterogeneity in single-molecule localization microscopy data

by

T.A.P.M. Huijben

to obtain the degree of Master of Science
at the Delft University of Technology and Erasmus Medical Center Rotterdam,
to be defended publicly on Wednesday August 26, 2020 at 13:00 PM.



Student number:	4403843	
Project duration:	November 1, 2019 – Augustus 26, 2020	
Committee:	Prof. B. Rieger,	TU Delft, supervisor
	Prof. S. Stallinga,	TU Delft
	Dr. F. M. Vos,	TU Delft
	Dr. C. S. Smith,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft ImPhys

Preface

This thesis comprises multiple chapters that describe in detail the classification of structurally heterogeneous samples imaged with single-molecule localization microscopy.

The first chapter contains a general introduction to the field of (localization) microscopy. Here I introduce the research problem of my thesis: particle fusion conceals structurally different particles. The second chapter is the Manuscript I wrote under supervision of Bernd Rieger and Sjoerd Stallinga, which we will soon submit for publication. The Manuscript, together with its supplement, contains a thorough description of the classification algorithm and the performance on multiple experimental and simulated datasets. The third chapter contains further details, considerations, alternatives and applications of the classification method described in the Manuscript. In the fourth and concluding chapter, I discuss the strong and weak points of the developed method and shed light on possible future directions of this research. The appendix consists of a literature review about image classification techniques for cryo-electron microscopy. This review is not officially part of my thesis, since it has been evaluated and graded already. Although, the classification techniques used in electron microscopy have been a great inspiration for my work.

Teun Huijben
Delft, August 2020

Abstract

For decades, the resolution of fluorescent light microscopy has been bounded by Abbe's diffraction limit to $\lambda/2NA$. Super-resolution methods, awarded with the 2014 Nobel Prize in Chemistry, use tricks to overcome this limit. The general idea is to image blinking fluorophores for a multitude of frames, such that each frame only contains a sparse subset of fluorophores. Assuming that single emitters give rise to a sparse subset of diffraction-limited spots, their locations can be determined with nanometer precision. The resolution of the final reconstructed image is limited by the localization precision and incomplete fluorescent labeling. To even further improve the resolution, by increasing the signal-to-noise and overcoming the problem of a low labeling density, single-particle averaging can be used if multiple copies of the same target particle (e.g. macromolecular complex) can be imaged. All emerging localization patterns are computationally merged into one super-superresolution image. Despite the increase in resolution, potential structural variation among the particles will blur the particle fusion result and possible (small) subsets of structurally different particles cannot be detected in the reconstruction. We present an a-priori knowledge-free, unsupervised classification method that splits the dataset into conformationally different groups of images prior to the merging process, which can subsequently be fused per class. The implemented algorithms are validated on multiple experimental and simulated datasets. We achieved classification performances of 96% on experimental datasets with up to four different DNA origami structures, are able to detect rare classes of mirrored origami's occurring at a rate of 2%, and capture the variation in the ellipticity of nuclear pore complexes. This new classification tool will allow microscopists to study heterogeneous samples with single-particle averaging techniques and discriminate between different particles, structures or conformations with a high resolution.

Contents

Preface iii

Abstract v

1 Introduction 1
- 1.1 Superresolution microscopy . . . 1
- 1.2 Single-particle averaging . . . 2
- 1.3 2D particle fusion . . . 3
- 1.4 Conformational variability concealed by particle fusion . . . 4
- 1.5 Comparison to electron microscopy . . . 4

2 Manuscript 7
- Main text . . . 8
- Figures . . . 11
- Methods . . . 14
- Supplementary information . . . 16

3 Extended method development, applications and considerations 29
- 3.1 Normalization of the Bhattacharya cost function . . . 29
- 3.2 Hierarchical agglomerative clustering approach as alternative to MDS . . . 35
- 3.3 Refining the classification result . . . 40
- 3.4 3D classification . . . 47

4 Discussion 49

A Classification techniques in single-particle averaging for cryo-electron microscopy 53

Bibliography 59

1

Introduction

Microscopy is an important workhorse in current day biomedical research. Researchers gain a better understanding of the biological object of interest, such as a tissue, a cell, a cellular component or even a single protein, by looking at microscopy images of the object. Presently, there are two major microscopy techniques available to study biological materials in detail, light- and electron microscopy. In cryogenic electron microscopy (cryo-EM), biological macromolecules are rapidly frozen in a thin layer of virtuous ice. An electron beam is passed through the sample creating a tomographic 2D projection of the molecule on a planar detector. This technique gives the highest resolution but suffers from elaborate sample preparation and can only be used on a fixed sample.

Fluorescence light microscopy, however, relies on a completely different imaging principle. Here, fluorescent molecules, fluorophores, are attached to specific locations on the molecule of interest. After illuminating the sample with light of a specific wavelength, the fluorophores absorb the light and subsequently emit light of a longer wavelength. The illumination light is separated from the much weaker emitted light by using filters. In this way, the fluorescent signal originating from the fluorophores is detected on the camera, which gives indirectly information about the molecule of interest. Fluorescence light microscopy gives, compared to cryo-EM, higher contrast, allows for specific labeling, allows for multiple labels, can be used at physiological temperatures and is compatible with living cells.

Despite the many advantages of fluorescence light microscopy, the obtainable resolution is theoretically bounded. Abbe's diffraction limit tells us that the best resolution one can obtain is $\lambda/(2NA)$, where λ is the wavelength of light and NA the numerical aperture of $n \cdot \sin(\alpha)$, where n is the refractive index of the immersion medium and α the marginal ray angle of the collected light. Using reasonable parameters for the wavelength of visible light and the numerical aperture, the highest resolution of 200 nm can be obtained. This resolution is sufficient to visualize subcellular complexes, but is not high enough to discriminate individual proteins [36].

1.1. Superresolution microscopy

Due to the diffraction limit, each fluorophore creates a 200 nm blurred spot in the image, which means that if two fluorophores are in close proximity, they cannot be distinguished as unique molecules. In recent years, multiple methods have been published that try to overcome the diffraction limit and bring the resolution of light microscopy (LM) closer to the subnanometer resolution of electron microscopy (EM). The general idea is to discriminate individual fluorophores by separating their emission light in time. Here, we will discuss this idea shortly.

The main idea of this technique is imaging a multitude of frames with blinking fluorophores. The blinking nature of the fluorophores makes that in each frame only a subset is present (Fig. 1.1). Consequently, it is highly unlikely that two nearby fluorophores, within the diffraction-limited distance from each other, are emitting photons simultaneously. By assuming, therefore, that each spot in the image represents a single fluorophore, the centroid of the spot can be determined typically with nanometer precision. This procedure results in coordinates accompanied with a localization uncertainty for every fit.

The final super-resolved image is a rendering of the list of coordinates of the calculated centroids of the time-separated diffraction limited dots that are detected in all frames.

Different methods are used to make sure that not all fluorophores are constitutively active. In PALM (photoactivated localization microscopy) the fluorescent proteins are photoactivated randomly and bleach after they have emitted light [3]. In STORM (stochastic optical reconstruction microscopy) the fluorophores undergo rapid photo-switching cycles, meaning that they are present in multiple series of frames [38]. In DNA-PAINT (points accumulation for imaging in nanoscale topography), dye-labeled DNA probes dynamically bind and unbind binding sites on the substrate [23]. Since the probes are constantly replaced, photobleaching is negligible and each site will have approximately the same number of localizations.

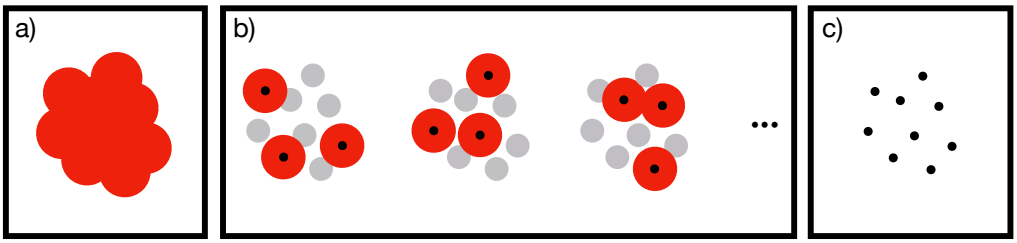


Figure 1.1 | Principle of localization microscopy. a, Conventional fluorescence microscopy image of nine fluorophores that are positioned closely together. **b**, In localization microscopy, the fluorophores are blinking stochastically, which has as a result that only a few are emitting photons in each frame (red dots). Three of the many frames are shown here. The centroids (black dots) of the spots can be found with high precision. **c**, Computer rendered superresolution image of the centroids of the emitting fluorophores in all frames. (Image inspired by [14])

1.2. Single-particle averaging

The technique of imaging blinking fluorophores for many frames, estimating centroids of fluorophores and rendering the final image by plotting the coordinates, is referred to as single-molecule localization microscopy (SMLM). The resolution of the final reconstructed image is limited by the localization uncertainty and incomplete fluorescent labeling [31]. The latter means that not all labeling sites of the structure of interest are occupied by fluorophores, resulting in an incomplete image. Single-particle averaging can be used to improve the resolution even further, if multiple copies of the same target particle (e.g. a macromolecular complex) can be imaged. Averaging multiple structures will increase the signal-to-noise ratio, since effectively it overcomes the problem of low labeling density of single particles. All emerging localization patterns are computationally merged into one super-superresolution image. An example is the particle fusion result given by Löschberger *et al.* [25] of the nuclear pore complex (NPC). The NPC is a big complex present in the nuclear membrane of eukaryotic cells and acts as the gateway between the nucleus and the cytoplasm for proteins and other molecules. The NPC is composed of multiple copies of more than 30 different proteins. These proteins are arranged into eight homomeric subunits that form a circular structure. Single-particle averaging of *d*STORM images of labeled *Xenopus laevis* pores clearly shows the eightfold symmetric nature of the pore complex, resolved with 7 nm resolution (Fig. 1.2).

There are different strategies to perform particle fusion. The first method registers all particles to a chosen template [25, 44]. Here, the number of required registrations scales linearly with the number of particles. The downside is that template registrations suffer from template-bias. Consequently, the final reconstruction tends to resemble the template, irrespective of the presence of such structure in the data [16]. A second strategy, which is less sensitive to template-bias, is the pyramid registration [5]. In this algorithm, the particles are registered in an iterative pairwise fashion. The fusion of pairs of particles is repeated until all particles are fused and one final reconstruction is created. The disadvantage of the pyramid approach is the dependency on the choice of initial particles, which act effectively as templates. Registration errors in the initial step will be propagated throughout the entire registration pipeline and will have a major effect on the final result.

A third method, which is template-free and does not suffer from initialization bias, is the 2D particles fusion pipeline published by our lab [17]. This method finds the global reconstruction by performing all-to-all registration, where every particle is registered to every other particle. Here, we will explain this method in more detail.

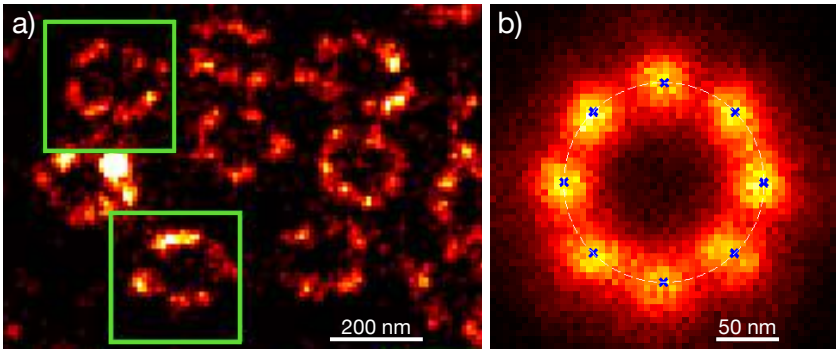


Figure 1.2 | Single-particle averaging result of the *Xenopus laevis* nuclear pore complex. a, Superresolution *d*STORM image of multiple nuclear pore complexes, labeled with Alexa647 bound with a secondary antibody to nucleoporin gp210, imaged with *d*STORM. Isolated NPCs are outlined by green squares and used for further averaging. **b**, The superresolution averaged structure of 426 pore complexes, which are combined by rotational alignment. (Image adjusted from [25])

1.3. 2D particle fusion

The particle fusion pipeline exists of three main steps [17]. In the first step, every particle is registered to every other particle. A Gaussian-mixture-model-based (GMM) registration, initialized with multiple angles, results in transformation parameters for each angle. The optimal transformation is determined by calculating the Bhattacharya cost function for each found GMM initialization:

$$\mathrm{D(a,b)} = \sum_{i=1}^{K_a} \sum_{j=1}^{K_b} \exp\left(-\frac{(\vec{r}_{a,i} - \vec{r}_{b,j})^2}{\sigma_{a,i}^2 + \sigma_{b,j}^2} \right), \tag{1.1}$$

where the two particles a and b are represented by K_a and K_b localizations, r_a and r_b are the localization coordinates and σ_a and σ_b the isotropic localization uncertainties. There is no need to convert the particles to pixelated images since this cost function has the advantage that it works directly on the localization coordinates and uncertainties. The all-to-all registration procedure results in $N(N-1)/2$ transformation parameters for N particles (Fig. 1.3), containing the relative orientation between two respective particles.

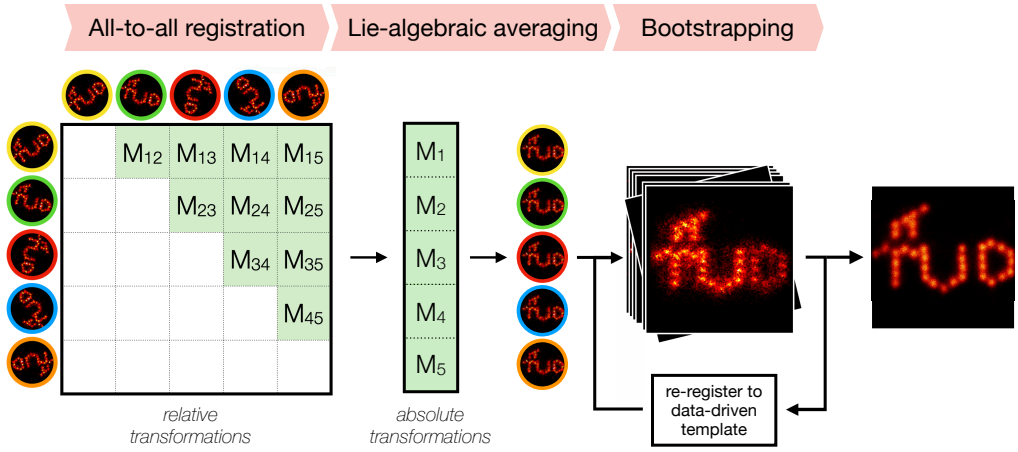


Figure 1.3 | The 2D particle fusion pipeline. In all-to-all registration all N particles are registered to all other particles, resulting in $N(N-1)/2$ relative transformation parameters (M_{ij}). The absolute transformation parameters (M_i) are obtained by Lie-algebraic averaging of the relative parameters. The absolute transformations are used to align all particles. In bootstrapping, the reconstruction is improved by iteratively re-registering every particle to an ensemble of the remaining particles.

To obtain a global alignment of all particles, the relative transformation parameters are converted to absolute transformations. This conversion is performed in the second step of the pipeline, using Lie-algebraic averaging. Averaging over all relative transformations provides robustness to mis-registrations, since we make use of the large redundancy of the $N(N-1)/2$ relative transformation parameters. Direct averaging of the transformation matrices is not possible, since the rotational part of the transformation is modulo 2π.

This problem is overcome by using the Lie-algebraic representation of the transformation. The Lie-algebraic averaging procedure is performed twice, where the result of the first averaging step is used to eliminate outliers from the all-to-all registration matrix for the second averaging step. The N absolute transformation parameters describe the orientation of all particles to a common reference frame and are used to transform the N particles to get them aligned (Fig. 1.3).

In the third and last step of the particle fusion pipeline, the alignment of the particles is refined with bootstrapping. Here, the individual particles are one-by-one re-registered to a resampled ensemble of all other particles. This process is iteratively repeated for all particles for multiple rounds, resulting in an improved reconstruction.

1.4. Conformational variability concealed by particle fusion

Fusing the localizations from many imaged molecules is necessary to increase the signal-to-noise ratio (SNR) and to overcome the problem of underlabeling. Despite the increase in resolution, possible conformational differences between groups of particles will disappear in the fusion process. In this thesis, we will develop and implement a multi-state classification method to split the dataset into conformationally different groups of images and fuse them separately (Fig. 1.4). When a heterogeneous sample contains molecules that differ from each other, the fusion result will conceal the differences. Classifying the data into different groups prior to the particle fusion is important to preserve the heterogeneity in the data. It is especially needed in cases where different conformations only vary slightly, or when only a small subset is different. Using current techniques, these differences between conformational states will be lost.

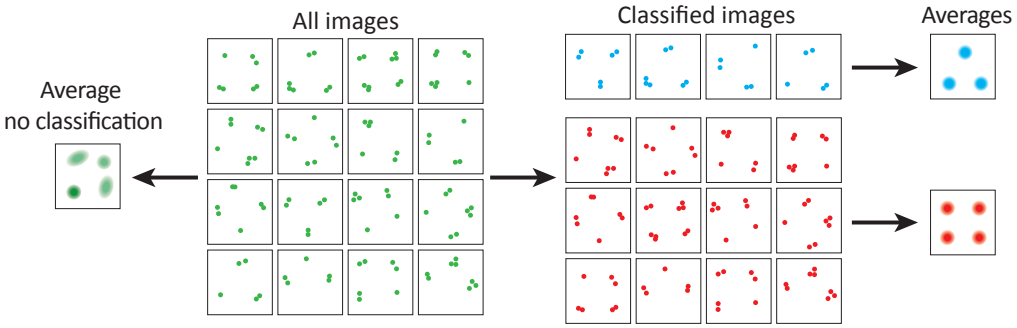


Figure 1.4 | Principle of multi-state classification in single-particle averaging. All particles (green) contain square and triangular structures, which is hard to see without particle fusion. Averaging without classification results in a unclear fusion that is in between a square and a triangle (left green image). When multi-state classification is used to group the images into a number of clusters (two in this example) and average per cluster, the two different structures are clearly visible (blue and red average on the right side).

The implemented classification algorithms will be tested and applied to different types of experimental data. One of the datasets is the nuclear pore complex data shown in Fig. 1.2 [25]. The other type of data is generated using DNA origami constructs. In DNA origami, a long single-stranded DNA molecule, referred to as the scaffold, is folded into the desired shape by binding specifically designed short DNA oligos, known as the staples [37]. Besides the favourable property of being able to design and construct any desired shape, DNA origami allows for imaging with DNA-PAINT. Here, the staples are extended with a short single-stranded docking site complementary to the DNA sequence of dye-labeled imaging probes. Since the affinity between the imaging probes and the docking sites is weak, the probes constantly bind and unbind the DNA structure. The stochastic binding characteristics of these imaging probes creates the desired temporal sparsity required for single-molecule localization microscopy. The constant turnover of fluorophores solves the problem of photobleaching of certain docking sites during the lengthy acquisition, which is a common problem in STORM and PALM.

1.5. Comparison to electron microscopy

Currently, cryo-EM is used for studying biological structures with nanometer precision. Despite the higher resolution, EM has many disadvantages compared to light microscopy (LM). In the first place, LM allows for specific labeling whereas in EM it is hard to distinguish different molecules. In LM, labeling multiple sites is possible, which allows to measure the relative distances between molecules and it aids in identifying

different structures in correlative EM images. Normally, this is difficult, because everything in EM is greyscale and challenging segmentation methods are needed to identify the structures. The second advantage of LM is that it is cheaper and easier than EM. The experiments require less preparation and the setup is simpler and more affordable. Thirdly, LM requires significantly fewer images of identical particles compared to EM. Consequently, the variability between different molecules can be detected more easily since fewer images are necessary and the images themselves have a higher SNR. The final advantage is that in LM it is easier to extend the reconstruction to 3D than in EM, since the EM images are 2D projections of the electron density of a 3D object, and the 3D structure must be reconstructed from all these projections. In LM, the data is inherently 3D, so registration and particle fusion are sufficient to obtain a superresolution 3D reconstruction.

Multi-state classification is already widely used for single-particle averaging in the field of cryo-EM. See chapter A of this thesis for a review on classification strategies used in cryo-EM. The averaging algorithm from the cryo-EM field can be applied to SMLM data [13, 40] after converting the localization coordinates to pixelated images, by binning the localizations in a 2-dimensional histogram. However, these methods suffer from the incomplete labeling problem in SMLM data. We will adopt these classification methods from the cryo-EM field and tailor them for localization microscopy. Classification in SMLM will enable us to study heterogeneous samples with single-particle averaging techniques. Finally, it will allow for preserving and finding different structures within a dataset at high resolution, that would otherwise be lost due to averaging.

2

Manuscript

Detecting structural heterogeneity in single-molecule localization microscopy data

Teun A.P.M. Huijben[1]*, Hamidreza Heydarian[1]*, Alexander Auer[2,3], Ralf Jungmann[2,3], Sjoerd Stallinga[1] and Bernd Rieger[1]

1 Department of Imaging Physics, Delft University of Technology, The Netherlands
2 Department of Physics and Center for Nanoscience, Ludwig Maximilian University, Munich, Germany.
3 Max Planck Institute of Biochemistry, Martinsried, Germany

Abstract
Particle fusion for single-molecule localization microscopy improves signal-to-noise ratio and overcomes underlabeling, but ignores structural heterogeneity or conformational variability. We present a-priori knowledge-free unsupervised classification of structurally different particles employing the Bhattacharya cost function as dissimilarity metric. We achieve 96% classification accuracy on mixtures of up to four different DNA origami structures, detect rare classes of origami's occuring at 2% rate, and capture variation in ellipticity of nuclear pore complexes.

Main text
Single-molecule localization microscopy (SMLM) enables imaging below the diffraction limit [15, 24]. The image quality can be improved further by fusing hundreds of super-resolution images of identical bio-molecular structures, further referred to as particles, into a single reconstruction [17, 25, 40, 44]. This approach overcomes the problem of incomplete labeling using the central assumption that all particles represent the same underlying structure. In reality, however, the sample might be heterogeneous in structure due to the biology itself, sample preparation or drug induced variations. These potential variations between structures blur standard fusion and keep small subsets of structurally different particles undetected. Sporadic 9-fold symmetric nuclear pores [19, 26], for example, cannot be detected in the reconstruction.

Image classification is commonly used in single-particle averaging (SPA) for cryo-electron microscopy (cryo-EM) [9, 45] to find the viewing direction of each particle from its projection. Using EM classification techniques as such to SMLM data [13, 40] does not employ the full potential of SMLM data, due to the different image formation process, in particular the incomplete labeling [5, 6], and the use of localization coordinates instead of pixelated images. Previous work to separate classes in SMLM uses a deep neural network for classification [1]. Here, however, the different classes need to be known a-priori and imaged in separate experiments to form learning sets in order to train the neural network. This strong a-priori knowledge is not compatible with discovering unknown data variation and is, therefore, inapplicable to most cellular imaging applications.

Here, we present an unsupervised classification tool to cluster SMLM data into classes, that assumes no prior knowledge about the different classes. Our approach uses pairwise registration of all particles to obtain a dissimilarity metric. Subsequently, a feature space is obtained by performing multidimensional scaling (MDS) on the dissimilarity representation which is followed by k-means clustering and particle fusion per cluster (**Fig. 1**).

We employ the Bhattacharya cost function, which we used earlier in the all-to-all registration of template-free particle fusion [17] (**Methods**). In contrast to this earlier use, we use the optimum value of this cost function as similarity metric, we are not interested in their relative translations and rotations. The Bhattacharya metric works directly on the localization data, takes (possibly anisotropic) localization uncertainties into account, and is robust against underlabeling. The pairwise registration of N particles results in $N(N-1)/2$ similarity values. After converting the similarity to dissimilarity values, we use MDS to translate the pairwise dissimilarities into spatial coordinates of the particles in a multidimensional space [29]. This constellation preserves the pairwise dissimilarities by minimizing the so-called metric stress loss function (**Methods**).

Clusters are obtained by applying k-means clustering in the multidimensional scaling space [21]. Merging the particles per cluster is quick since the computationally most intensive procedure, the all-to-all registration, is already performed prior to the classification. This fast reconstruction makes it possible to investigate

multiple values for the numbers of clusters, *K*. A small value of *K* (~5) works well for classes with equal number of particles or for splitting a continuous range of structural variation. For detecting a small subgroup of particles, a large value for *K* is advised (~50), where the obtained *K* clusters can be further grouped by using classification via eigen images (**Supplementary Note 1**).

We applied our classification algorithm on different multi-class SMLM datasets in **Figures 2** and **3**. We show the nanoTRON dataset (previously described [1], further referred to as 'digits'), which consisted of four classes: the digits 1, 2, 3 and a 3x4 grid (**Fig. 2a**). Each class comprised 2,500 instances of each structure. The data was imaged separately per class. We randomly selected 5,000 particles from all classes and fused them without classification. This resulted in a blurred reconstruction which resembled vaguely the digit 3 (**Fig. 2b**), caused by different brightness levels per class (**Supplementary Fig. 1a**). Classification of the particles followed by fusion, however, resulted in four clearly distinct classes (**Fig. 2c-f**). The performance of the classification was quantified by the confusion matrix (**Fig. 2g**) as we knew the ground truth class of each particle, we found that 96.4% is correctly classified.

To demonstrate that the classification is not selecting on different imaging conditions between the four classes, we repeated the same classification procedure on 5,000 particles, 1,250 per class, where all four designs were mixed in a single sample and were acquired together. Here, the fusion without classification showed the digit 1, again explained by different brightness levels (**Supplementary Fig. 1b**). The fusions after classification in **Fig. 2h-l** show similar improvements as before, with an overall classification performance of 96.4% (**Fig. 2m**), using manual labeling [1] as ground truth. The worse reconstruction of digit 1 (**Fig. 2c**) and digit 2 (**Fig. 2j**) compared to other classes, is due to lower quality of the individual particles in this experiment (**Supplementary Fig. 2**), not to suboptimal classification performance. The Fourier ring correlation (FRC) image resolution [31] after reconstruction ranges from 3.7 nm to 5.7 nm per class (**Supplementary Fig. 3**).

We further tested our algorithm on a more challenging DNA-origami dataset that is imaged with DNA-PAINT (**Methods**), containing three designed structures: the letters T, O and L (compare **Fig. 2n**). These structures are more similar to each other and therefore more difficult to separate. We investigated a set containing 600 particles (200 per class) which were imaged separately. Reconstruction without classification gave an unrecognizable outcome (**Fig. 2o**), whereas classification prior to fusion resulted in clearly visible classes (**Fig. 2p-r**). In contrast to the digits dataset, this dataset contained misfolded and unrecognizable particles (**Supplementary Fig. 4**). Therefore, classification into four classes was needed to capture the misfolded structures into a separate class (**Fig. 2s**). An average classification performance of 97.3±1.9% was obtained for two independent repeats (**Fig. 2t**). The same three classes were visible after classification when all three designs were acquired from a single sample (**Fig. 2u-w**). Here two 'misfold classes' were needed as the misfolds themselves vary (**Supplementary Fig. 5**). The digits dataset did not need a separate misfold class, as particle picking was successful in automatically removing all misfolded particles.

In addition to datasets that contain multiple, equally abundant classes, we tested the classification algorithm on experimental data with skewed distributions for the amounts of particles per class. The TUD-logo DNA-origami data [17] contains a small fraction that is unintentionally imaged upside-down, resulting in mirrored logos. Normally, these mirrored images are manually removed from data before fusion. Both for 80% and 50% density of labeling (DoL), the mirrored particles were not visible when all of them are fused (**Fig. 3a,d**), due to their low abundance. However, classification found these classes containing, respectively, 10 and 8 mirrored particles (**Fig. 3c,f**), from a total of 456 and 381 particles. We used *K=4* for the 80% DoL data and *K=40* and for the 50% DoL data, followed by further grouping (**Supplementary Note 1**). To test how low we can go in the detection of rare events, we also applied our method to simulated nuclear pore complexes (NPC), with a rare 9-fold symmetric class. Classification could find these cases, even when only 2% was 9-fold symmetric (**Supplementary Fig. 6**). For a low rare class rate, it was necessary to increase the total number of particles as the rare class needs at least approximately 10 particles for successful detection.

Next to PAINT data, we applied our method to data acquired with *d*STORM of the integral membrane protein gp210, part of the NPC (data previously described [25]). Previously, Heydarian *et al.*[17] used prior knowledge about the 8-fold symmetry for particle fusion and obtained a circular reconstruction with a homogeneous distribution of localizations. We applied our method to the same data and found classes with a range of ellipses (**Fig. 3g-l**). The classification separated the particles on their ellipticity (**Supplementary Fig. 7c,d**). The major axes of the ellipses align in the field-of-view, suggesting that the deformations were caused by the sample preparation (**Supplementary Fig. 7a,b**).

We further characterized and quantified the classification performance of our method on simulated data with discrete variation, TUD-logos with and without flame, and continuous variation, NPCs with a varying diameter (**Supplementary Fig. 8**) In the latter case, the classification perfectly separates the NPCs on diameter. When a dataset had no structural variations, but was classified nevertheless, we found differences in localization densities between the particles grouped into different classes (**Supplementary Fig. 9**).

In summary, we have developed an a-priori knowledge-free classification tool to identify (rare) structural subclasses in SMLM data. Once data heterogeneity is found, particle fusion algorithms can be employed. We successfully classified experimental PAINT and *d*STORM datasets, containing either multiple structures, rare subclasses or continuous structural variation. In the future, our method can be applied to detect structural variations in SMLM datasets.

Author contributions
T.A.P.M.H. and H.H. developed the method. T.A.P.M.H. wrote code, performed simulations and analyzed data. B.R. and S.S. directed the research. A.A. and R.J. designed DNA origamis and acquired images. T.A.P.M.H., wrote the paper with support from S.S. and B.R. and all authors commented on the paper.

Funding
This work was supported by the Dutch Research Council (NWO), VICI grant no. number 17046, to B.R. and European Research Council (MolMap, grant no. 680241 to R.J. and Nano@cryo, grant no. 648580 to B.R. and H.H.).

Software
The algorithm is available as source code written in Matlab, CUDA and C. The software is available for download under the terms of the MIT license from ftp://qiftp.tudelft.nl/rieger/outgoing/... (available on Github at a later time).

Figures

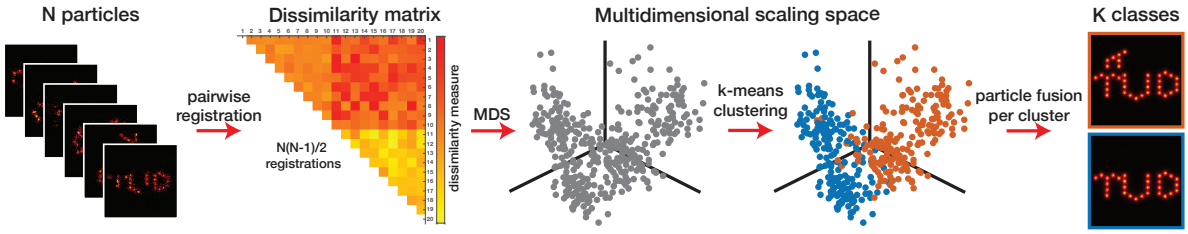


Figure 1 | Classification pipeline. N particles are pairwise registered, resulting in $N(N-1)/2$ dissimilarity values. Multidimensional scaling (MDS) embeds the elements of the dissimilarity matrix in a multidimensional space (only the first 3 dimensions are shown). K-means clustering in this space results in K clusters and the particles are fused per cluster.

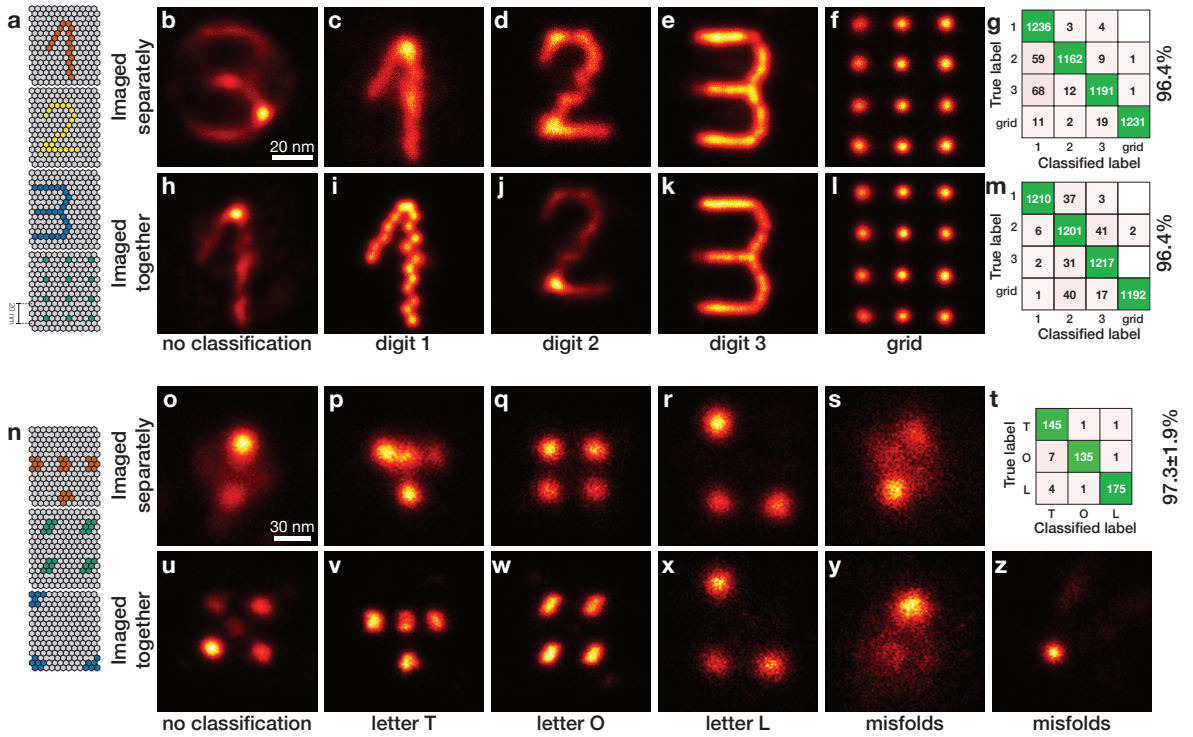


Figure 2 | Classification of experimental DNA-origami datasets. **a**, Templates of four DNA-origami designs in the digits dataset: the digits 1, 2 and 3 and a 20 nm grid. **b**, Fusion result without classification of 2,000 particles (randomly selected from a set of 10,000 images, 2,500 per class). This data is imaged separately per class and combined into one dataset prior to the classification. **c**-**f**, The four classes resulting from classification of 5,000 images (randomly selected from a set of 10,000 images, 2,500 per class) containing 1374, 1179, 1223 and 1233 particles, respectively. **g**, Confusion matrix of the classifications **c**-**f** with an overall performance of 96.4%. **h**, Fusion result without classification of 2,000 particles imaged in one FOV. **i**-**l**, The four classes resulting from the classification of 5,000 particles images in one FOV, containing 1219, 1309, 1278 and 1194 particles, respectively. **m**, Confusion matrix of the classifications **i**-**l**, with an overall performance of 96.4%. **n**, Templates of three DNA-origami designs in the letters dataset: 'letters' T, O and L. **o**, Fusion result without classification of 600 particles (200 per class). This data is imaged separately per class and combined into one dataset prior to the classification. **p**-**s**, The four classes resulting from the classification of **o**, containing 207, 122, 176 and 95 particles, respectively. **t**, Average confusion matrix of two independent classifications performed as in **p**-**s** with an average performance of 97.3±1.9%, where the class of misfolds, **s**, is not taken into account. **u**, Fusion result without classification of 800 particles imaged in one FOV. **v**-**z**, The five classes resulting from the classification of **u**, containing 170, 238, 130, 139 and 123 particles, respectively. Scale bar of **b** applies to **c**-**f** and **h**-**l**. Scale bar of **o** applies to **p**-**s** and **u**-**z**.

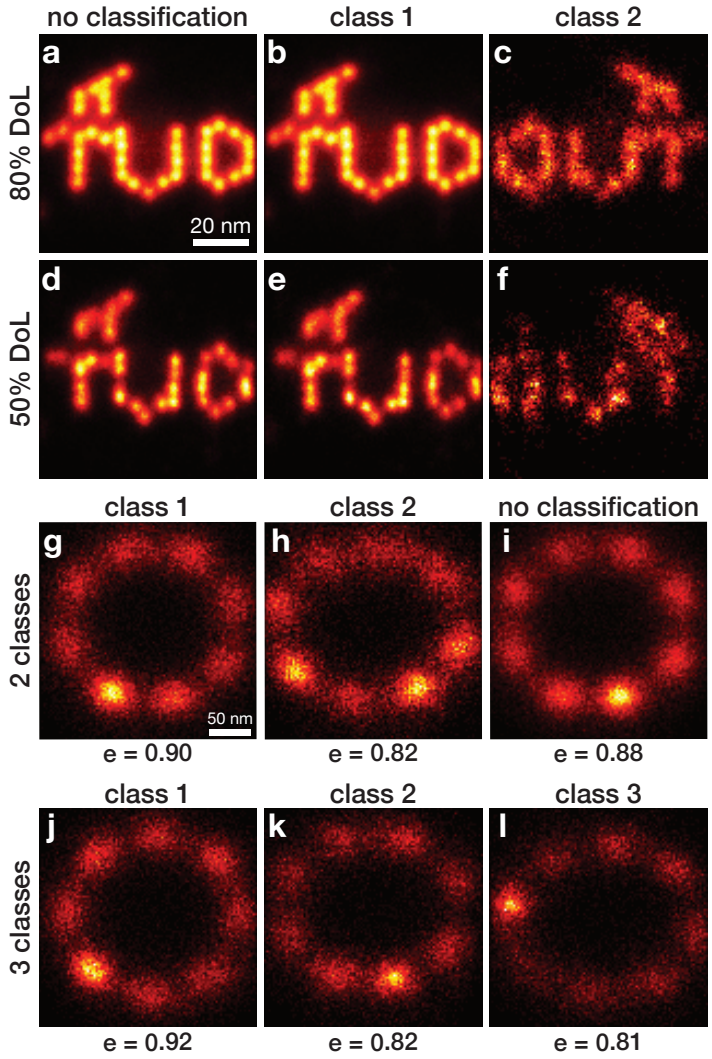


Figure 3 | Classification of 'flipped' DNA-Origami data and NPC data. a, Fusion result without classification of 456 DNA-origami structures of the TUD-logo with 80% DoL. **b**-**c**, The two classes resulting from classification of **a**, containing 446 (normal orientation) and 10 (flipped orientation) particles per class, respectively. **d**-**f**, Same as **a**-**c**, but with 50% DoL and 381 and 8 particles per class, respectively. **g**-**h**, The two classes resulting from classification of **i**, containing 168 and 136 particles, respectively. **i**, Fusion result without classification of 304 NPCs. **j**-**l**, Same as **g**-**h**, but with classification into three classes, with 133, 90 and 81 particles per class, respectively. For **g**-**l**, the ellipticities are indicated below each class. Scale bar of **a** applies to **b**-**f**. Scale bar of **g** applies to **h**-**l**.

Methods

Classification

The developed clustering or classification pipeline (**Fig. 1**) consists of four main building blocks: (1) pairwise registration of all particles resulting in an upper triangular matrix of dissimilarity values; (2) multidimensional scaling based on the dissimilarity representation; (3) k-means clustering of the multidimensional embedding; and (4) per cluster particle fusion.

Pairwise registration

The registration of every particle to every other particle is performed as in Heydarian *et al.* [17]. A Gaussian-mixture-model-based (GMM) registration in combination with the Bhattacharya cost function results in transformation parameters and a cost function value for each pair of particles. The GMM registration [22] finds the optimal transformation by placing a Gaussian distribution onto every localization and maximizing the overall overlap between all Gaussians of the two particles. The Gaussian distributions are all given the same width, the so-called scale, which is a parameter that is dataset specific. The optimal scale parameter is determined by performing a scale-sweep in the range 0.001 - 0.5 camera pixels (corresponding to 0.13 - 65 nm). Since the minimum spacing between fluorophore binding sites ranges from 5 nm (DNA origami) to 60 nm (*Xenopus* nuclear pore), this range is sufficient. A smaller scale would result in overfitting on individual localizations and a larger scale would blur the localizations belonging to different binding sites. In the scale-sweep, 10 random combinations of particles are registered using GMM for 50 scales linearly distributed over the above range. The optimal scale parameter corresponds to the maximum GMM value after normalizing and averaging the obtained 10 GMM values over all scales. This procedure results in a scale value of 0.03 for the digits, 0.15 for the letters, 0.1 for NPC and 0.01 for the TUD-logo (in camera pixels). Each GMM registration is initialized with 6 angles (uniformly distributed between 0 and 2π) and the optimal registration is determined by evaluating the Bhattacharya cost function (eq. 1 of [17]) on the 6 found transformations. The Bhattacharya cost function used in Heydarian *et al.* is adapted by adding a normalization with respect to the number of localizations and with respect to the localization uncertainties as follows:

$$\mathrm{S(a,b)} = \frac{1}{K_a K_b} \sum_{i=1}^{K_a} \sum_{j=1}^{K_b} \frac{1}{(\sigma_{a,i}^2 + \sigma_{b,j}^2)} \exp\left(-\frac{1}{2} \frac{(\vec{r_{a,i}} - \vec{r_{b,j}})^2}{\sigma_{a,i}^2 + \sigma_{b,j}^2} \right), \tag{2.1}$$

where K_a and K_b are the number of localizations, r_a and r_b the localization coordinates and σ_a and σ_b the isotropic localization uncertainties for particles a and b, respectively. Both normalizations are crucial to reliably compare the similarity of different pairs of particles. Without the normalizing with respect to the number of localizations, particles with a high number of localizations will give a high cost function value. This is unfavorable, since we want the cost function to reflect the degree of similarity between particles, not the number of localizations. The normalization with respect to the uncertainty is necessary to prevent that localizations with a high uncertainty result in a high cost function value. The cost function essentially calculates the overlap between two Gaussian mixtures. Without normalization, the area under the curve is not equal to 1, and the overlap can become very high for large uncertainties.

The pairwise registration of N particles results in an upper triangular matrix of $N(N-1)/2$ cost function values, where a higher value indicates a better match between the particles. The cost function values S are converted to dissimilarity values D by subtracting all of them from the highest value in the matrix:

$$D(a,b) = \max(S) - S(a,b). \tag{2.2}$$

Multidimensional scaling

To cluster the particles based on the dissimilarity matrix, multidimensional scaling (MDS) [29] is used to embed the particles in a multidimensional space. MDS provides a spatial representation of the data by a nonlinear mapping while preserving the pairwise, symmetric dissimilarities between the particles. Essentially, every particle is given a position in space, in such a way, that the Euclidean distances between pairs of particles are approximately equal to their dissimilarity measure. In this way, particles with a high dissimilarity will be placed far apart in the new space, and particles with a low dissimilarity will end up close to each other.

Basically, this is the inverse process to determining distances. Instead of having a spatial embedding and measuring the distances between all pairs of particles, we start with the distances (dissimilarities) and try to find their spatial embedding. We use nonclassical MDS which iteratively updates the MDS coordinates by minimizing the metric stress loss function, defined as:

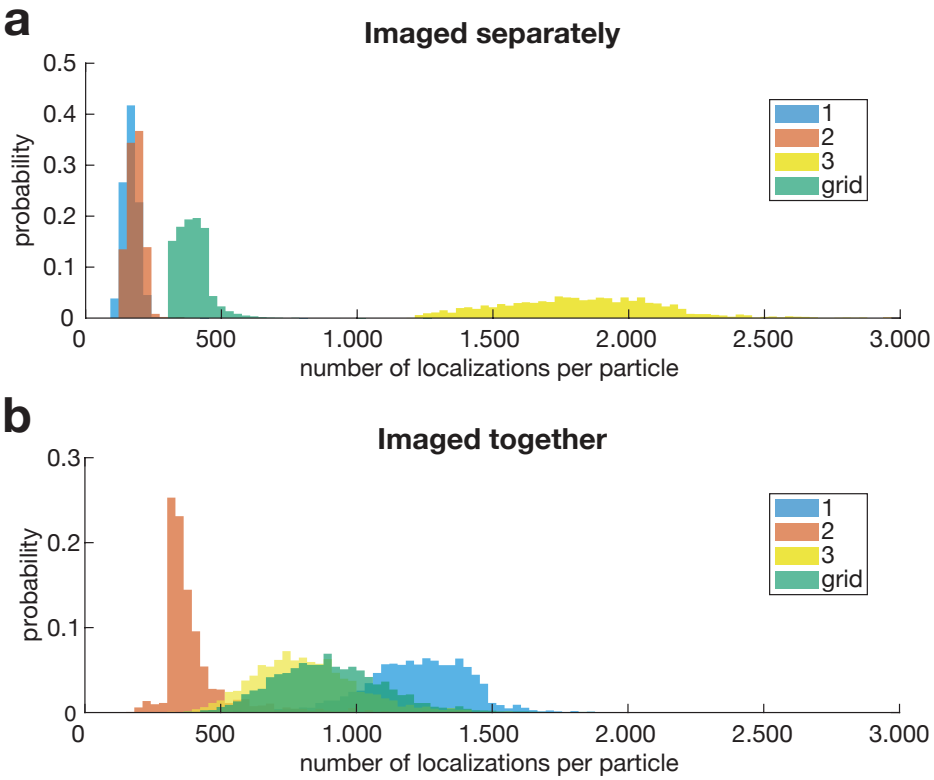
$$\text{Stress} = \left(\frac{\sum_{ij} (d_{ij} - \| x_i - x_j \|)^2}{\sum_{ij} d_{ij}^2} \right)^{1/2} \tag{2.3}$$

where d_{ij} is the dissimilarity between particles i and j, and x_i the MDS coordinates of particle i. This stress loss function assures that the dissimilarities are preserved as the inter-particle distances in the multidimensional space. We choose to embed the particles into 30 dimensions, since we empirically determined that any value between above 15 is sufficient (**Supplementary Fig. 10**) and we take as rule of thumb twice this value.

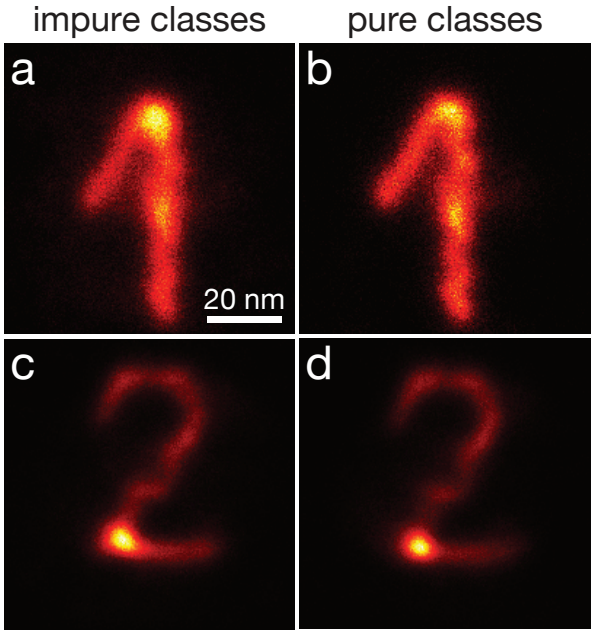
K-means clustering
K-means clustering in the multidimensional scaling space results in K clusters of particles. To prevent finding suboptimal clusters due to the random initialization of the k-means algorithm, the clustering procedure is repeated 1000 times. Using a high number of repeats is especially important in cases where a group containing a small number of particles has to be found. Here, we advise taking the number of repeats equal to the number of total particles, to assure that every particle is on average at least once selected as initial seed for the k-means algorithm. From all initializations, the clustering with the lowest within-cluster sums of points-to-centroid distances is chosen. To determine the value of K, the scatterplot containing the first 3 dimensions of the MDS space can be inspected visually. Reviewing only the first three dimensions is sufficient, since they contain the most important variation within the MDS space and reviewing the order in a higher-dimensional space is visually challenging. If the first dimensions show separated clusters, K can be chosen to match the number of clusters. However, when the variation between the different classes of particles is small, the scatter plot will be a single point cloud, and does not give a clear indication for the choice of K. In this case, as always, selecting the optimal K requires some tweaking. The user can start with an educated guess of K and then vary K to manually inspect what number of clusters is preferred. This can be done quickly as the computationally most time-consuming part of the method is the pairwise registration, which only has to be done once. The fusing of clusters is significantly faster and can, therefore, be repeated for multiple values of K. If the goal is to find a small subgroup with an estimated occurrence of 2%, for example, it is advised to use a high value for K on the order of 40 (as for the mirrored TUD-logo's with 50% DoL, **Fig. 3d**-**f**) or when the variation is a continuous spectrum, a low value for K in the range of 2-3 will suffice (as for the elliptical NPCs, **Fig. 3g**-**l**).

Particle fusion per cluster
Each of the clusters found in the previous step is reconstructed according to the particle fusion pipeline of Heydarian *et al.* [17]. The reconstruction of the grid structure in the digits dataset is done differently. The data contains 12 binding sites at 20 nm spacing in a 4x3 pattern. Due to the regular, symmetric nature of this structure in combination with an unbalanced number of localizations per binding site, reconstruction with the above-mentioned algorithm is suboptimal. Therefore, the localizations are first clustered per binding site. This is done by using the density-based clustering for applications with noise (DBSCAN) technique [8], with an epsilon value of 0.03 pixels and 4 as the minimal number of points per cluster. Subsequently, the above-mentioned particle fusion algorithm is applied using the centers of the found clusters. The mean uncertainty of localizations per cluster is used as the uncertainty for each center. Afterwards, the centers are replaced by all the original localizations of the particles.

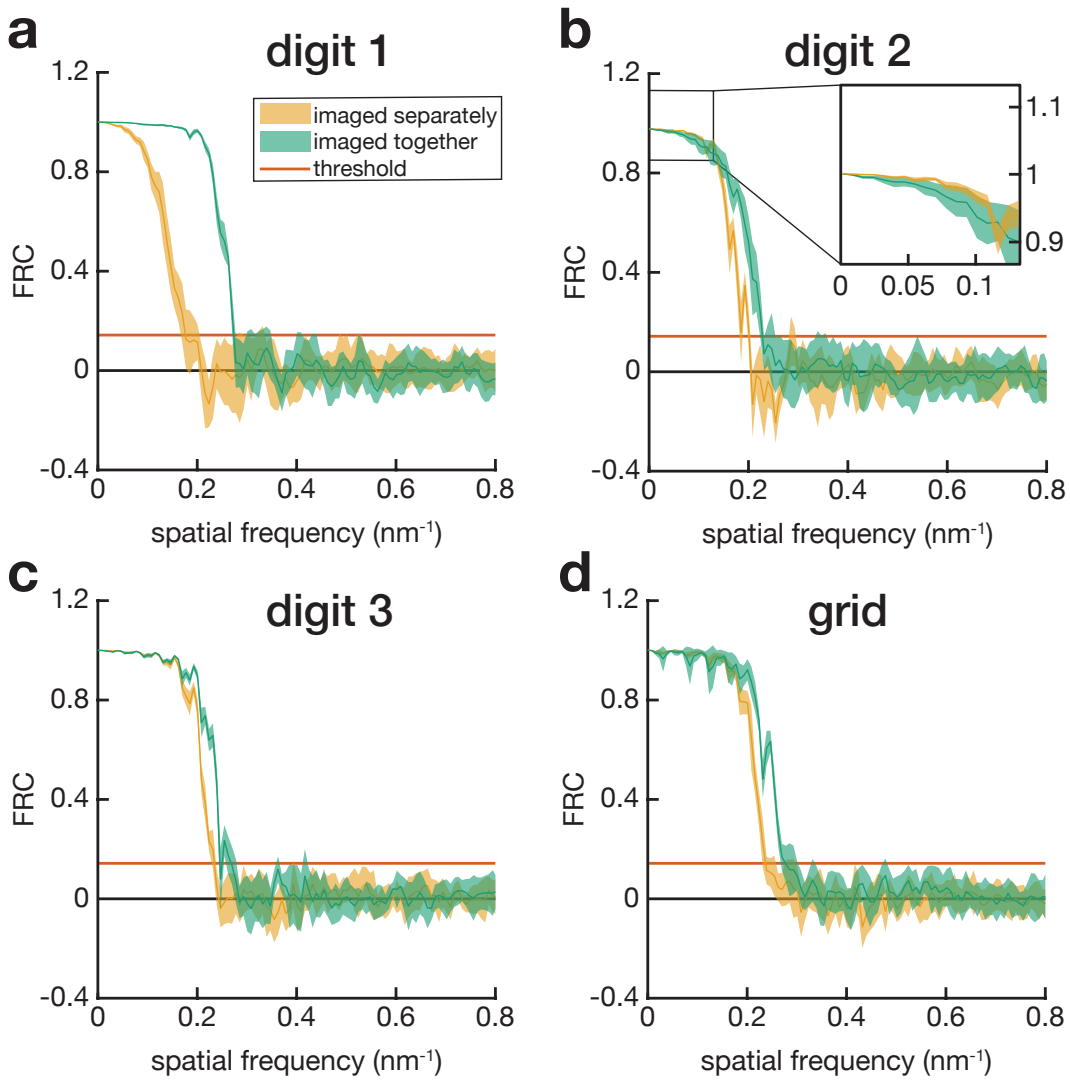
Supplementary Information



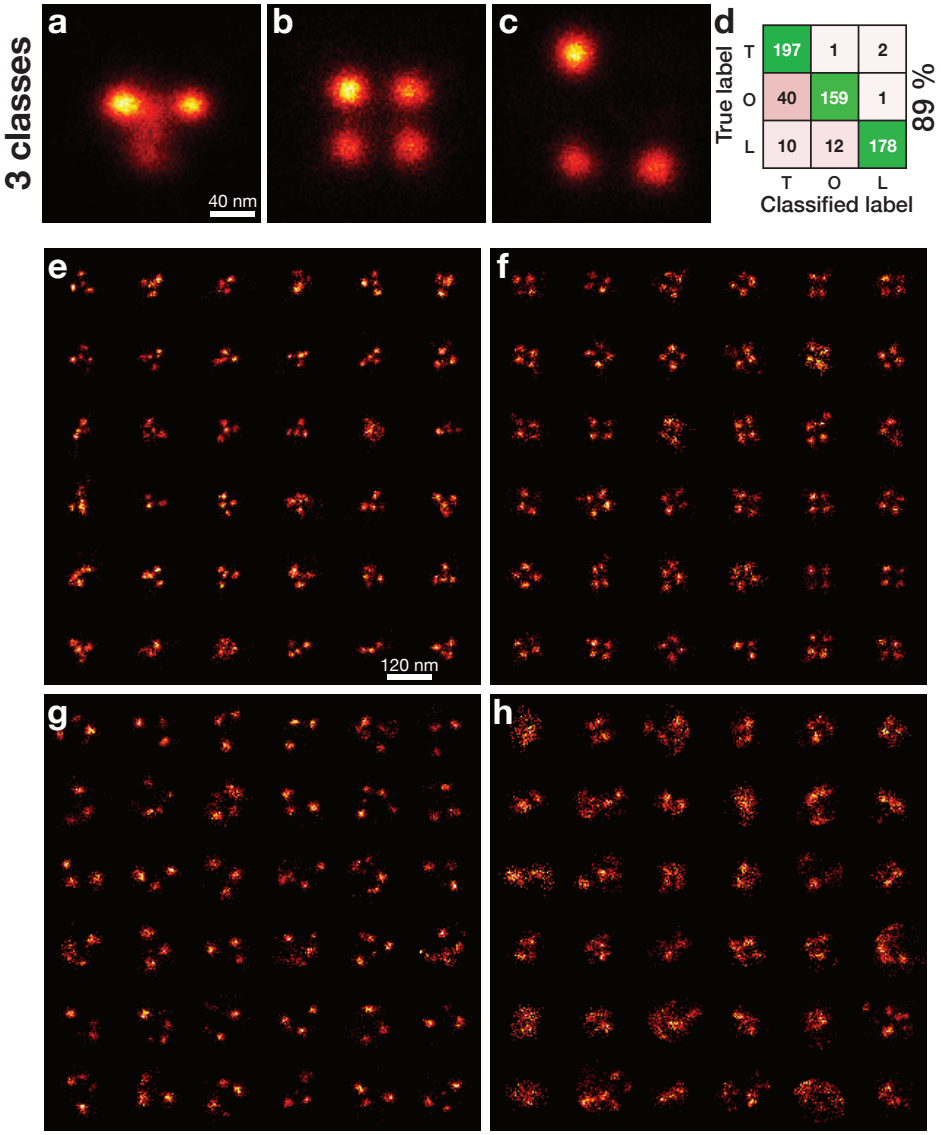
Supplementary Figure 1 | Distributions of number of localizations per particle for digits datasets. **a**, Normalized distributions of number of localizations for digits 1, 2, 3 and grid that are imaged per class in separate experiments. Distributions contain 4155, 4943, 2541 and 7696 particles, respectively. **b**, Normalized distributions for digits 1, 2, 3 and grid that are imaged in one field-of-view. Distributions contain 2832, 1328, 3292 and 3653 particles, respectively. All distributions are individually normalized to have unit sum probability. We see that in **a**, digit 3 contains significantly more localizations per particle than the other structures and in **b**, digit 1 contains the most localizations. Due to this imbalance, the digit 3 and digit 1, respectively, are mostly visible in the fusion results of all particles (**Fig. 2b** and **2h**), since they are the "brightest" class per dataset.



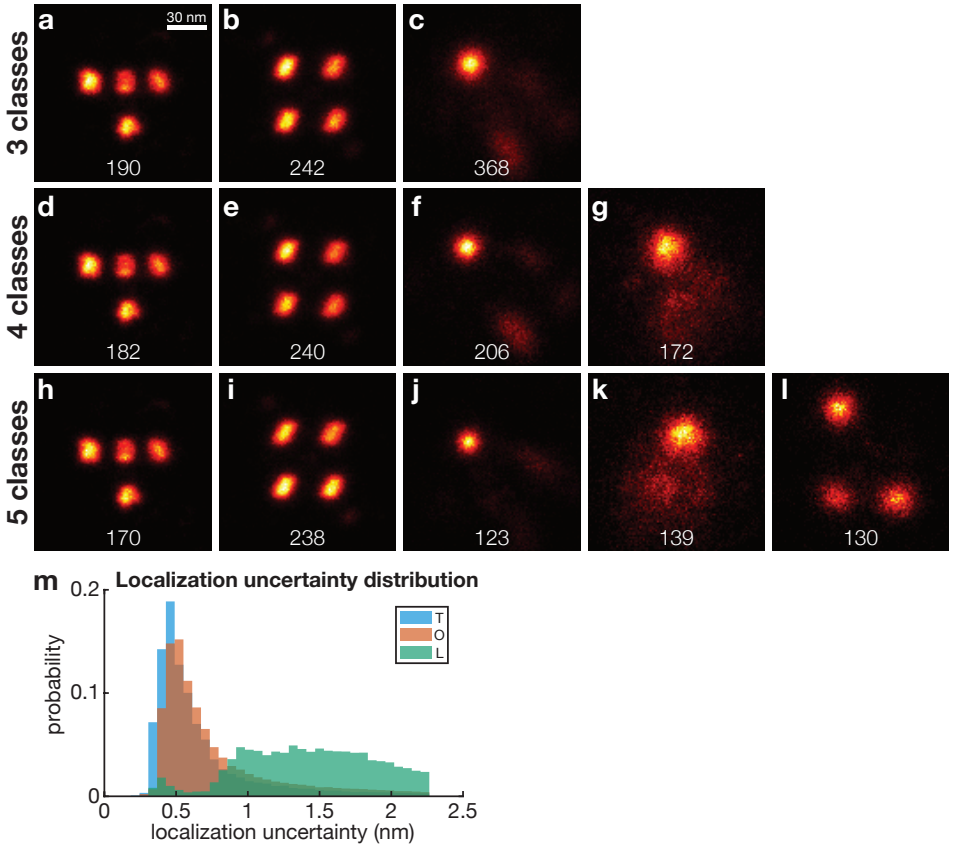
Supplementary Figure 2 | Reconstruction of pure classes. **a**, Class 1 resulting from classification of 5,000 images of the digits dataset, which are imaged separately. Same image as **Fig. 2c** in the main text. As can be seen in the confusion matrix (**Fig. 2g**), this class contains 1236x digit 1, 59x digit 2, 68x digit 3 and 11x grid. **b**, Particle fusion result of only the 1236x digit 1 particles of **a**. **c**, Class 2 resulting from classification of 5,000 images of the digits dataset, which are imaged together. Same image as **Fig. 2j** in the main text. As can be seen in the confusion matrix (**Fig. 2m**), this class contains 37x digit 1, 1201x digit 2, 31x digit 3 and 40x grid. **d**, Particle fusion result of only the 1201x digit 2 particles of **c**. Scale bar of **a** applies to all.
From these particle fusion results, we can conclude that the lower quality of the reconstructions of **a** and **c** (compare to **Fig. 2d**,**i** of the main text) is because the data itself is of lower quality, not due to suboptimal classification performance.



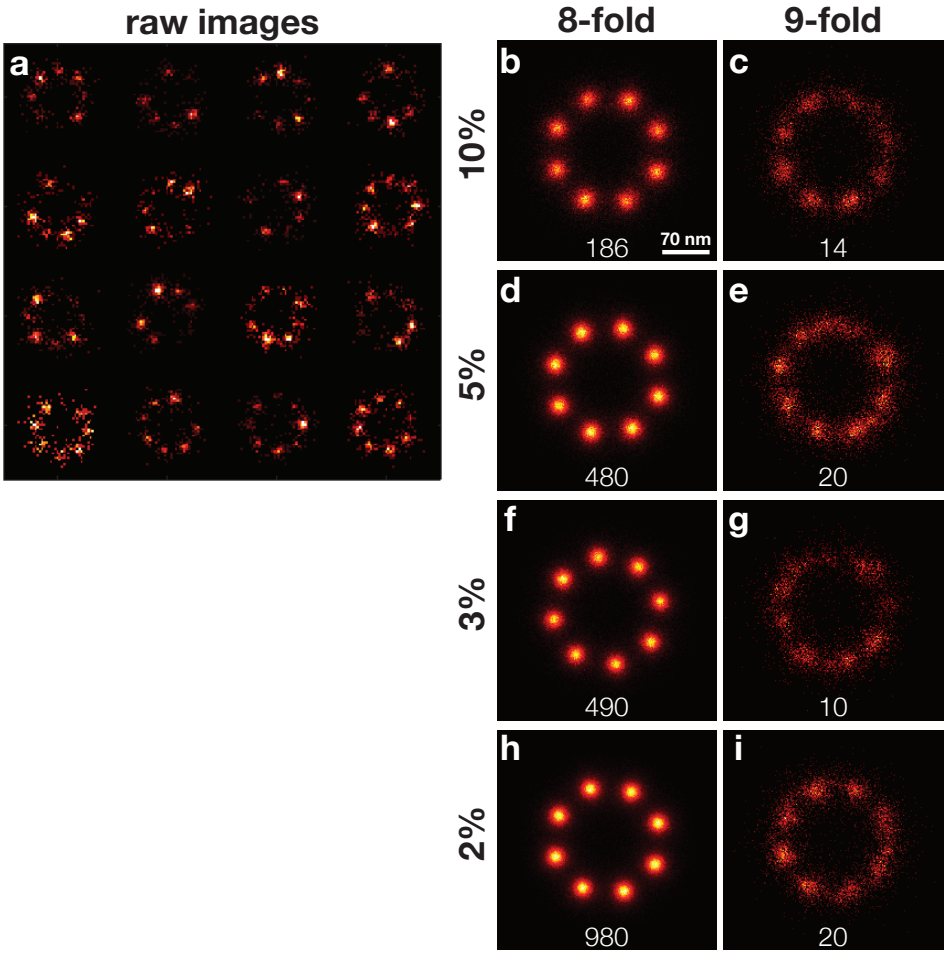
Supplementary Figure 3 | Fourier ring correlation measurements for the digits reconstructions. a, The images classified as digit 1 (**Fig. 2c** and **2i**) are randomly split into two equal groups, reconstructed, and FRC curves are calculated. Solid line is the mean FRC curve for 10 random splits per class and the shaded area represents one standard deviation. The FRC image resolution (crossing with 1/7 threshold) is 5.62 ± 0.39 nm for 'imaged separately' and 3.69 ± 0.02 nm for 'imaged together', defined as the average crossing of 10 random splits with one standard deviation. The significant difference in resolution is visible in the class images (**Fig. 2c** and **2i**). **b**, Same as in **a**, but for digit 2. The FRC image resolution is 5.10 ± 0.28 nm for 'imaged separately' and 4.40 ± 0.19 nm for 'imaged together'. Even though the intersection of the FRC curve with the 1/7 threshold suggests a better resolution for the 'imaged together' experiment, the images (**Fig. 2d** and **2j**) suggest otherwise. The fact that for the low frequencies (see inset), the 'imaged separately' has a higher correlation, explains why the class image for 'image separately' shows a visually better digit 2. **c**, Same as in **a**, but for digit 3. The FRC image resolution is 4.27 ± 0.08 nm for 'imaged separately' and 3.98 ± 0.22 nm for 'imaged together'. **d**, Same as in **a**, but for the 3x4 grid structure. The FRC image resolution is 4.19 ± 0.13 nm for 'imaged separately' and 3.59 ± 0.15 nm for 'imaged together'. Legend of **a** applies to all.



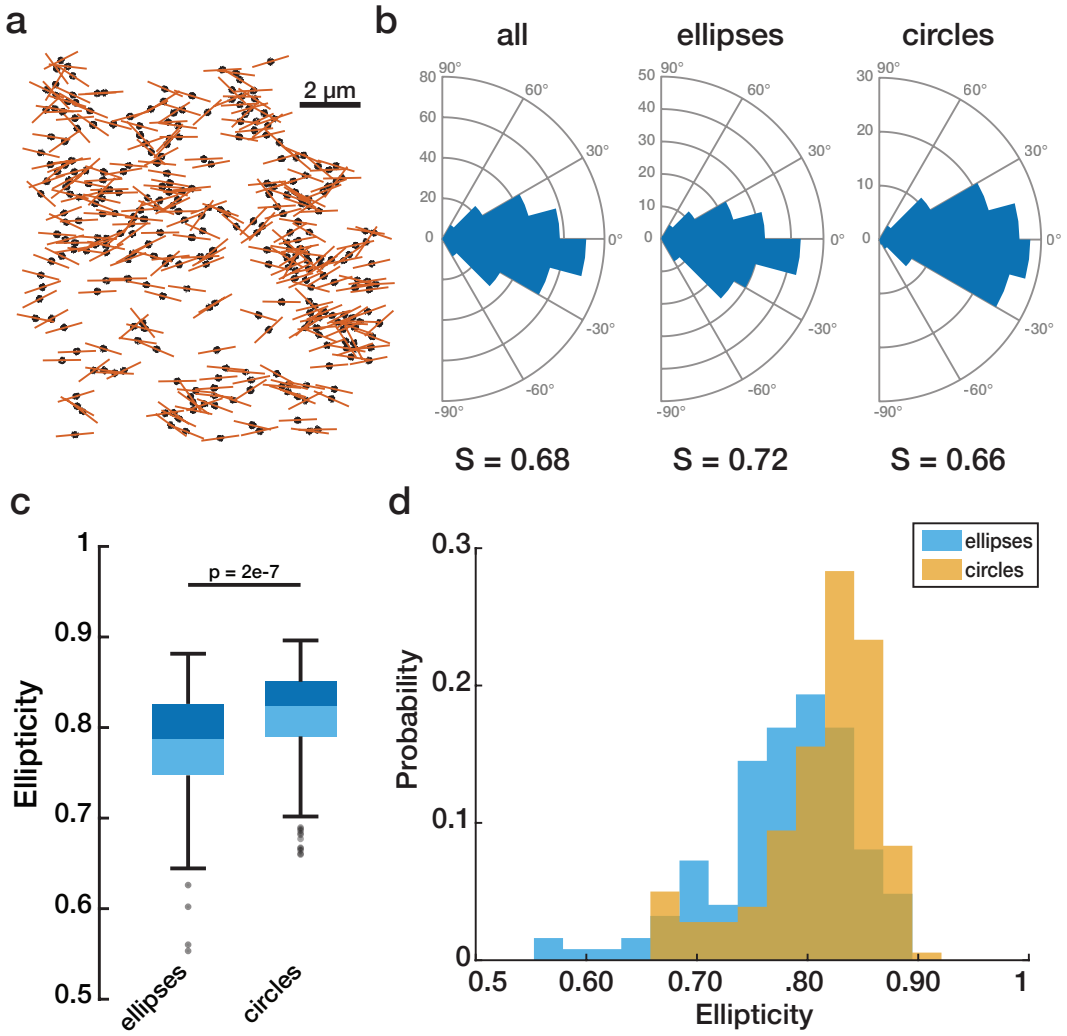
Supplementary Figure 4 | Justification for using four classes to capture the misfolds. **a**-**c**, Classification result for 600 particles, 200 per class, of the letter dataset with separate imaging (**Fig. 2o**) are classified into three classes, containing 247, 172 and 181 particles per class, respectively. The three structures (the letters T, O and L) are clearly visible. However, the letter T is of lower quality than with classification into four classes (**Fig. 2p**). The reason is that the dataset contains misfolded structures that mixed into the other classes (as seen in the confusion matrix **d**). When classified into four classes, the misfolded images are captured in the fourth class. **d**, Confusion matrix for classification into three classes. Due to misclassification of misfolded images, the classification performance of 89% is lower than the 97.3% when a class is added to capture the misfolds (**Fig. 2t**). **e**-**h**, Panels show 36 representative particles per class, when the dataset is classified into four classes (**Fig. 2p-s**). It is clearly visible that **e**-**g** contain the letters T, O and L, where h contains the misfolded images. Scale bar of **a** applies to **b**,**c** and scale bar of **e** applies to **f**-**h**.



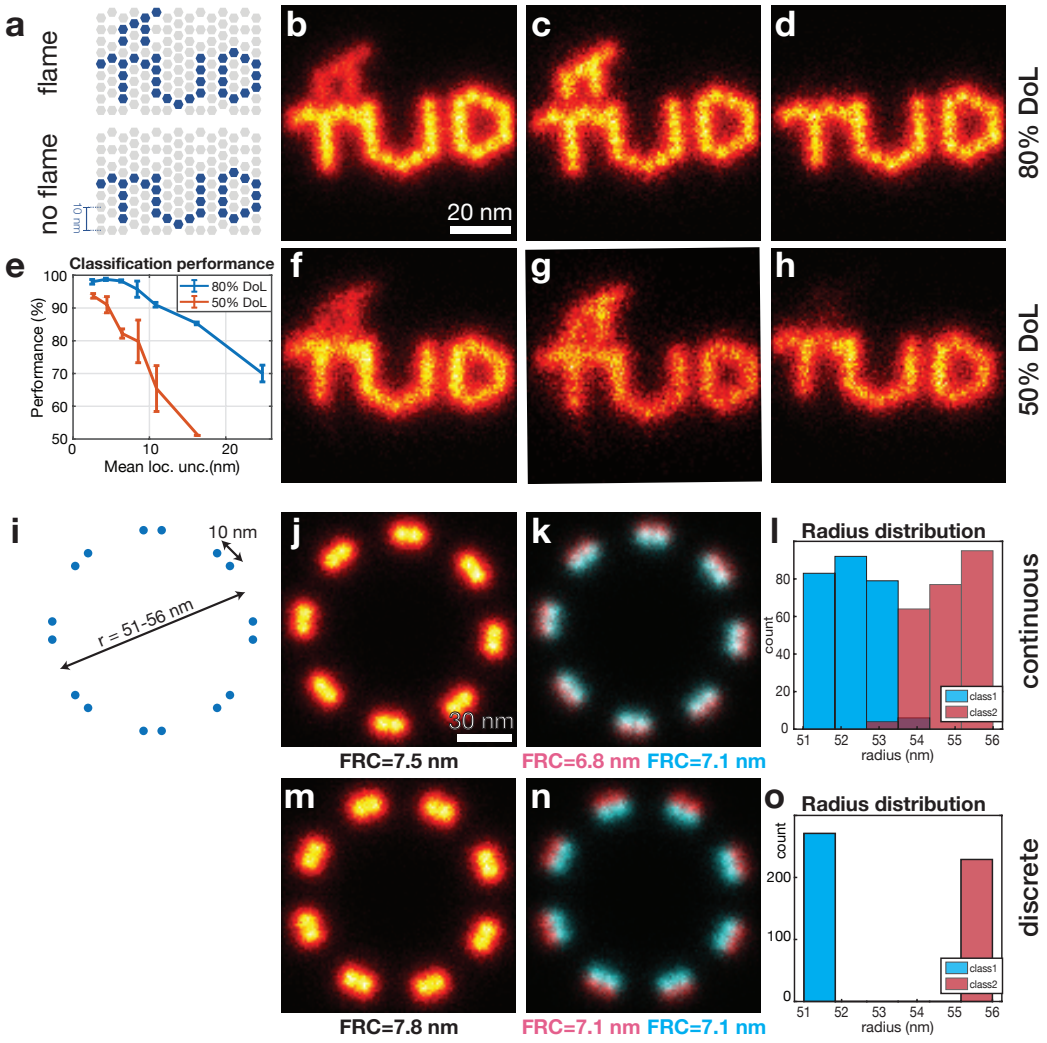
Supplementary Figure 5 | Classification into 5 classes needed to capture the letter L. Classification of 800 particles of the letter dataset, which are imaged together (**Fig. 2u**) into: **a**-**c**, three classes, **d**-**g**, four classes, and **h**-**l**, five classes. The numbers below each panel indicate the number of particles contained in that class. The bottom row, **h**-**l**, contains the same five images as **Fig. 2v**-**z** in the main text, but is shown here again for convenience. **m**, Distribution of localization uncertainties for the T, O and L particles of classes **h**, **i** and **l**, respectively. Scale bar of **a** applies to **b**-**l**.
It is clear that classification into five classes is necessary in order to capture the letter L. The reason is that the L particles have a bigger spread in localization uncertainty. This, in combination with the simple design of having only three clusters of binding sites, makes that the L particles have a low quality and resemble misfolded and unrecognizable particles.



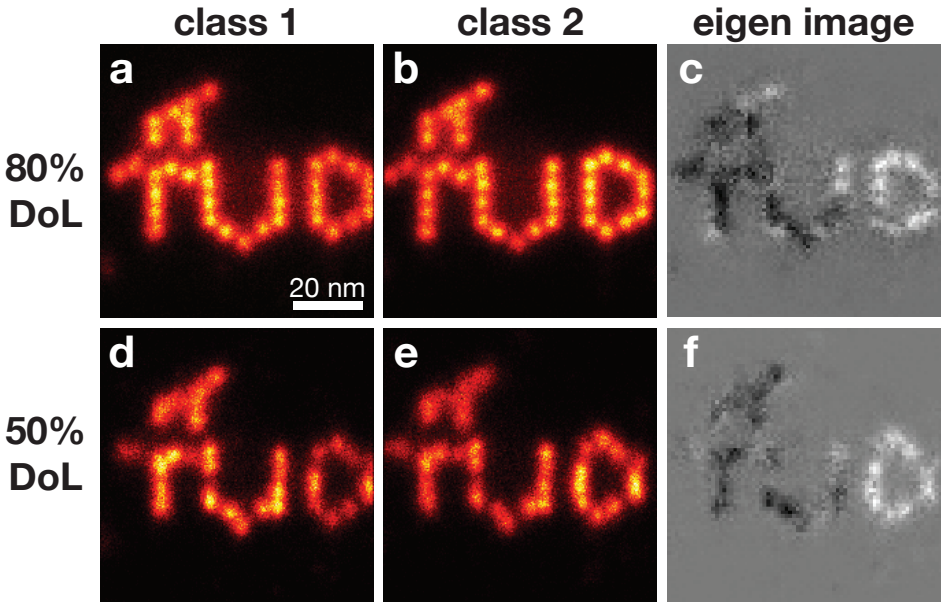
Supplementary Figure 6 | Detecting rare classes of 9-fold simulated nuclear pores. **a**, 16 representative examples of simulated 8-fold symmetric rings with a diameter of 160 nm, a mean localization uncertainty of 8 nm and full labeling. The rings are simulated including a bleaching rate of 0.025 frame-1 as if they are imaged with STORM, and therefore have different numbers or localizations per binding site. We performed classification on four datasets containing 10, 5, 3 and 2% 9-fold symmetric rings of a total number of particles of 200, 500, 500 and 1000. The classification separates the 9-fold symmetric particles (**c**, **e**, **g**, **i**) from the 8-fold symmetric particles (**b**, **d**, **f**, **h**). Numbers below each class indicate the number of particles contained in that class, where for lower rates of 9-fold symmetric rings, larger datasets are required in order to classify them. For classification, the multidimensional scaling space is clustered with k-means in 5, 8, 15 and 20 clusters, respectively, followed by the eigen image method to group them into two classes. The shown class results (**b**-**i**) are reconstructed with implying the 8- and 9-fold symmetry during bootstrapping [17] for visual improvement of the class results. However, the prior symmetry knowledge is only used for the reconstruction, not for the classification itself. Scale bar of **b** applies to **c**-**i**.



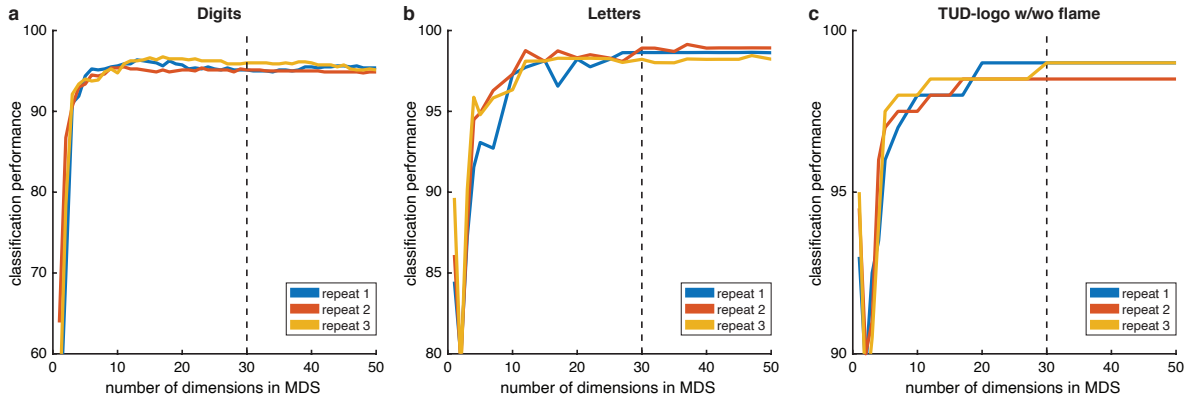
Supplementary Figure 7 | Ellipticity of NPC integral membrane protein gp210. a, Field-of-view (FOV) showing 304 particles [25], with the orientation of the major axis of fitted ellipses for every particle shown in orange. **b**, Polar histograms for the orientation of the major axes of the elliptical fits of **a**. First histogram contains all 304 particles. Second and third histogram contain the particles for the elliptical and circular class, with 180 and 124 particles, respectively. The particles are classified into four classes, of which two resulted in ellipses and two in circles, grouped here per shape. **c**, Boxplots of the ellipticity values for the elliptical and circular classes. Kolmogorov-Smirnov statistic shows that the distributions are significantly different with a p-value of $2.28 \cdot 10^{-7}$. **d**, Same distributions as in **c**, but shown as histograms.



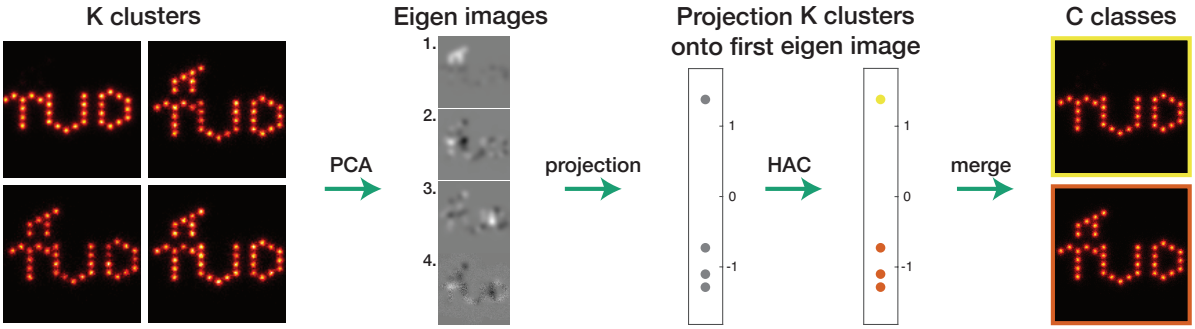
Supplementary Figure 8 | Classification of simulated data. **a**, Templates of two DNA-origami structures used for simulating the particles, TUD-logos with and without flame. **b**, Fusion result of 200 particles (100 with and 100 without flame), 80% DoL, mean localization uncertainty of 4.3 nm and simulated with PAINT. **c**-**d**, Classification result of **b**, containing 98 and 102 particles, respectively. **e**, Classification performance over mean localization accuracy for different labeling densities (200 particles per experiment, equally divided over the two classes). Error bars represent one standard deviation between 2 independent datasets. **f**-**h**, Same as **b**-**d**, but with 50% DoL. **g** contains 78 particles and **h** 122. **i**, Model used to simulate the NPC particles. Ring with 8 doublets where the two emitters per doublet are 10 nm apart. **j**, Fusion result of 500 particles, simulated with a uniformly distributed radius in the range 51-56 nm, mean localization uncertainty of 6.3 nm, 70% DoL and simulated with PAINT. **k**, Two-color overlay of the classification result of j, containing 260 particles in class 1 (cyan) and 240 particles in class 2 (red). The overlay of red and cyan displays white. **l**, Radius distribution per class. Radii are ground-truth radii used in simulating the particles. **m**-**o**, Same as **j**-**l**, but with two discrete radii of 51 and 56 nm. For the average and the individual classes, the FRC values are mentioned below the figures. The mentioned values are the mean 1/7-crossing resolutions of 50 random splits. Scale bar in **b** applies to **c**, **d**, **f**-**h**. Scale bar in **j** applies to **k**, **m**-**n**.



Supplementary Figure 9 | Control classification on one class: DNA-origami data of TUD-logo. **a**,**b**, Classification of 383 experimental DNA-PAINT acquisitions of a DNA-origami TUD-logo with 80% DoL into two classes resulting in 177 and 206 particles per class, respectively. We see that the classification selects on particles with either a bright (**a**) left- or (**b**) right-side of the structure. **c**, Eigen image calculated for **a** and **b**, which is essentially the difference image. Here we clearly see that the letter D and a part of the letter U are bright in **b** and the other part of the logo is bright in **a**. **d**-**f**, same as **a**-**c**, but for 442 particles with 50% DoL, resulting in 220 and 222 particles per class. Scale bar of **a** applies to **b**-**f**.
From this we conclude that the classification selects on the most prominent variation in the dataset. Since there is no variation in the structure here (one class), the algorithm selects on the active/inactive binding sites resulting in a left-right imbalance.



Supplementary Figure 10 | Optimal number of dimensions in multidimensional scaling. **a**, Classification performance for different numbers of dimensions in MDS for 800 particles (200 per class) of the digits dataset which is imaged separately. **b**, Classification performance for different numbers of dimensions in MDS for 600 particles (200 per class) of the letters dataset which is imaged separately. **c**, Classification performance for different numbers of dimensions in MDS for 200 particles (100 per class) of the simulated TUD-logos with a mean localization uncertainty of 4.3 nm and 80% DoL. Half of the particles does not have the flame above the first letter. Three colors in all plots represent independent repeats on different datasets.
We can conclude that the optimal classification performance is reached around 15 dimensions. We take as a rule of thumb twice this value and therefore embed the particles into 30 dimensions in all reported experiments in this paper.



Supplementary Figure 11 | Optional further classification for datasets with small subgroups of structurally different particles. After k-means clustering the multidimensional scaling space into K clusters (**Fig. 1**), principle component analysis (PCA) on the K reconstructions gives K eigen images. All clusters are projected onto the first eigen image and the resulting weights are grouped into C classes by hierarchical agglomerative clustering (HAC). The grouped clusters are merged into C classes, with $C < K$.

Supplementary Note 1 | Eigen image approach

When classifying small subgroups of structurally different particles, for example low abundant mirrored DNA-origamis or 9-fold symmetric NPCs, the multidimensional scaling space has to be clustered into many clusters (high value of K, see **Methods**). Here, the number of obtained K clusters is higher than the desired C classes. The clusters can optionally be grouped further using the eigen image method, which exists of four steps: (1) calculation of the eigen images using principle component analysis (PCA); (2) projection of the K clusters onto the first eigen image; (3) hierarchical agglomerative clustering (HAC) on the obtained weights; and (4) merging of the grouped clusters (**Supplementary Fig. 11**).

The K reconstructed clusters are optionally grouped further using the eigen image method with the aim to find $C<K$ classes. First, we need to align the K clusters with respect to each other, which is done by using the same particle fusion pipeline as explained previously [17]. Instead of fusing the particles, only the final transformation parameters are applied to align the different clusters. These clusters are converted to pixelated images by binning the localizations in an $N \times N$ grid, with $N=400$. This number of pixels results in a pixel size of 0.2 nm for the DNA origami experiments and 0.7 nm for the nuclear pore experiments. As a rule of thumb, the pixel size should be about $\frac{1}{4}$ of the localization uncertainty in the data [31]. A larger pixel size would result in blurring of the images and thereby loss of structural details, whereas a smaller pixel size will make the images too noisy. Thereafter, the images are normalized to have zero mean and a 2-norm of one. The K images are reshaped into column vectors and concatenated into a matrix X of size $N^2 \times K$, where every column represents an image. The eigen images are computed by applying singular-value decomposition (SVD) to the covariance matrix $M = XX^T$. SVD is a generalized version of eigen decomposition that works for non-diagonalizable and even non-square matrices. The decomposition of XX^T results in:

$$XX^T\vec{u} = s_u\vec{v}, \tag{2.4}$$

where $\vec{u}$ and $\vec{v}$ are the left- and right-singular vectors, respectively, with associated singular value s_u. Since the matrix XX^T is normal (it commutes with its conjugate transpose), the left- and right-singular vectors are identical, and the SVD algorithm is effectively the same as eigen decomposition. The resulting singular vectors, $\vec{u}$, represent the eigen images and the associated singular value, s_u, indicates how much variation of the data is explained by its respective singular vector. Instead of computing the SVD on the immense covariance matrix M (size $N^2 \times N^2$), applying SVD on X^TX (size $K^2 \times K^2$) is computationally less expensive:

$$X^TX\vec{a} = s_a\vec{a}. \tag{2.5}$$

After left-multiplying both sides of the equation with X:

$$XX^TX\vec{a} = s_aX\vec{a}, \tag{2.6}$$

we see this resembles the decomposition of XX^T as shown before and the sought-after singular vector, $\vec{u}$, can be calculated as $\vec{u} = X\vec{a}$. By reshaping and normalizing the singular vectors, the eigen images of size $N \times N$ are formed. Since the eigen images are sorted based on their singular values, the first one represents most of the variation between the different images. All K images are projected onto the first eigen image and the resulting weights are clustered into C classes by hierarchical agglomerative clustering with an average-linkage criterion [42]. The localizations of the K clusters are merged per class to form the final C output classes.

3

Extended method development, applications and considerations

The manuscript in the previous chapter discussed the developed classification method in detail and showed the results obtained on multiple experimental and simulated datasets. The four sections of this chapter contain the explanation for choices made in the manuscript, explored alternative methods and further improvements and applications of the classification procedure.

In the first section, we theoretically derive the normalization of the Bhattacharya cost function and show on different datasets that the normalization is beneficial for classification. In the second section, hierarchical agglomerative clustering is explored as an alternative to the multidimensional scaling step in the classification pipeline. The third section contains different ideas to further improve the obtained classification result and the fourth section will show the implementation and validation of the developed classification method on 3D data.

3.1. Normalization of the Bhattacharya cost function

In the original particle fusion pipeline of Heydarian *et al.* [17], the registration of particles is based on the Bhattacharya cost function (Eq. 3.1). This function calculates the cost between all combinations of localizations of two particles and sums them. Arguably, merit-function would have been a more suiting name, since a high similarity between particles gives a high cost function value.

$$\mathrm{D(a,b)} = \sum_{i=1}^{K_a} \sum_{j=1}^{K_b} \exp\left(-\frac{(\vec{r_{a,i}} - \vec{r_{b,j}})^2}{\sigma_{a,i}^2 + \sigma_{b,j}^2} \right) \tag{3.1}$$

The cost between two localizations is high when the localizations are close to each other, i.e. $(\vec{r_{a,i}} - \vec{r_{b,j}})^2$ is small. The squared Euclidean distance between two localizations is divided by the localization uncertainties σ_a and σ_b. In this way, two localizations that are further apart can still result in a relatively high value, if the corresponding localization uncertainties are also bigger.

In the existing particle fusion pipeline, the cost function is used to determine the best transformation between two particles. The Gaussian-mixture-model registration approach is initialized with multiple rotation angles and finds the best rigid transformation for each initial angle. Subsequently, the Bhattacharya cost function (Eq. 3.1) is calculated for each of them and the transformation resulting in the highest cost function value is chosen. In this way, the cost function values are only compared between different transformations of the same two particles.

In the classification pipeline, the pairwise dissimilarities between all pairs of particles are the input for the multidimensional scaling. The obtained Bhattacharya cost function values per registration, essentially similarity values since a high cost function value indicates a good similarity, are converted to dissimilarity values as explained in the Manuscript. Here, the cost function values between different pairs of particles are compared to each other, in contrast to the particle fusion pipeline. Since the cost function is sensitive to the number of localizations and the uncertainties, comparing different pairs of particles is unfair when particles contain different numbers of localizations or have a different distribution of localization uncertainties. Both problems can be solved by normalization of the cost function. In this section we will derive the 2D (as used in the Manuscript) and 3D normalized Bhattacharya cost function and investigate the improvement over the standard cost function.

Normalization with respect to the number of localizations

In order to compare cost function values for different pairs of particles, we have to take into account the different numbers of localizations per particle. We take an example where we have three particles, A, B and C, with 50, 10 and 100 localizations, respectively. The cost function value for pair $A-B$ depends on 500 combinations of localizations, whereas pair $A-C$ has 5000 combinations. Even if particles B and C have the same structure and quality, but only differ in the number of localizations per binding site, the cost function value for $A-C$ will be approximately 10 times higher than the value for $A-B$.

Particles with the same structure should give the same cost function value, regardless of the number of localizations, otherwise the multidimensional scaling will embed the particles based on the number of localizations and not the conformational similarity. To reliably compare different combinations of particles, we have to normalize the cost function with respect to the number of localizations, which results in:

$$\mathrm{D(a,b)} = \frac{1}{K_a K_b} \sum_{i=1}^{K_a} \sum_{j=1}^{K_b} \exp\left(-\frac{(\vec{r_{a,i}} - \vec{r_{b,j}})^2}{\sigma_{a,i}^2 + \sigma_{b,j}^2} \right), \tag{3.2}$$

where K_a and K_b are the number of localizations in particles a and b, respectively.

Normalization with respect to the localization uncertainties

The second issue with the standard cost function arises for localizations with a large localization uncertainty. In this case, a large value for σ results in a large denominator for the fraction within the exponent, which subsequently causes a high output for the cost function. This means that localizations with a large σ will always result in a high cost function value, disregarding how far the two localizations are separated from each other (Fig. 3.1).

This issue can be solved by normalizing the cost function with respect to the localization uncertainties. In this way, the cost function resembles the expression for a multivariate normal distribution. Instead of calculating the probability that a point has a certain distance to the mean, given the standard deviation, we calculate the probability that two points are a certain distance apart, given both localization uncertainties. The next sections contain the derivation for the cost function with normalization with respect to the localization uncertainty in both 2D and 3D.

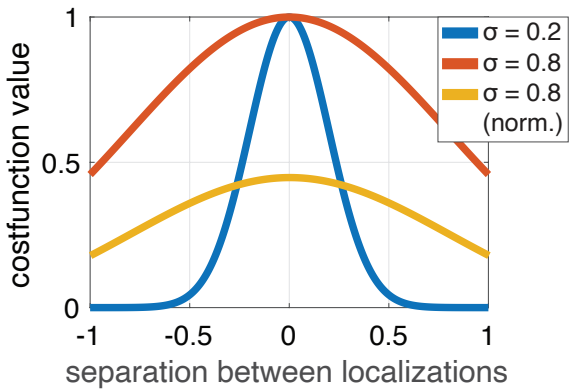


Figure 3.1 | Cost function with and without normalization with respect to localization uncertainty. Without normalization, the cost functions for $\sigma = 0.2$ (blue) and $\sigma = 0.8$ (red) have a different width but the same height. This means that localizations with a big uncertainty result in a high value for the cost function, disregarding the distance between the localizations. With normalization (yellow), the height of the function scales with the width, so localizations with a small uncertainty give a higher cost function value.

Derivation of 2D cost function

The Bhattacharya cost function is based on a multivariate normal distribution, of which the definition in D dimensions is given by:

$$\frac{1}{(2\pi)^{D/2}\sqrt{\det(\Omega)}}\exp\left(-\tfrac{1}{2}(\vec{r}-\vec{\mu})^T\Omega^{-1}(\vec{r}-\vec{\mu})\right). \tag{3.3}$$

This expression is the basis for the 2D Bhattacharya cost function between two localizations a and b, which is defined as:

$$\frac{1}{2\pi\sqrt{\det(\Omega)}}\exp\left(-\tfrac{1}{2}(\vec{r_a}-\vec{r_b})^T\Omega^{-1}(\vec{r_a}-\vec{r_b})\right), \tag{3.4}$$

with $\vec{r}$ the 2D position-vector of a localization and $\Omega = \Sigma_a + \Sigma_b$, the sum of both covariance matrices. The covariance matrix Σ_a is correctly rotated according to the orientation of the corresponding particle and is defined as:

$$\Sigma_a = \mathrm{R}\Sigma_{a,0}\mathrm{R}^T = \mathrm{R}\begin{bmatrix}\sigma_{ax}^2 & 0\\ 0 & \sigma_{ay}^2\end{bmatrix}\mathrm{R}^T, \tag{3.5}$$

where $\Sigma_{a,0}$ is the covariance matrix in the standard orientation with σ_{ax} and σ_{ay} the localization uncertainties of localization a in respectively the x- and y-direction. In the case of isotropic localization uncertainty, σ_{ax} and σ_{ay} are equal and we will denote them as σ_a. Hereby, we can simplify the expression for the covariance matrix:

$$\Sigma_a = \mathrm{R}\begin{bmatrix}\sigma_a^2 & 0\\ 0 & \sigma_a^2\end{bmatrix}\mathrm{R}^T = \sigma_a^2\mathrm{R}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\mathrm{R}^T = \sigma_a^2\mathrm{R}\mathrm{R}^T = \sigma_a^2\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}. \tag{3.6}$$

We use this expression to find Ω^{-1} and the determinant of Ω:

$$\begin{aligned}\Omega^{-1} &= (\Sigma_a+\Sigma_b)^{-1}\\ &= \begin{bmatrix}\sigma_a^2+\sigma_b^2 & 0\\ 0 & \sigma_a^2+\sigma_b^2\end{bmatrix}^{-1}\\ &= \begin{bmatrix}\frac{1}{\sigma_a^2+\sigma_b^2} & 0\\ 0 & \frac{1}{\sigma_a^2+\sigma_b^2}\end{bmatrix}\\ \det(\Omega) &= (\sigma_a^2+\sigma_b^2)^2,\end{aligned} \tag{3.7}$$

which gives us the expression of the normalized cost function:

$$\frac{1}{2\pi(\sigma_a^2+\sigma_b^2)}\exp\left(-\frac{1}{2}\frac{(\vec{r_a}-\vec{r_b})^2}{\sigma_a^2+\sigma_b^2}\right). \tag{3.8}$$

This expression only gives us the cost for a pair of localizations. To calculate the cost between two particles a and b, with respectively K_a and K_b localizations, we sum over all possible combinations of localizations and normalize with respect to the number of localizations:

$$\text{Cost(a,b)} = \frac{1}{K_aK_b}\sum_{i=1}^{K_a}\sum_{j=1}^{K_b}\frac{1}{2\pi(\sigma_{a,i}^2+\sigma_{b,j}^2)}\exp\left(-\frac{1}{2}\frac{(\vec{r_{a,i}}-M(\vec{r_{b,j}}))^2}{\sigma_{a,i}^2+\sigma_{b,j}^2}\right), \tag{3.9}$$

where M is the rigid transformation used to register particle b onto particle a. Note that the isotropic uncertainties are not affected by the transformation. The factor 2π in the denominator of the prefactor only scales all cost function values with a constant and is therefore omitted in the computation.

Derivation of 3D cost function

The derivation for the 3D normalized cost function follows the same reasoning as above. The difference is that in 3D the uncertainties are not isotropic, because the localization uncertainty in the z-direction, σ_z, is bigger than in the x- and y-direction, σ_{xy}. As a result, the uncertainties have to be rotated along with the localizations. We start with the same equation for the multivariate normal distribution (Eq. 3.3). Now, we have to correctly rotate the uncertainties of particle b using the rotation matrix R:

$$\Omega = \Sigma_a + R\Sigma_b R^T. \tag{3.10}$$

Using this expression for Ω and summing over all combinations of localizations, we arrive at the expression for the normalized 3D cost function:

$$\text{Cost(a,b)} = \frac{1}{K_a K_b} \sum_{i=1}^{K_a} \sum_{j=1}^{K_b} \frac{\exp\left(-\frac{1}{2}(\vec{r_{a,i}} - M(\vec{r_{b,j}}))^T (\Sigma_{a,i} + R\Sigma_{b,j}R^T)^{-1} (\vec{r_{a,i}} - M(\vec{r_{b,j}}))\right)}{2\pi\sqrt{\det(\Sigma_{a,i} + R\Sigma_{b,j}R^T)}}. \tag{3.11}$$

Note that in 2D that uncertainties are isotropic, which means that only one scalar value per localization is sufficient in the cost function calculation (σ). In 3D, the localization uncertainty is described by two values, σ_{xy} and σ_z. However, these uncertainties are rotated along with the orientation of the respective particle, which means that we need the full 3x3 covariance matrix Σ to describe the uncertainties.

Improvement due to normalized cost function

Normalizing the cost function with respect to the number of localizations and the localization uncertainty significantly improves the ability to separate different structures. Here, we will show the improvement by means of a 2D and 3D example.

As an example for 2D data, we simulated 500 TUD-logos of which 25 were mirrored as if they were imaged upside-down with DNA-PAINT on a total internal reflection fluorescence (TIRF) microscopy setup. The simulations were performed with a density-of-labeling (DoL) of 50% and a mean localization uncertainty of 2.5 nm. The MDS embeddings of the dissimilarity matrix for the cost function with and without the two normalizations show clearly that the normalizations improve the clustering of similar structures (Fig. 3.2). For the case with normalizations, the class of mirrored particles (blue dots) is significantly better separable from the other particles, than without normalizations.

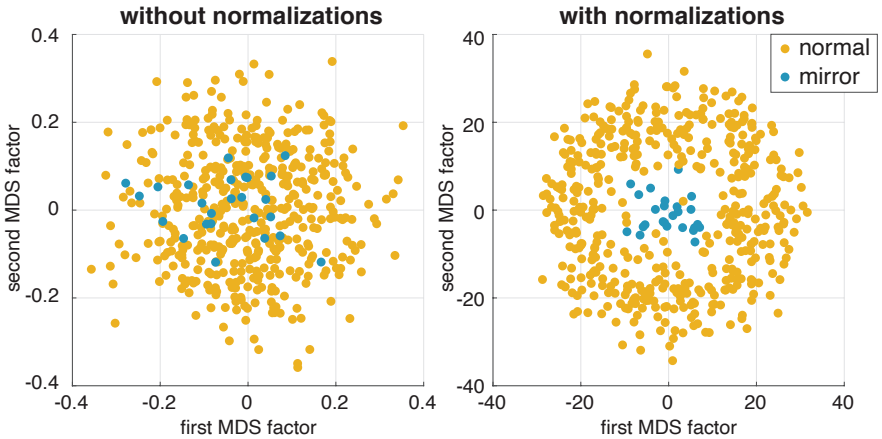


Figure 3.2 | Effect of normalizing the cost function on MDS embedding (2D). The scatterplots show the first two dimensions of the MDS embedding of 500 TUD-logos, of which 25 are mirrored. The particles are simulated as if they were imaged with DNA-PAINT, with 50% DoL and a localization uncertainty of 2.5 nn. (Left) using the standard cost function, (Right) using the cost function normalized with respect to the number of localization and the localization uncertainty. It is clear that with the normalized cost function the 25 mirrored particles (blue dots) are better separable from the 475 normal images (yellow dots), than without the normalization. This will allow for better classification.

As an example of 3D data, we simulated 200 3D nuclear pore complexes with labeled NUP107 (nucleoporin 107). The model for NUP107 is a double ring of 16 binding sites per ring, grouped in sets of 2. Both rings have a diameter of 96 nm and they are 58 nm separated in height [18] (see Figure 3.16 in Section 3.4 for the model). We simulated two classes of NUP107, one class of 100 particles with a 13° shift between the nuclear and cytoplasmic ring and one class of 100 particles without a shift. The particles were simulated as if they were imaged with DNA-PAINT on a TIRF setup. The simulations were performed with 100% DoL, a mean lateral localization uncertainty of 3.5 nm and axial uncertainty of 10.5 nm (3 times the lateral uncertainty). The MDS embeddings of the dissimilarity matrix for the cost function with and without the two normalizations show that the normalization improves the clustering of the similar structure (Fig. 3.3). Even though the two classes are perfectly separated for both cases, the separation is more clear and the classes more compact with normalizations.

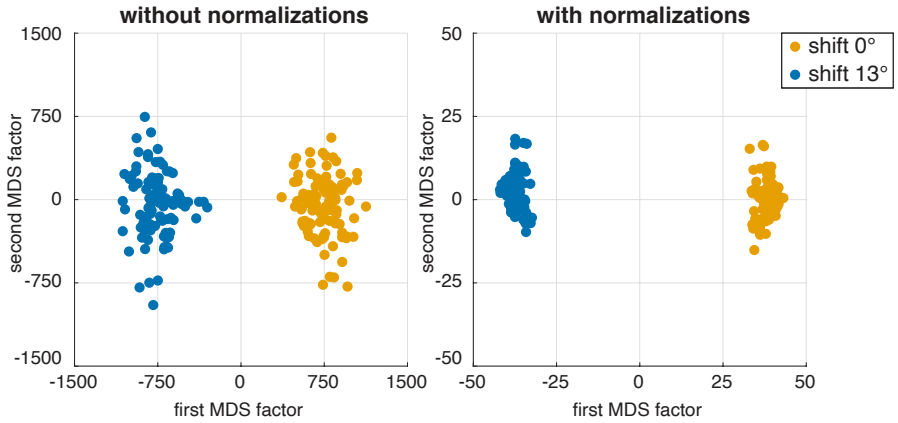


Figure 3.3 | Effect of normalizing the cost function on MDS embedding (3D). The scatterplots show the first two dimensions of the MDS embedding of 200 NUP107 structures (left) without, and (right) with the normalized cost function with respect to the number of localizations and the localization uncertainty. Of the 200 particles, 100 have a 13° shift between the rings and 100 have no shift. The particles are simulated as if they were imaged with DNA-PAINT, with 100% DoL and a lateral localization uncertainty of 3.5 nn. It is clear that with the normalized cost function the two classes are more compact and better separated, than without the normalization. This will allow for more robust classification.

These are two example datasets where the normalized cost function gives a clear improvement. Testing the normalized cost function on multiple other datasets (data not shown) always gave an equally good or improved classification result. The question is whether the increase in performance is caused by the normalization with respect to the number of localizations, to the localization uncertainty, or a combination of both. The answer to this question is dataset-specific.

The normalization with respect to the number of localizations is important when the particles have different numbers of localizations. These differences can be caused by natural variations in blinking statistics, varying labeling densities between particles and the imaging method that is used. In the case of STORM imaging, the fluorophores undergo bleaching, which results in an uneven distribution of localization over the binding sites. This is not a problem for PAINT imaging, where the fluorophores are constantly replaced due to binding and unbinding to the target locations. It is necessary to normalize with respect to the number of localization to have a fair comparison between combinations of particles. In the situations where different classes have different numbers of binding sites, there is an inherently different expected number of localizations due to the structure. Here, the classification would be better without normalization with respect to the number of localizations, because then the algorithm selects on these numbers instead of on conformational differences.

The normalization with respect to the localization uncertainty, next to being theoretically correct, only makes a difference when there is a significant variation in the uncertainties. For experimental data, the localizations are filtered on localization uncertainty, so the uncertainties usually only differ a factor of two. This means that all uncertainties are approximately in the same range and the normalization with respect to uncertainty will not give a significant difference. For simulated data, on the contrary, the localization uncertainties are drawn from a geometric distribution. Here, the spread in uncertainties can be up to a factor of 10, even when the localization with high uncertainties are filtered out.

Due to the presence of a wider variation in uncertainties, the normalization with respect to uncertainties will make the cost function significantly better, otherwise, the localizations with high uncertainties get effectively more weight in the registration of two particles. We conclude that the normalization with respect to localization uncertainty is theoretically correct and will give the largest improvement when the range of uncertainties is wider. In general, both normalizations are necessary to make the classification fair for all kind of particles and types of structural variation.

3.2. Hierarchical agglomerative clustering approach as alternative to MDS

The most important step in the classification pipeline is clustering based on the dissimilarities between all the particles. In general, there are two techniques available to cluster based on pairwise dissimilarities: hierarchical clustering and clustering based on a spatial embedding of the particles. We have implemented and optimized both strategies and tested them on multiple datasets. The performance of the second method, i.e. clustering based on a spatial embedding, performs significantly better than hierarchical clustering. Therefore, the second method, hereafter referred to as the multidimensional scaling (MDS) approach, is further optimized and used in all subsequent experiments (see Manuscript).

In this section, we will explain the implementation and results of hierarchical clustering. Albeit it has not been tested extensively on all available datasets, we will show its capabilities and compare the performance with the multidimensional scaling approach.

The HAC algorithm

Hierarchical clustering is a method that can cluster objects based on a built hierarchy. The hierarchy is visualized in a dendrogram, which is a tree-like graph that indicates the distances between all clusters. Dendrograms are commonly used in phylogeny, to study the evolutionary history among organisms.

Types of hierarchical clustering

There are two types of hierarchical clustering: agglomerative and divisive. In hierarchical agglomerative clustering (HAC), every object is initially seen as an individual cluster. The dendrogram is constructed by iteratively merging the two closest clusters until all objects are part of the same cluster. In hierarchical divisive clustering, all objects start as part of one cluster, which is iteratively split until all clusters contain one object. When using an exhaustive search to find the best merge or split, respectively, the computational complexity for agglomerative is $\mathcal{O}(N^3)$ and for divisive $\mathcal{O}(2^N)$, for N objects. The procedure of divisive clustering can be accelerated by using smart alternatives to determine the split or only computing the first few layers of the hierarchy. We use the agglomerative approach since we want to compute the complete hierarchy and we are dealing with hundreds of particles, and for large N, the computational complexity for agglomerative clustering is lower.

Linkage criterion

It is not trivial to define the distance between two clusters since only the pairwise distances between the objects are available. The are multiple criteria that can be used as a metric for the distance between two clusters (Fig. 3.4), the most common variants are the single-, average- and complete-linkage criterion. With the single-linkage criterion, the distance between two clusters is defined as the shortest distance between any two objects in these two clusters. In each step of building the dendrogram, the two clusters are merged that contain the closest pair of objects that are not yet belonging to the same cluster. This strategy tends to find elongated clusters, since adding a point to an existing cluster only depends on the shortest distance to that cluster, not on the distances to all the other points. The complete-linkage criterion defines the distance between two clusters as the longest distance between any two objects in these two clusters. This strategy tends to find spherical, compact clusters, since adding a point to an existing cluster depends on the distance to the furthest point within that cluster. A third criterion is average-linkage, where the distance between two clusters is defined as the average distance between all pairs of objects. This is the most robust criterion, since the distance between to clusters depends on all pairwise distances among objects [42].

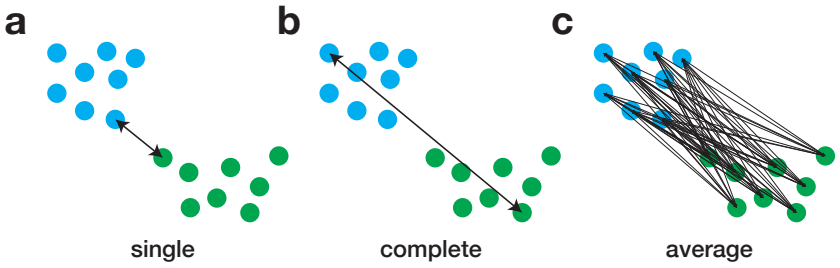


Figure 3.4 | Different linkage criteria. a, In single-linkage, the distance between two clusters (depicted in green and blue) is defined as the shortest distance between any two objects in these two clusters. **b**, In complete-linkage, the distance between two clusters is defined as the longest disntance between any two objects in these two clusters. **c**, In average-linkage, the distance is defined as the average distance among all pairs of objects.

Constructing the dendrogram

Here we will illustrate with a simple example how the dendrogram is built, using the single-linkage criterion. We start with an upper-triangular matrix of $N(N-1)/2$ dissimilarity values for N particles. The dissimilarity values result from the Bhattacharya cost function values obtained with all-to-all registration (see Manuscript). The cost function gives similarity values, S(a,b), which are converted to dissimilarity values, D(a,b), by subtracting them from the highest value in the matrix: D(a,b) = max(S)-S(a,b). The dissimilarity values are used as the 'distances' between the particles in the hierarchical agglomerative clustering.

Building the dendrogram starts by considering every particle as a separate cluster. In the first step, we merge the two clusters that have the lowest dissimilarity, in our example, the yellow and green particle (Fig. 3.5a). Merging the yellow and green cluster is visualized in the dendrogram by connecting the two particles, where the branch that connects the particles to their common node (orange) has a length equal to the dissimilarity between the particles, indicated on the x-axis of the dendrogram. After merging the yellow and green particle into one cluster, the dissimilarity matrix is updated accordingly.

To determine the dissimilarities between all particles and the new yellow-green cluster, we use the single-linkage distance criterion. This means that the dissimilarity with the red particle becomes 3, since 3 is the smallest dissimilarity between the red particle and the yellow-green cluster. By determining the new dissimilarities according to this strategy, we obtain an updated dissimilarity matrix reduced by one column and one row (Fig. 3.5b). This procedure is repeated until all particles are members of the same cluster. In our example, this means first the merge between red and blue (Fig. 3.5b), then the merge between yellow-green and red-blue (Fig. 3.5c) and at last the addition of the orange particle (Fig. 3.5d).

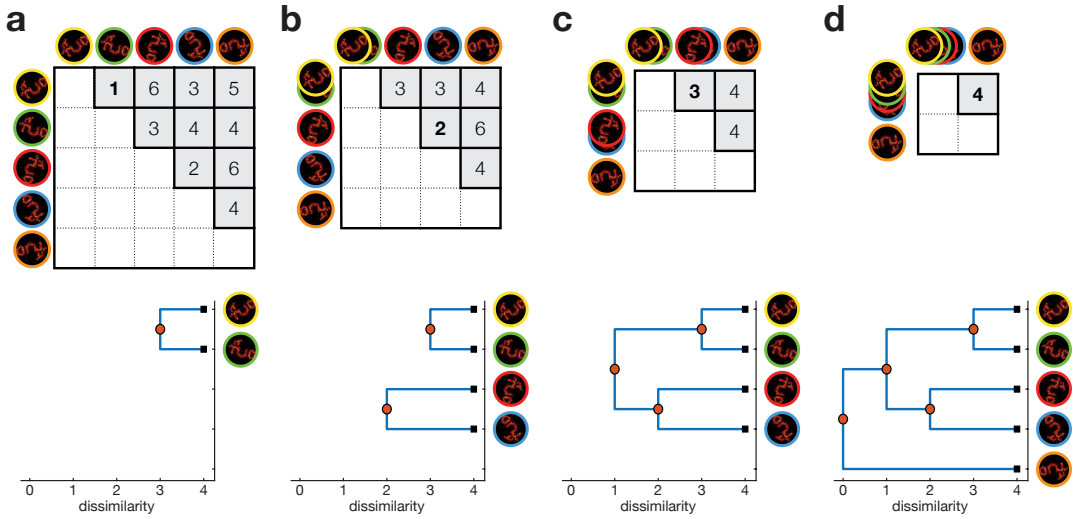


Figure 3.5 | Schematic representation of how the dendrogram is constructed from the dissimilarity matrix. **a**, We start by merging the two particles that have the lowest dissimilarity, in this example the yellow and green particle. In the dendrogram (**a**,bottom) the two particles are connected with a branch that has the length of their dissimilarity. After merging yellow and green, the matrix is updated using the single-linkage criterion. In the next steps: (**b**) the red and blue particles are merged, (**c**) the yellow-green and red-blue clusters are merged and (**d**) the orange particle is added, until all particles are part of the same cluster.

Clustering

The constructed dendrogram is used for clustering of the particles. Since the dendrogram represents the hierarchy for the particles, the clustering is performed by defining a threshold and pruning of the dendrogram. The particles of each pruned sub-dendrogram belong to one cluster (Fig. 3.6), and can subsequently be reconstructed per cluster.

The bottleneck of this clustering approach is determining the threshold for pruning the dendrogram. If we prune the dendrogram in too few clusters, the clusters are large and will contain particles that may belong to different classes. Whereas pruning the dendrogram in too many clusters will create small clusters. In this way, each class will be separated into many clusters, which makes that the clusters only contain a few particles and therefore do not reconstruct properly. Pruning the dendrogram in too many clusters also has the danger of overfitting. Here, the user finds classes that are not present in the data but are just an artifact of reconstructing a few particles together.

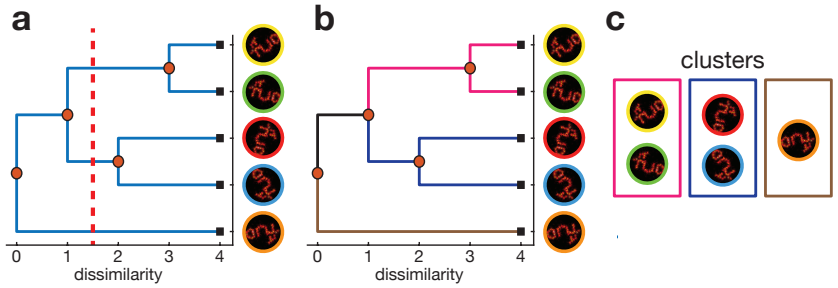


Figure 3.6 | Clustering the dendrogram. a, First, the user determines the desired number of clusters (3 in this example) or a threshold for the dissimilarity values (red dotted line). **b**, This threshold is used to prune the dendrogram into subgroups, indicated in pink, dark-blue and brown branches. **c**, The particles belonging to the same subgroup end-up in the same cluster.

Comparison of the HAC to the multidimensional scaling approach

The HAC approach for clustering the particles is tested on multiple datasets. In the method development process, it became clear that the multidimensional scaling (MDS) approach (see Manuscript) performs significantly better than HAC, especially for the more challenging datasets. Here, we will show three examples where MDS outperforms HAC.

Example 1: Manually flipped experimental TUD-logos (80% DoL)

For the first example, we used the experimental dataset of DNA-origami TUD-logos imaged with DNA-PAINT (data previously described [17]). The dataset contained 380 particles in which we created two classes by manually flipping 50% of the particles, so 190 particles have the normal orientation and 190 particles are mirrored. The dissimilarity values were obtained with all-to-all registration and used by HAC and MDS to cluster the particles. The dendrograms created by HAC, for both the average- and complete-linkage criterion (Fig. 3.7a,b) show that the particles are correctly clustered per class (single-linkage dendrogram gave no result, not shown). In the case of average-linkage, the two classes are clearly separated, whereas for complete-linkage each class is split into two groups. Although the dendrogram can group the classes, there is no trivial number of clusters which we can use for pruning the dendrogram into separate classes. For average-linkage, we needed to prune the dendrogram into 8 clusters in order to separate the classes (Fig. 3.7a), because 6 particles form separate branches in the bottom-part of the dendrogram. For complete-linkage, pruning the tree into 4 clusters perfectly separated the two classes. The MDS approach, on the other hand, gives a clear separation between the two classes and allows for perfect clustering with k-means (Fig. 3.7c).

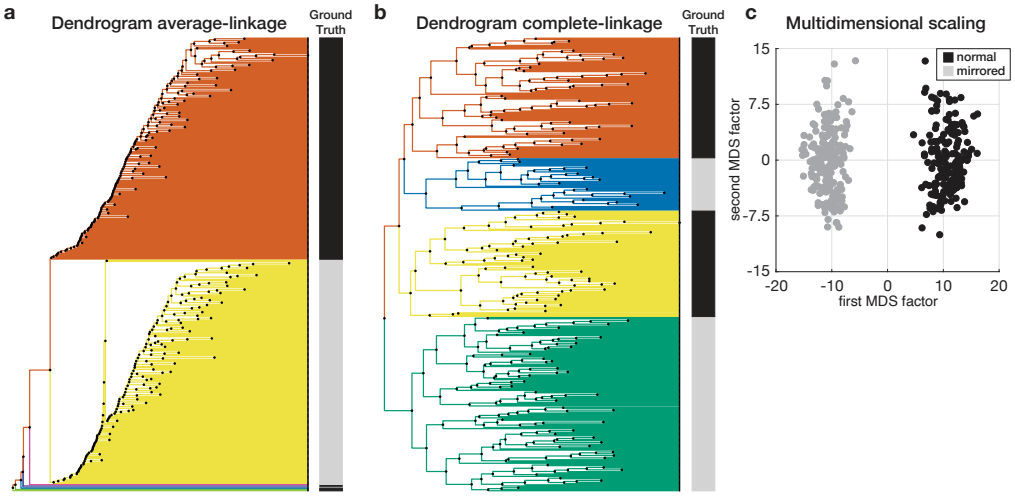


Figure 3.7 | Comparison between HAC and MDS on 80% DoL experimental dataset with 50% flipped TUD-logos. a, HAC dendrogram with average-linkage criterion. Colors of the branches indicate pruning result into 8 clusters, note the 6 clusters in the bottom-part of the dendrogram consisting of only one particle. The gray-scale bar on right-side of the dendrogram indicates the ground truth label for the particles, for legend see **c**. **b**, Same as **a**, but for the complete-linkage criterion and pruning into 4 clusters. **c**, The scatterplot shows the first two dimensions of the MDS embedding. The colors indicate the ground truth class labels.

For this dataset, HAC performs equally well as MDS. However, for HAC we have to choose whether to use average- or complete-linkage and in how many clusters to prune the dendrogram. These choices are not trivial since the dendrograms do not always show a clear grouping in their architectures. Only the ground-truth labels next to the dendrograms give a hint on how to prune them, which are of course not available on real unlabeled data. Another problem is that the architecture of the average-linkage dendrogram is dependent on all pairwise distances between the particles. Since these distances contain outliers due to misregistrations, some particles are therefore positioned as isolated nodes on the outside of the dendrogram. Subsequent pruning of the dendrogram creates clusters that contain only 1 particle (as for the dendrogram in Fig. 3.7a). Since these single clusters are the result of outliers and not because they belong to a structural class of which only one particle is present, they are discarded for further analysis. In this way, particles are lost, which is not the case for MDS. We can conclude that the MDS approach gives a clearer separation between the classes and relies on fewer parameters.

Example 2: Manually flipped experimental TUD-logos (50% DoL)
The second example contains a similar dataset as example 1, but with 50% labeling density, which renders classification more complex (data previously described [17]). The dataset contained 440 particles in two classes, of which 50% are manually flipped, so 220 particles have the normal orientation and 220 particles are mirrored. The dendrograms created by the HAC approach show some grouping of the classes, but the classes are fragmented into many clusters, which makes proper clustering impossible (Fig. 3.8a,b). On the contrary, the first two dimensions of the MDS approach (Fig. 3.8c) show clear grouping of the two classes. Subsequent k-means clustering of the MDS space results in a clustering efficiency of 95%, which shows that the MDS approach is able to better classify the images than the HAC method.

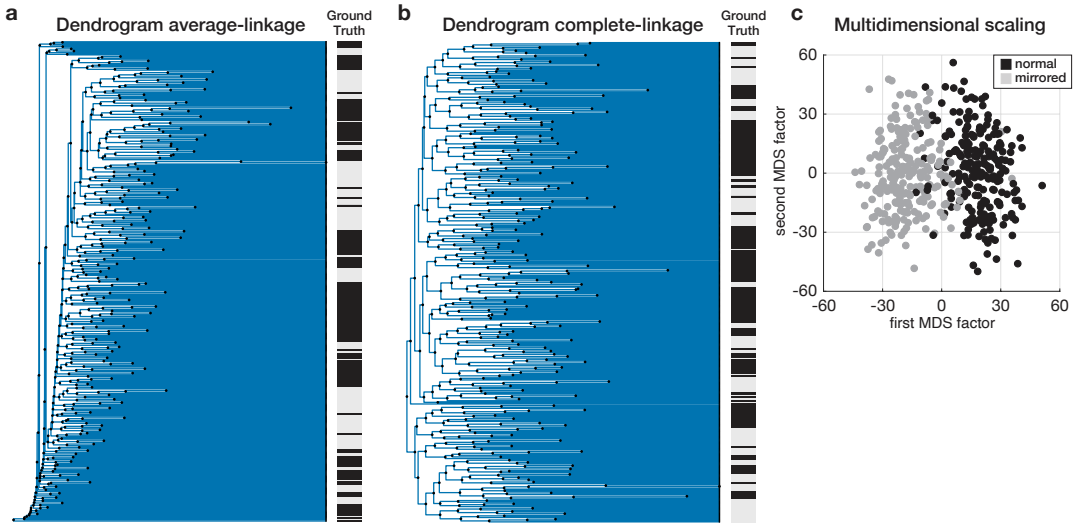


Figure 3.8 | Comparison between HAC and MDS on 50% DoL experimental dataset with 50% flipped TUD-logos. a, HAC dendrogram with average-linkage criterion. The gray-scale bar on right-side of the dendrogram indicates the ground truth label for the particles, for legend see **c**. **b**, Same as **a**, but for the complete-linkage criterion. **c**, The scatterplot shows the first two dimensions of the MDS embedding. The gray-scale colors indicate the ground truth class labels.

Example 3: Experimental four-class nanoTRON dataset
As a third example, we examined the classification performance of HAC and MDS on the nanoTRON dataset (data previously described [1] and shown in Fig. 1 of the Manuscript). The dataset used here contained 200 particles, 50 per class, which were imaged separately per class. The dendrograms created by HAC show clustering of the classes, however, the architecture of the network does not show a clear pruning threshold that will separate all four classes correctly (Fig. 3.9a,b). Contrary to HAC, the first dimensions of the MDS approach show a clear clustering of the particles per class (Fig. 3.9c). With subsequent k-means clustering (not shown), the particles can be classified with a 93% performance.

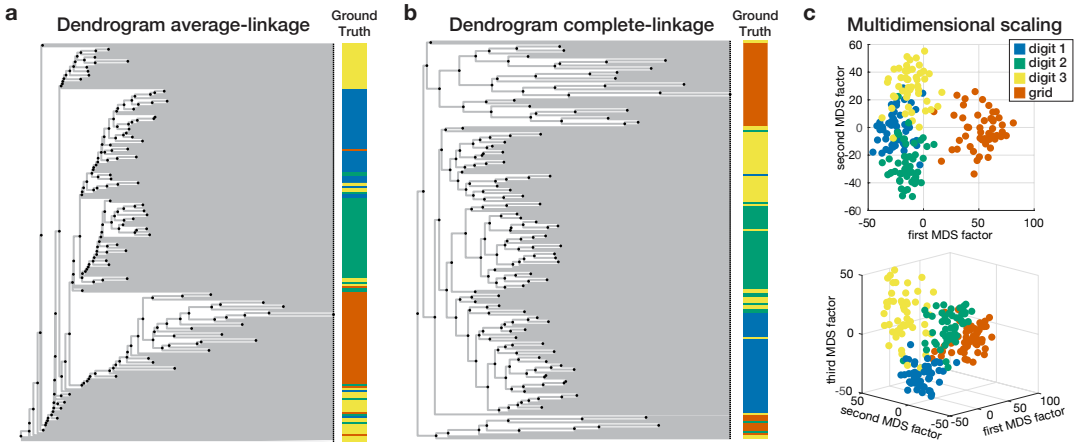


Figure 3.9 | Comparison between HAC and MDS on four-class nanoTRON dataset consisting of 50 particles per class. a, HAC dendrogram with average-linkage criterion. The color-bar on the right-side of the dendrogram indicates the ground truth label for the particles, for legend see **c**. **b**, Same as **a**, but for the complete-linkage criterion. **c**, (top) The scatterplot showing the first two dimensions of the MDS embedding, (bottom) scatterplot showing the first three dimensions of the MDS embedding. Multiple view are shown to get a better idea about the three-dimensional configuration of the particles. The colors indicate the ground truth class labels.

Conclusion

In conclusion, the first example shows that easy-to-classify data can be classified correctly by both HAC and MDS, although HAC does require the delicate choice of the right parameters. The second and third example show that more complex data or data containing multiple classes, can only be correctly classified using the MDS approach and not by HAC. Other advantages of the MDS approach are that there is no choice for the linkage criterion and that no particles are lost in pruning the tree, because of the creation of single-particle clusters.

The MDS approach still requires to choose the number of K clusters in k-means clustering, however, this is usually a low number and is significantly less delicate than the choice of the pruning parameter in the HAC approach. A possible reason for the better performance of the MDS approach could be that the multidimensional embedding of the particles in the MDS approach allows for more flexibility than the hierarchy captured in a dendrogram. In principle, all dissimilarity values from the registration matrix are used and preserved in the multidimensional embedding, whereas the hierarchical clustering can be too dependent on certain particles, since one dissimilarity can have an major effect on the inter-cluster distance. The multidimensional scaling effectively performs dimensionality reduction of the dissimilarity space, removing the noise that results from individual misregistrations.

From these examples, we can conclude that the MDS approach performs significantly better than the HAC approach, which supports the decision to continue only with the MDS approach (as used in the Manuscript). In further sections of this chapter, we will refer to the HAC approach as the dendrogram approach.

3.3. Refining the classification result

Once the classification is performed, all particles are assigned to a class. Afterwards, the class images are formed by performing particle fusion per class of particles. The success of the classification determines the purity of the classes and therewith the quality of the class images, because particles that end up in the wrong class will decrease the quality of the reconstruction. Here, we will discuss different strategies to improve the purity of the classes after the classification is performed. The principle is to either exchange particles between classes to improve the match between the respective particle and the class average, or to remove particles from a class to improve the within-class alignment.

Since the MDS approach (see Manuscript) is performing significantly better than the dendrogram approach, as is shown in the previous section, the need to further refine the classes is less abundant for MDS. For this reason, the refinements of this section are tested and validated on classification results obtained with the dendrogram approach. For future work, however, it would be interesting to see what class refinement can bring to the already satisfying classification results obtained with MDS.

Iterative template matching

One way to refine the classes after classification is the concept of iterative template matching (ITM). In cryo-electron microscopy (cryo-EM), ITM is a common method to do image classification (see multi-reference alignment in the Review, Chapter A). ITM starts with several reference images (templates) to which all images are registered. Every image is assigned to the template that best matches the image, and in this way, the classes are formed. All images belonging to the same class are averaged to form an updated template representing this class. The process of image assignment and template updating is iteratively repeated until convergence. However, initialization of the templates forms a bottleneck. In cryo-EM, hundreds of thousands of images with low SNR are available, so starting with random templates and performing ITM for many iterations will eventually result in the desired classes. In the case of localization microscopy, the particles have a higher SNR and fewer particles are available. Therefore, we will only utilize the concept of ITM to refine the classes after the classification is performed.

Here, we will discuss the strategy where the classes are formed using the hierarchical agglomerative clustering approach (previously described in section 3.2). Particle fusion is performed per class and the resulting reconstructions are used as templates for ITM. In one round of ITM, every particle is registered to both templates and assigned based on the highest Bhattacharya cost function value. In order to reliably compare the registrations of the particle to different templates, we have to take into account three important aspects.

- First of all, the presence of large localization uncertainties hampers the ability of the Bhattacharya cost function to reliably evaluate the similarity between particles. This problem is solved by normalizing the Bhattacharya cost function with respect to the localization uncertainties (see section 3.1) and by filtering the localizations based on the uncertainty. For simulated particles, we decide to discard the 5% localizations with the highest localization uncertainty. The localization uncertainties are drawn from a geometric distribution, where the long tail of the distribution gives rise to unrealistically high uncertainties. Filtering on the 95% lowest uncertainties ensures realistic values, while preserving most of the localizations.

- Secondly, the registration between the particle and the template has to be as precise as possible. In the conventional all-to-all registration step of the particle fusion pipeline, a few misregistrations are accepted, since the Lie-algebraic averaging and subsequent bootstrapping will resolve them. In ITM, every registration has to be correct, since the resulting cost function value determines to which class the particle will be assigned. In order to achieve correct registration, we initialize the GMM-registration with 18 angles (three times more than usual) and use the optimal value for the scale parameter determined by a scale sweep. The number of initialized angles and the scale parameter need to be optimized for every dataset, since different geometries and sizes of the particles will require different parameters.

- The third important aspect is downsampling of the template. The classes consist of many particles, which results in templates that contain up to a million localizations. For computational efficiency, the templates are downsampled to have approximately 5000 localizations. To account for the fact that different structures can have different numbers of fluorophore binding sites, the templates are downsampled taking into account the average numbers of localizations of the particles in that class. Practically, we calculate the average number of localizations of the particles per class and downsample the

templates to have a maximum of 5000 localizations, while maintaining the ratio between the average number of localizations between classes. As an example: if class *A* has on average 175 localizations and class *B* 200, then we downsample their respective templates to 4375 and 5000 localizations. Furthermore, we have to carefully think about how to downsample the templates. In the bootstrapping step of conventional particle fusion [17], weighted downsampling is used based on the local density of localizations. This means that localizations surrounded by many other localizations have a higher probability to be selected in the downsampling process, compared to isolated localizations without neighbours. For ITM, we experience that random downsampling is more reliable since the weighted downsampling enhances bright spots in the reconstruction. With random downsampling, all localizations have the same probability for being selected, which results in uniform downsampling of the templates.

ITM on simulated data

We tested the ITM approach on 500 simulated TUD-logos, of which 50 logos are mirrored. The particles were simulated as if they were imaged with DNA-PAINT, with 50% DoL and a mean localization uncertainty of 2.5 nm. Particle fusion without classification does not reveal the mirrored particles (Fig. 3.10a). Classifying the particles with the dendrogram approach (section 3.2), gives two classes that show the two structures, even though 6 particles are wrongly classified (Fig. 3.10b-d). For the dendrogram method, we used the average-linkage criterion and pruned the dendrogram into 20 clusters, on which the eigen image method (see Manuscript) was applied to obtain two classes. ITM was applied (templates shown in Fig. 3.10e,f) to refine the classes, which resulted in perfect classification (Fig. 3.10g-i). In this case, the two downsampled templates already resemble the two structures, so only one iteration of template matching sufficed to obtain a stable solution and perfect classification.

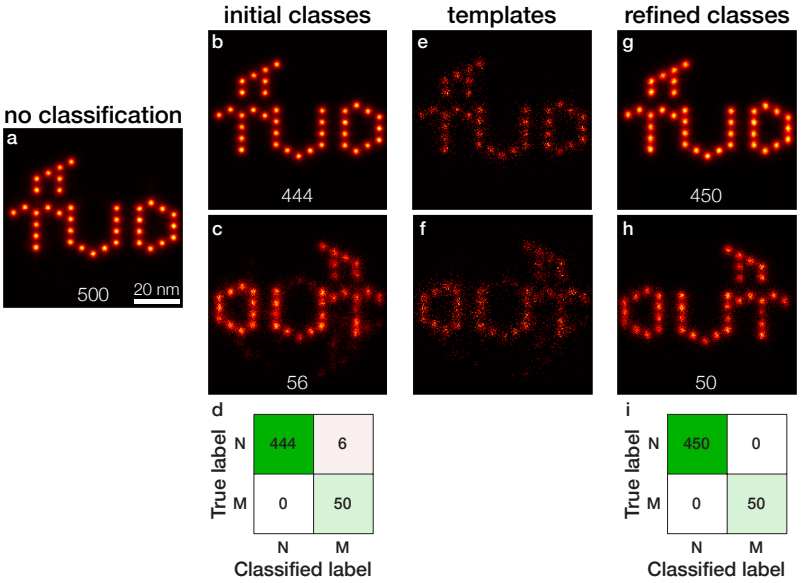


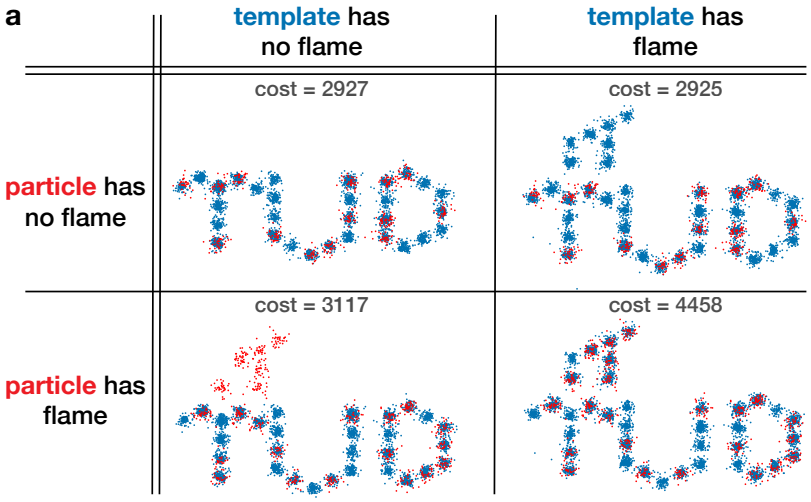
Figure 3.10 | Iterative template matching on 500 simulated TUD-logos containing 50 mirrored particles. **a**, Without classification, the result of particle fusion does not reveal the mirrored particles. **b**,**c**, Classification with the dendrogram approach gives two classes with 444 and 56 particles, respectively. **d**, Confusion matrix for classification result **b** and **c**, indicating the amount of correctly and wrongly classified particles in the normal (N) or mirrored (M) orientation. **e**,**f**, The classes are further refined with the ITM approach. The initial classes are downsampled into two templates, to which all particles are registered and assigned. **g**,**h**, The resulting refined classes containing 450 and 50 particles, respectively. **i**, Confusion matrix for classification result **g** and **h**. Scale bar of **a** applies to **b**, **c**, **e**-**h**.

Partial overlap between classes requires extra analysis

In the above experiment, both structures have the same number of binding sites. Problems occur when the two structures have a different number of binding sites and one structure is a subset of the other structure. We explain this issue with the example of the TUD-logo with and without the flame above the letter T. The logo with the flame consists of 37 binding sites, whereas the logo without the flame only has 30 sites. When a particle without a flame is registered to two templates, one with and one without a flame, the cost function value will be approximately the same for both templates since both contain the letters TUD (top row of Fig. 3.11a).

From the perspective of the particle, it matches equally well with both templates since the flame of the second template does not overlap with the particle and does therefore not result in a different cost function value. A possible solution would be to look at the similarity from the perspective of the template, because one of the templates matches the particle completely, whereas the other does not. However, looking at the similarity from the template-perspective is incorrect when the particle has a flame (bottom row of Fig. 3.11a). In that case, both templates are completely covered by the particle, but the particle itself is only covered by the template with the flame. This results solely in a correct class assignment when looking from the particle-perspective.

The solution to this problem is inspired by the idea of Manders coefficients [28], used in dual-color microscopy to quantify the co-localization of objects. The core principle of this metric is that the co-localization can be quantified from the perspective of both colors, by normalizing either with respect to the total intensity of one or the other color. We use this idea to quantify the similarity, observed from the perspective of the particle or the template, by dividing the cost by either the number of localizations of the particle or the template. Using these two measures is useful since we have observed previously that both perspectives give different results, depending on the structure of the particle. Following this logic, we can construct the tables of Fig. 3.11b,c that indicate the value of the cost function for the two templates (columns) from either the particle- of template-perspective (rows). When the particle does not have a flame (Fig. 3.11b), the cost function values from the particle-perspective will be similar for both templates, but will be different for the template-perspective, and the inverse is true when the particle has a flame (Fig. 3.11c). To determine to which template we have to assign the particle, we take the perspective for which the cost function values differ the most and assign the particle to the template with the highest value.



b

particle has no flame

	template no flame	template flame
particle	~	~
template	high	low

c

particle has flame

	template no flame	template flame
particle	low	high
template	~	~

Figure 3.11 | Manders coefficients for ITM. a, Four different situations that occur in template matching on data with two classes. The particle (red), either having a flame or not, is registered to two templates (blue), one with and one without a flame. **b**, A particle without a flame gives approximately the same cost function (indicated with ~) for both templates, seen from the particle-perspective, since the 'overlap' with both templates is the same (also shown in top row of **a**). On the contrary, from the template-perspective, the template without a flame gives a higher cost function than the template with a flame. **c**, A particle with a flame gives approximately the same cost function (indicated with ~) for both templates, seen from the template-perspective, since both templates completely 'overlap' with the particle (also shown in bottom row of **a**). On the contrary, from the particle-perspective, the template with a flame gives a higher cost function than the template without a flame.

ITM with Manders coefficients on simulated data

We tested ITM with the Manders coefficient approach on 200 simulated TUD-logos, of which 100 logos miss the flame above the letter T. The particles were simulated as if they were imaged with DNA-PAINT, with a 50% DoL and a mean localization uncertainty of 2.5 nm. Particle fusion of all particles does not reveal the particles without the flame (Fig. 3.12a). Clustering the particles with the dendrogram approach (section 3.2), gives two classes with 12 and 143 particles, respectively, that weakly show the two structures (Fig. 3.12b,c). The classification inability results from the presence of 44 misclassified particles in the first class (Fig. 3.12d) and from 45 discarded particles that are missing from these classes since pruning the dendrogram resulted in many small clusters which are discarded in the eigen image approach. For the dendrogram method, we used the average-linkage criterion, pruned the dendrogram into 20 clusters, and only applied the eigen image method on clusters with more than 5 particles. ITM with the Manders coefficient approach was applied for two iterations (templates are shown in Fig. 3.12e-h) to refine the classes. The resulting classes (Fig. 3.12i,j) clearly show the two structures and only 7 out of 200 particles are classified wrongly (Fig. 3.12k), which is a significant improvement compared to the initial classification without Manders coefficients, which contained 89 mistakes.

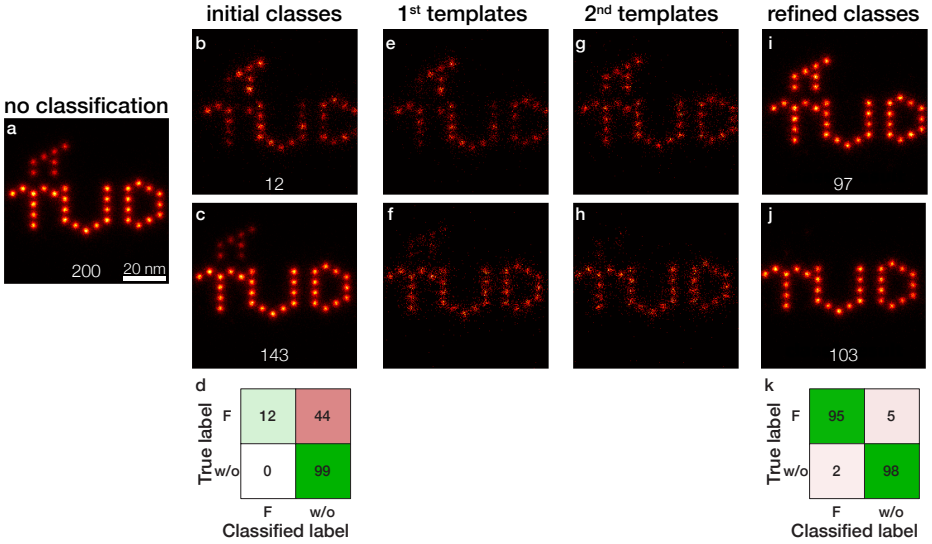


Figure 3.12 | Iterative template matching with Manders coefficients on 200 simulated TUD-logos containing 100 particles without a flame above the letter T. a, Without classification, the result of particle fusion does not reveal the particles without a flame. **b**,**c**, Classification with the dendrogram approach gives two classes with 143 and 12 particles, respectively. **d** Confusion matrix for the classification result of **b** and **c**, indicating the amount of correctly and wrongly classified particles with either a flame (F) or without a flame (w/o). **e**-**h**, Two iterations of the ITM approach with Manders coefficients are performed to refine the classes. After the first iteration, the templates are updated according to the new classes. **i**,**j**, The resulting refined classes containing 103 and 97 particles, respectively. **k** Confusion matrix for the classification result of **i** and **j**. Scale bar of **a** applies to **b**, **c**, **e**-**j**.

Conclusion

Given the two examples, we can conclude that iterative template matching, especially in combination with the use of Manders coefficients, is a promising technique to refine the classes and obtain a higher classification performance. However, there are a few points to take into consideration.

First of all, the initial classes that are used to construct the templates have to be of sufficient quality. If one class is significantly better reconstructed than the other class, the class with the highest SNR will attract all particles, which is also a common problem in template matching in cryo-EM [2]. Secondly, the classification performance is not the most optimal measure to quantify the success of the classification. Regarding the TUD-logos with and without a flame, it can happen that particles that are simulated having a flame, miss most of their flame due to low DoL. If we investigate the 7 wrongly classified particles of Fig. 3.12i,j, most of them are particles that are supposed to have a flame, but in reality they only have one or two active binding sites in the flame area (data not shown). In light of the classification performance, they are wrongly classified, but effectively, these particles belong to the correct class since they miss most of the flame. As an advantage, ITM can help to classify the particles that were lost in the process of pruning the dendrogram. As shown in Fig. 3.12, 45 particles are lost, due to the creation of clusters with too little particles. ITM can classify all particles by assigning them to one of the two templates, including these lost particles.

One concern regarding template registration is that it can enforce the template into the final result. We show the severity of this problem by template matching 307 experimental NPC acquisitions to three different templates: a circle and two different ellipses (Fig. 3.13). The same set of 307 particles perfectly show the structure of all three templates, indicating the danger of inferring structural knowledge about the data from a template-based registration approach. In our case, however, we do not suffer from template bias, since the templates are created by the data itself, using training- and prior knowledge-free classification, and are iteratively updated. Consequently, the templates are unbiased and can reliably be used for template matching.

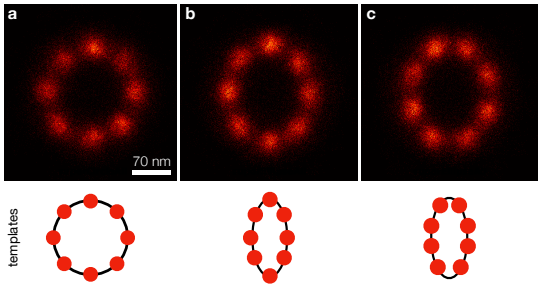


Figure 3.13 | Template-bias in template matching. Registration of 307 experimental NPC acquisitions [25] to **a**, a circular template with 8 sites, **b**, an elliptical template with ellipticity of 0.82 with sites at the vertex and, **c**, the same elliptical template but no sites at the vertex of the ellipse. Schematic representation of the used templates is shown below the reconstructions. This proves that the data is so 'weak' that it can be reconstructed into any desired shape, indicating the need for template-free registration. Scale bar of **a** applies to **b**,**c**.

As a last note, the ITM approach is designed and implemented to improve the classification using the dendrogram approach. Given the limitations of the dendrogram approach, the classification is suboptimal and we show that ITM improves the performance. The MDS approach in combination with the eigen image method (see Manuscript) is performing significantly better, eliminating the need to further refine the classes with ITM. However, for future work, it would be interesting to see the improvement ITM can bring to the already satisfying classification results obtained with MDS and the eigen images.

Particle exclusion based on intraclass statistics

Instead of exchanging particles between classes to refine the classification performance, another approach to improve the quality of the classes is to remove outliers. One idea to remove outliers from classes is to look at intraclass statistics, where we assume that all particles within one class share the same similarity. To quantify this similarity, we calculate the Bhattacharya cost function between every pair of particles within the same class. Since the particles are already aligned with each other, we do not have to perform registration, but simply calculate the cost function on the orientation they have in the class reconstruction. Ideally, every particle should have a high similarity with the other particles, where outliers result in a low similarity. For an ideal class consisting of N particles, we assume all $N(N-1)/2$ similarity measures to be high. Following this logic, removing an outlier from the class should decrease the standard deviation between the similarity measures. We iteratively remove one-by-one the particle that results in the biggest drop in standard deviation after their removal.

We test this idea on a dataset of 200 simulated TUD-logos with 50% DoL, of which 100 particles are mirrored. Classification with the dendrogram approach results in two classes, with 104 and 77 particles, respectively. The class with 104 particles contains 13 wrongly classified particles and the class with 77 particles contains 3 wrongly classified particles (Fig. 3.14a,b). When we perform one-by-one particle removal based on decreasing the total standard deviation between all pairwise similarity measures, we see that the wrongly classified particles are removed relatively early in the process (Fig. 3.14c,d). It does not show that this method can perfectly filter-out the outliers, but it indicates that the intraclass statistics hold information that can be used for outlier removal. Other options would be to remove particles based on increasing the mean or median, or by decreasing the variance, between all pairwise similarities. Note that we used the dendrogram classification approach to illustrate this idea, since the MDS method would result in near-perfect classification, not giving any incorrect classified particles to exclude from the classes.

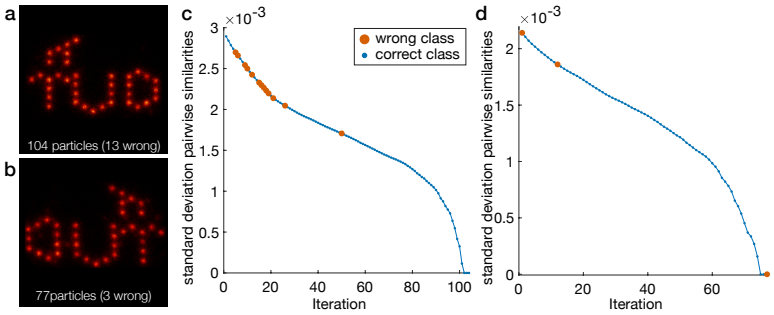


Figure 3.14 | Outlier removal based on intraclass statistics. **a,b** Classification result of the dendrogram approach on a dataset of 200 simulated TUD-logos of which 100 are mirrored. Class **a** contains 104 particles of which 13 are mirrored and wrongly classified. Class **b** contains 77 particles of which 3 are in the normal orientation and therefore wrongly classified. **c**, Evolution of the standard deviation over all pairwise similarities for class **a** after removing one-by-one the particle that causes the biggest drop in the standard deviation. Blue dots represent particles that are correctly classified and orange dots represent particles that are incorrectly classified. **d**, same as **c**, but for class **b**.

Particle exclusion based on MDS coordinates

Another approach to determine the outliers in a class is by using the multidimensional scaling space, which is formed by MDS on the dissimilarity matrix (see Manuscript). The classes are created by k-means clustering in this multidimensional scaling space, where we assume that the classes form spherical clusters. Within such clusters, the particles in the centre are the most representative particles of that class and the particles in the periphery are probably less similar. Presumably, outliers will be located on the outside the cluster and the best particles can be obtained by only taking the particles close to the centre.

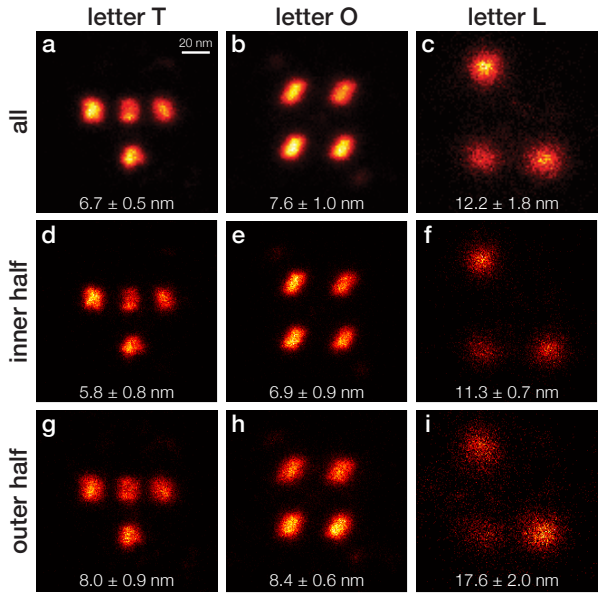


Figure 3.15 | Particles that are closest to the centre of the cluster in the multidimensional scaling space, are of the highest quality. **a-c**, The three representative classes obtained by classification of 800 particles from the letters dataset, which are imaged together. The particles are split into five classes (**Fig. 2v-z** of Manuscript), where here the classes with misfolds are not shown. Classes contain 170, 238 and 130 particles, respectively. Average FRC image resolution of 10 random splits is indicated below each reconstruction. **d-f**, Reconstructions of the half of the particles of **a-c**, which were closest to the centre of the cluster in the multidimensional scaling space. **g-i**, Same as **d-f**, but for the outer half of the particles. Scale bar of **a** applies to all.

To investigate whether the distance to the centre of each cluster holds information about their quality, we split each cluster into two equally sized sets based on the distance of every particle to the centre of the cluster. We test this idea on 800 particles from the letters dataset, where the classes are imaged together (Fig. 2u-z of the Manuscript). The particles are classified into five classes, including two classes containing the misfolds. The three classes showing the letters T, O and L are each split in half, the inner half with particles that are closest to the centre of the cluster and the outer half with particles that are furthest from the centre.

The particle fusion results of each half show that the particles closest to the centre create a reconstruction with better quality than the particles that were located far from the centre (Fig. 3.15). The image resolution calculated with FRC confirms that the reconstruction of the inner half has a higher resolution, indicating that the best quality particles are located in the middle of each cluster. We can conclude that the position of particles within the clusters in the multidimensional scaling space holds information about the quality of the particles. Taking only the inner core of the clusters, and therewith discarding particles, will improve the resolution of the classes.

3.4. 3D classification

In the Manuscript, we have shown the performance of the developed classification approach on multiple 2D SMLM datasets. Recent developments have extended the field of localization microscopy to three-dimensional imaging. One option to obtain information about the z-position of the emitter is by using astigmatism, in which the ellipticity and orientation of the PSF vary along the z-axis. By fitting a 2D elliptical Gaussian to the emitted spot, the x- and y-coordinate of the fit reveal the lateral position of the emitter and from the ellipticity and orientation of the fit, the z-position can be deduced. With this procedure, each localization is described by five parameters, three for the 3D position of the emitter (x, y and z) and two for the lateral and axial localization uncertainty (σ_{xy} and σ_z). In this chapter, we will explain how the classification method is adjusted to handle 3D SMLM data, and show the classification performance on simulated and experimental data as proof-of-principle.

Translation of the classification algorithm to 3D

The core of the classification approach for 3D data is the same as for 2D (see Fig. 1 of the Manuscript). The all-to-all pairwise registration is performed as in [18]. The registration procedure is essentially the same as in 2D, however, the Bhattacharya cost function has been extended to handle 3D coordinates and anisotropic localization uncertainties. We further optimized the 3D Bhattacharya cost function by adding normalizations with respect to the number of localizations and the localization uncertainties (see Section 3.1). The following steps of multidimensional scaling and subsequent k-means clustering are applied to the pairwise dissimilarities in the same way as in 2D. The 3D particle fusion per cluster is performed as described in [18].

The optional further clustering with the eigen images exploits the same principle as in 2D, however, eigen volumes are used instead of eigen images. In the process of calculating the eigen volumes, the localization data is converted into 3D pixelated images of size $N \times N \times N$, which are subsequently reshaped into vectors of size $N^3 \times 1$. The following steps of calculating the eigen volumes, projection onto the eigen volumes and clustering of the weights are the same as in the 2D approach (see Supplementary Note 1 of the Manuscript).

Example 1: Simulated NUP107 data with a shift between the two rings

The NUP107 model

We tested the 3D classification approach on simulated NPCs of which nucleoporin NUP107 is labeled. NUP107 is part of the NUP107-subcomplex [32] and consists of two rings with 16 copies arranged in pairs of two. We use the model for NUP107 of [18] where the rings are separated 59 nm in height and have a 13° azimuthal shift (Fig. 3.16a). Each ring consists of 16 emitters, grouped in pairs of two. The emitters of one pair are separated 13.59° and the two emitters are positioned on an outer and inner ring of 99.4 and 93.2 nm in diameter, respectively (Fig. 3.16b). We simulated a dataset of 500 particles using the NUP107 model, with 50% DoL and a mean lateral localization uncertainty of 3.5 nm (10.5 nm in the axial direction). The particles are simulated including a bleaching rate of 0.1 frame^{-1} as if they are imaged with *d*STORM, and therefore have different numbers or localizations per binding site. To create variation within the dataset, each NUP107 particle is simulated with a random shift between the upper and lower ring, uniformly distributed between 0° and 13°.

Classification result

The particle fusion result of all 500 particles clearly shows the 16 doublets of emitters (Fig. 3.16c). However, the continuous variation in the shift between the two rings is blurring the result. When we apply the 3D classification method to the same 500 particles, we find two classes with 266 and 234 particles, respectively. An isosurface plot based on the localization density shows that one class contains the particles with a small shift and the other class the particles with a bigger shift (Fig. 3.16d). Further investigation of the distribution of the shifts indicates that the shifts are divided over the two classes, where each class contains one half of the distribution (Fig. 3.16e). This example shows that the 3D classification algorithm is, as its 2D variant, able to correctly classify data with continuous variation, even though only 50% of the binding sites are labeled.

Note that for all shown data of NUP107, we used prior knowledge about the 8-fold symmetry of the NPC in the particle fusion algorithm. During bootstrapping, the third and last step of the particle fusion algorithm, every particle is in each iteration given a random additional rotation of $k \cdot \frac{2\pi}{8}$, with $k = 1 \ldots 8$, around the z-axis. This approach results in an equal division of the localization over the NPC structure, resolving the hotspot problem [17, 18].

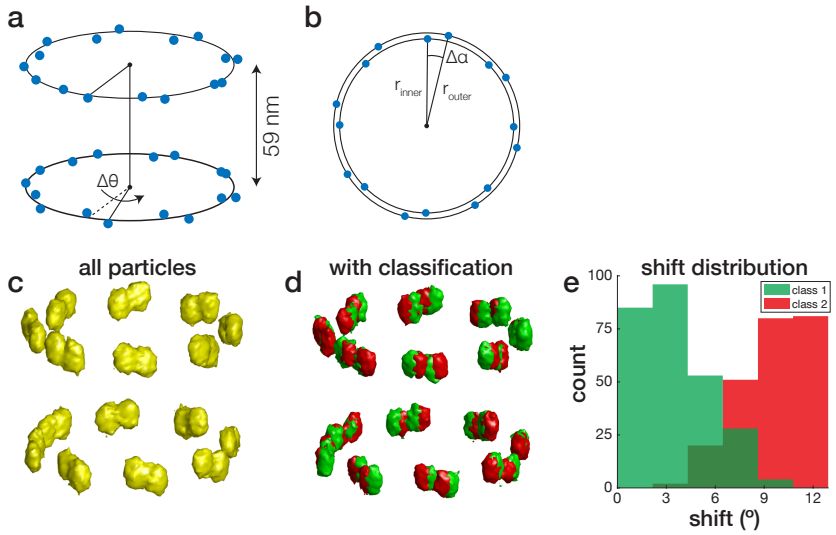


Figure 3.16 | Classification result on simulated NUP107. **a**, Model for NUP107 with two rings of 16 binding sites. The rings are 59 nm separated in height and have a relative azimuthal shift of 13° ($\Delta\theta$). **b**, The binding sites of each pair are separated by 13.59° ($\Delta\alpha$) and are located on one of two rings with radii of 46.6 (r_{inner}) and 49.7 nm (r_{outer}), respectively. **c**, Isosurface visualization of the particle fusion result of 500 simulated NUP107 particles with a uniformly distributed shift in the range 0-13°, mean lateral localization uncertainty of 3.5 nm, 50% DoL and simulated as if they are imaged with *d*STORM. **d**, Two-color overlay isosurface visualization of the classification result of **c**, containing 266 (green) and 234 (red) particles per class, respectively. **e**, Shift distribution per class. Shifts are the ground-truth shifts used in simulating the particles.

Example 2: Experimental NUP107 data

We also tested the 3D classification approach on an experimental Alexa647-labeled NUP107 dataset of 750 particles, imaged with 4Pi STORM in U2OS cells (previously described [18]). The 3D particle fusion result of all 750 particles clearly shows the 8-fold symmetry of the NPC (Fig. 3.17a). When we apply the 3D classification method to the same 750 particles, we find two classes with 391 and 359 particles, respectively. An isosurface plot based on the localization density shows that one class contains the particles with most of their localizations in the upper ring and the other class the particles with more localizations in the lower ring (Fig. 3.17b). Quantification of the difference between the upper and lower ring shows indeed that the green class contains all particles with more localizations in the upper ring and the other class for the lower ring (Fig. 3.17c). To prevent that the classification selects on the density of each ring, we flip the green class to make sure that the lower ring of each particle has more localizations than the upper ring. When we classify all particles again (after flipping the green class), the classification selects the left (orange class, 372 particles) and right side (blue class, 378 particles) of the ring (Fig. 3.17d). This example shows that when there is no (or not enough) structural variation, the classification selects on localization densities (similar to Supplementary Fig. 9 of the Manuscript).

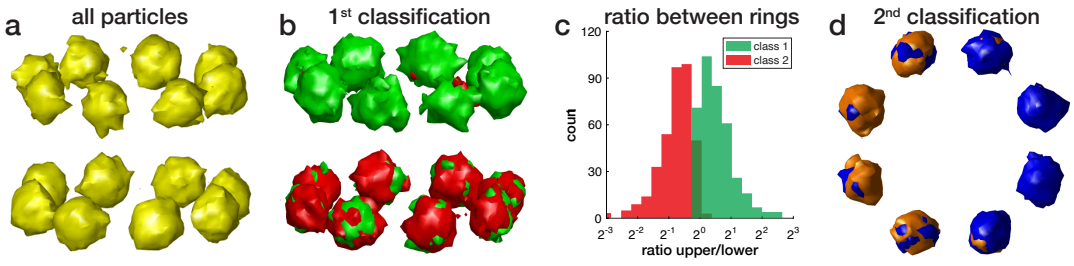


Figure 3.17 | Classification result of experimental NUP107 data. **a**, Isosurface visualization of the particle fusion result of 750 experimental NUP107 particles imaged with 4Pi STORM microscopy. **b**, Two-color overlay isosurface visualization of the classification result of **a**, containing 391 (green) and 359 (red) particles per class, respectively. **c**, Quantification of the difference in localizations in the two rings, expressed in the ratio of number of localization in the upper ring divided by the number of localizations in the lower ring. Note the logarithmic x-axis. **d**, Bottom view of the two-color overlay of the two classes resulting from a second round of classification on all particles, after flipping the green class upside-down. The classes contain 372 (orange) and 378 (blue) particles per class, respectively. Note that this bottom view only shows the lower ring. The upper ring has negligible density and does not show a difference between the classes.

4

Discussion

Disadvantages of the classification appraoch
Even though the promising results on multiple experimental and simulated datasets, some aspects of the classification method require careful consideration. First of all, the method knows one free parameter, *K*, which is the number of clusters in k-means clustering of the multidimensional scaling space. The parameter *K* has to be chosen by the user and determines the number of output classes. If the number of classes within the dataset is known, the choice for *K* is trivial. Often, however, the number of classes is unknown and selecting the optimal *K* requires tweaking. As explained in the Manuscript, visual inspection of the organization of the first dimensions in the multidimensional scaling space can give a valid indication about what value for *K* to use. If the first dimensions show clearly separated clusters, *K* can be chosen to match the number of clusters. However, when the particles have low DoL, high localization uncertainties or the variation between the different classes is small, the scatter plot will be a single point cloud and does not give a clear indication for the choice of *K*. If the goal is to find a small subgroup of structurally different particles, we advise to use a high value for *K*. As an example, we use *K=40* for the dataset with 2% mirrored TUD-logos. The user can start with an educated guess for *K* and then vary *K* to manually inspect what number of clusters is preferred. This can be done quickly as the process of clustering and subsequent particle fusion per cluster is significantly faster than the already-performed all-to-all registration.

Another point of concern is the computational cost of the classification approach. The method depends on the all-to-all registration of all particles, which is the most time-consuming step in the algorithm. Classifying 5000 particles of the digits dataset would take multiple days, even when parallelizing the process over 40 cores in a high-performing GPU cluster. Since the all-to-all registration of N particles scales with N^2, it is beneficial to split the dataset into multiple subgroups, perform the classification per group and combine the results in the end. We experience that the classification performance of classifying a subgroup of 500 particles is equal to the performance on all 5000 particles. This allows for speeding up the process while maintaining the same classification performance.

As a third point, one can question what degrees of structural variability can be detected with the developed method. Important factors determining whether variation in the data can be found are the localization uncertainty and the density of labeling. When the localization uncertainty is higher than the variation between the classes, the variation will be concealed and can therefore not be detected with classification. Furthermore, a low density of labeling will hinder the classification. When the particles are labeled too sparsely, two particles cannot be registered correctly to each other. This will result in low classification performance since the classification is completely dependent on the similarity values resulting from the pairwise registration. A lower bound for the density of labeling necessary for correct classification is hard to define, since it depends on the total number of binding sites, the structure of the particle, the degree of variation between the classes and the localization uncertainty.

In case of a small subset of structurally different particles, we experience that it is required that the rare class contains at least approximately 10 particles, in order to be able to detect them. The reason is that a class with fewer particles is not easily captured by k-means clustering of the multidimensional scaling space.

Improvements
Chapter 3.3 already described a number of strategies to improve classification performance. The classes can be finetuned by interchanging particles between classes based on the similarity between a particle and the class averages, or by discarding outliers from classes to increase the purity/uniformity of the particles within the same class.

A fundamentally different approach for classification would be to iteratively alternate between clustering and particle fusion. In this way, we start by clustering the particles into many small groups and merge the particles per group. These small groups are iteratively clustered and merged until the desired number of clusters is reached. This approach, in contrast to the method in the Manuscript, is not solely based on the pairwise registration between pairs of particles. Since the clusters gradually grow in the number of particles, the effective labeling goes up and therewith the reliability of the registration between clusters. The disadvantage is the computational cost of the pairwise registration in every iteration of the algorithm. Even though the number of clusters will decrease each iteration, the total number of necessary registrations is significantly higher than in our developed method.

There are other smaller improvements to the developed method that are worth investigating in order to increase the classification performance. First of all, the conversion from similarity to dissimilarity values can possibly be improved by using other ideas than the linear conversion used in the Manuscript. Secondly, it would be interesting to investigate if there are other similarity measures between point clouds that are less sensitive to the uneven distribution of localizations over the binding sites, from which the Bhattacharya cost function suffers. A third improvement could be to optimize for the number of initialized angles that are used in the GMM registration for each dataset specifically. The pairwise registration results in the similarity measures on which the entire classification pipeline is based, better registration will, therefore, lead to more reliable classification. One way to optimize for the number of angles is by doing multiple registrations with increasing number of initialized angles and see when the cost function values plateaus, indicating that taking more angles is not beneficial. Another option could be to do multiple registrations with very high numbers of initialized angles and investigate the smoothness of the cost function over the angles, where a smooth function allows for initialization with less angles. As a fourth improvement, it would be interesting to, instead of assigning every particle to a specific cluster using k-means clustering, giving each particle a weight that expresses the probability of belonging to each cluster. In this way, each class can be constructed using only the particles that have a probability for that class above a certain threshold.

Conclusion
Despite the above-mentioned points of concern about the method, we have proven the indisputable classification performance on a range of different datasets, with only one free parameter. The developed classification tool works directly on localization data, including the (possibly anisotropic) localization uncertainties and it can handle dataset with incomplete labeling and class imbalances. Next to that, the method can be easily extended to 3D and does not require template matching, prior knowledge or any training. In the future, the goal is to further develop the classification method and apply it to biological data. It will allow for structural classification in cellular imaging, to study biological variability and drug induced variations.

Acknowledgements

First of all, I would like to thank Bernd Rieger for giving me the opportunity to conduct my master thesis research in his group and trust me with the classification project. Thank you for answering the many questions I had throughout the project, the open-door policy, all the suggestions during our weekly meetings and the help with orienting on my next step after finishing my master. Secondly, I would like to thank Sjoerd Stalling for enlightening our weekly meetings with great suggestions and new ideas. I would like to recognize the substantial feedback and help from Bernd and Sjoerd in constructing and editing the Manuscript.

Next, I would like to express my gratitude to Hamid for answering all my questions and helping me with understanding the code, Maarten for our countless useful discussions in the office about our projects, Wenxiu for the successful application of my method on her data, and Natalie, Ewout, Thijs, Anish, Valentino and Aida for the nice atmosphere in the student room.

I would like to recognize the valuable assistance of Ronald for helping me with all my questions about the hpc-cluster, mex-files and and git, and Serge Donkers and Frans Vos for all the bureaucratic trouble we had to fight to get my double degree program settled.

Special thanks to Gabriele for proofreading this thesis and listening to my endless stories about classification in the last year.

A

Classification techniques in single-particle averaging for cryo-electron microscopy

Introduction

In order to understand the working principles of life and use this knowledge to develop new drugs and therapies, an understanding of all molecular components is necessary. To gain insight into the function and activity of life's smallest building blocks, it is crucial to identify the structure of proteins and protein complexes. Since the 1950s, the field of structural biology aims to visualize the smallest structures of life. Over the last decades, giant improvements have been made which brought the resolution down to near-atomic, sub-Ångström, length scales. In this review, we will focus on electron microscopy and aim to summarize the current classification strategies that are used in the image reconstruction software packages.

There are multiple techniques available to study the structure of biological molecules. Using X-ray crystallography, the structure is reconstructed by looking at the X-ray diffraction pattern of crystallized molecules. Disadvantages are that for many proteins it is hard to find the experimental conditions in which the protein crystallizes, the crystal structure may change the physiological conformation of the protein and large amounts of sample are needed. Also, the structure is deduced from the diffraction of all molecules together, which makes it impossible to visualize individual proteins. Another technique, nuclear magnetic resonance (NMR), can measure the structure of molecules in water solution. However, this only works for small molecules, in the range of 50 kDa [2].

The most popular modality in structural biology is cryo-EM. This technique was selected as the Method of the Year 2015 by Nature Methods [30] and awarded the Noble Prize in Chemistry in 2017. In cryo-EM, biological macromolecules are rapidly frozen in a thin layer of virtuous ice. An electron beam is passed through the sample creating a tomographic 2D projection on a planar detector. Since the single molecules are randomly located and oriented in the layer of ice, the acquired 2D images contain projections covering all 3D rotation angles of the structure of interest. Using computational techniques, the 3D representation of the electron-density of the sample can be reconstructed from the large set of 2D projections.

The process of reconstructing a 3D volume out of approximately 10^5 particles involves multiple computational steps. In the last 40 years, many groups have worked on algorithms to perform this reconstruction. Consequently, there are currently multiple software packages available which all utilize different strategies and methods. However, most packages follow the same general pipeline (Fig. A.1). After picking the particles from the electron micrograph, further referred to as images, the images are grouped in classes based on similarity. Ideally, each class contains images originating from one projection angle. The classes are averaged to reduce the noise and used to reconstruct an initial 3D model. Further steps involve refinement of the 3D model and possible classification of different conformations.

Multiple factors make the reconstruction challenging, such as low signal-to-noise ratio (SNR), the localization and orientation of the 2D projections are unknown and the sets of images are immense. In the reconstruction pipeline, several steps require classification, which is also hampered by the above-mentioned issues. In this

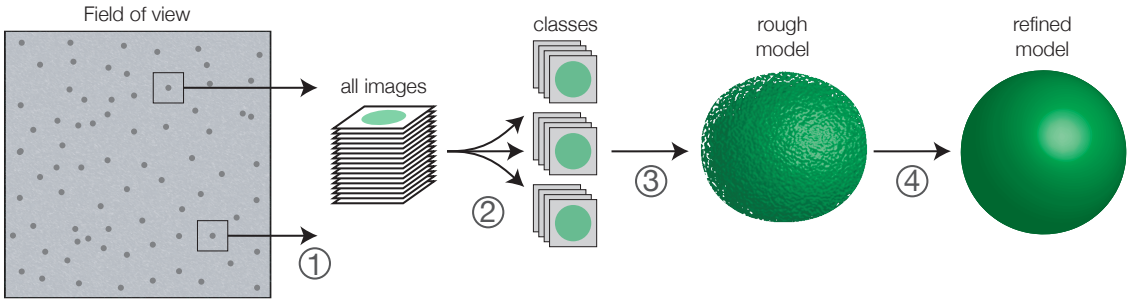


Figure A.1 | The general pipeline of single-particle averaging in EM. 1, Particle picking from the electron micrograph. **2**, Image alignment and classification into multiple classes based on similarity. Each class represents a 2D projection viewing direction of the 3D structure. **3**, The class averages are used to make a 3D reconstruction of the structure. **4**, Refinement of the 3D model.

review, we aim to summarize the current classification strategies in EM and indicate how they are used and combined in modern software packages.

Classification in cryo-EM pipeline

Given the general pipeline of Fig. A.1, there are at least two processes that involve classification. The first one is the classification of images into classes that represent the same viewing direction (step 2 in Fig. A.1). These classes are averaged to increase the SNR and can subsequently be used for constructing the 3D model, improving the particle picking, allowing for manual initial assessment of the images and removal of uninformative data. The second classification process is identifying possible different conformations of the protein, which is usually done in the 3D refinement step (step 4 in Fig. A.1). The to-be-determined 3D structure of the molecule is unknown, as well as the viewing directions underlying each 2D projection image. This means that the classification procedure of the images can be defined as an unsupervised classification problem. For that reason, the classification essentially comes down to clustering.

Clustering step 1: from images to class averages per viewing direction

The first process that requires clustering is the grouping of images into classes per viewing direction. The idea is that every class contains images that represent one viewing direction and therefore look similar. Each class of images will later be averaged to serve as input for the initial 3D model reconstruction. The entire classification of images into 2D classes knows three general steps: alignment, dimensionality reduction and clustering, where different software packages use different methods for each step.

1. Alignment

For the first step, the alignment of the images, two different groups of algorithms are available, namely multiple-reference alignment and reference-free alignment [49]. Software packages that use the multi-reference alignment algorithms are EMAN.2 [45] and SPARX [20], whereas Spider [12] uses the reference-free alignment algorithm.

Multiple-reference alignment (MRA) starts with a reference image for each viewing direction to which all images are registered and assigned [48]. By iteratively repeating the process of assigning images and updating the references, stable class averages can be found. A common technique for alignment is to cross-correlate the images with each other to find the correct translation and orientation [11]. Registration can also be done by using correntropy as similarity measure instead of correlation [43] or by using rotationally invariant features [50].

To make the alignment and assignment of images to classes less sensitive to noise, the robust criterion is proposed [43]. Instead of only looking at the similarity between the image and the class averages, every already assigned image is also compared with the class averages. The image is not simply assigned to the class average with the highest similarity, but the similarity value is compared to the similarities between the images within that class and the specific class average. Next to that, images tend to be assigned to the class with the highest SNR. Classes have different SNRs due to the variation in the number of images that are assigned to each class. To prevent these misclassifications and make the clustering more robust, a divisive approach is proposed [43]. Here, in each iteration the largest class is split into two smaller classes, to prevent the creation of unequal classes and therewith varying SNRs.

However, the greatest pitfall of MRA techniques is the initialization of class averages. If a 3D model of the protein of interest is known, multiple 2D projections can serve as initial class averages. Although, in most cases, an estimate of the structure is a priori not available. In this case, random images can be chosen as initial class averages [48].

In reference-free alignment (RFA), the second main group of alignment algorithms, all images are registered to each other without the use of a reference [33]. First, the images are combined into one initial average, which is later iteratively refined by a bootstrapping algorithm. In the first step, all images are combined in random sequential order. To start with, two random images are aligned to each other. In the next step, a third image is registered to the average image of the first two and combined into the average, which is sequentially repeated to include all images. In the second step, the global average is refined by iteratively taking one image out of the average and re-aligning it to the global average. In this way, sequentially all images are removed and re-aligned to the global average, and this can be repeated for multiple steps until a stable average is reached. Even though this algorithm is free of template-bias, it is not an optimal solution, since it depends on the choice of the first two images that are combined.

2. Dimensionality reduction
After aligning all the images using one of the above-mentioned strategies, the next step is to cluster the images into groups per viewing direction (Fig. A.2). To do this, we need a measure for the similarity between the images. Since the images consist of many pixels, taking every pixel as a feature would result in a high-dimensional problem. One way to reduce the complexity, is to apply dimensionality reduction techniques with so-called multivariate statistical analysis (MSA). In cryo-EM, correspondence analysis (CA) is the most commonly used MSA technique [10]. CA computes the low-dimensional space in which the mutual dissimilarities between the images are best preserved. The technique is comparable to principal component analysis (PCA), but uses the relative intensity values of the pixels. This makes CA less sensitive to the absolute intensity differences between images. The computation of the low-dimensional space is the result of remapping the high-dimensional image space onto new axes. The new axes, hereafter referred to as factors, are ordered based on how much variation of the data they explain. In principle, the algorithm gives as many factors as dimensions in the original dataset, yet the power of CA rests in the fact that taking only the first few factors is enough to preserve the greater part of the variance in the data, while essentially removing the noise.

In the case of RFA, all images are aligned to each other and CA can be applied to represent each image in a lower dimension [47]. This procedure can be difficult because individual images have a low quality since they contain high levels of noise. The MRA algorithm produces multiple aligned clusters, which, after averaging, each have a higher SNR than the individual images. When CA is applied on the (mutually aligned) cluster averages, this results in more robust principle component in which all the images can subsequently be decomposed to give the factors [48]. To go from all images to the factors, one can either take the MRA or the RFA approach (left part of Fig. A.2).

3. Clustering
After the CA procedure, each image is represented by 2-8 factors [47, 48]. The next step is to use this low-dimensional representation to cluster the images into distinct groups. Different algorithms are available to perform the clustering, for instance, k-means and hierarchical agglomerative clustering [46] (right part of Fig. A.2). The first and most simple method, which makes use of the new coordinates directly, is k-means clustering. One option is to represent the images as points in a two-dimensional scatterplot [47], where the x- and y-coordinates represent the first two factorial components. The resulting clusters of images have similar factorial components, so probably look similar. The k-means algorithm can in principle be applied to any number of dimensions [46].

Another way to divide the images into groups is hierarchical agglomerative clustering (HAC). A commonly used variant of this algorithm is the variance-oriented HAC method [48]. This algorithm starts by taking every image as being a cluster and sequentially merges the two clusters which together have the lowest intra-class variance. The merging of classes proceeds until all images are in one big class. The merging history and the corresponding values of the intra-class variance at each step can be plotted in a classification tree (Fig. A.2) . The desired number of classes can be obtained by pruning the classification tree at a particular level. The same principle can be applied in the reverse direction by starting with one class and repeatedly making a split that results in the biggest drop in intra-class variance. The number of factorial components taking into account when calculating the intra-class variance usually varies from two to six [33].

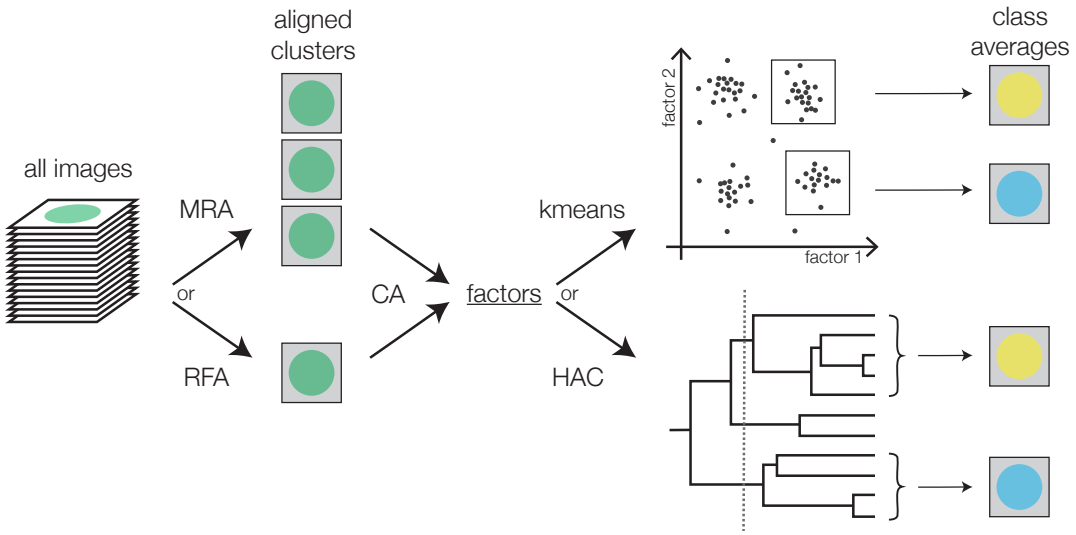


Figure A.2 | Classification to form class averages. All images are aligned with each other. This can be done by either multi-reference alignment (MRA) into multiple aligned clusters or by reference-free alignment (RFA) into one aligned cluster. Correspondence analysis (CA) is applied to the clusters and reduces the dimensionality into several factors per image. The images are clusters based on these factors, either by k-means clustering or hierarchical agglomerative clustering (HAC). The clustering results in several clusters of images, which are averaged into class averages.

Bayesian approach

A completely different and very popular technique is using a maximum-likelihood (ML) method. Initially, this technique was only used for the alignment of the images [41]. However, applying ML to the alignment and classification simultaneously resulted in the popular and successful software package RELION [39]. The idea behind ML is to find the most likely model that describes the data, where in this case, the models are the class averages. The models are iteratively updated by making use of the expectation-maximization algorithm. In the expectation-step, for each image, the probability that it is a result of every model is calculated. In the maximization-step, the model parameters are updated using partial derivatives of the expectation with respect to all the model parameters. The Bayesian character of the algorithm results in a more robust routine to find the class averages. A disadvantage is that the calculations are computationally expensive since the algorithm needs to go through all images in every iteration.

Other classification strategies

To further improve the performance of the classification in general or for a specific problem, numerous variants on the above-described algorithms have been developed. The aforementioned CA technique reduces the dimensionality of the dataset by projecting the images into a lower-dimensional vector space, which is used for clustering. Despite the reduction to two to eight dimensions, analysis of the structure of the data and the investigation of trends that are captured in more than two features, remains challenging. To allow for visual inspection of the result of the correspondence analysis, non-linear mapping can be used [35], which further reduces the dimensionality to only two features. This technique maps the data onto a two-dimensional map, by preserving the underlying dissimilarities as well as possible. The new representation shows the overall ordering in a two-dimensional map, in which trends and patterns can be visually detected.

Clustering step 2: finding different conformations of the 3D model

In the first part of the review, we described different algorithms that are used to cluster the images into different classes that each represent a viewing direction of the projected structure of interest. These images are averaged per classes and used for the 3D reconstruction of the electron density map. Another step in the reconstruction pipeline that requires classification is finding the different conformational states. Here we will describe several methods that are currently used to find these different structures.

e2refinemulti

The most simple method for 3D classification is e2refinemulti, part of the EMAN.2 software package [27]. This algorithm (Fig. A.3a) starts with any number of 3D models as input, which can be estimated guesses [4] of the different conformational states or randomly perturbed versions of the general structure [7]. From each of the N 3D models, a number of 2D projections are made over different angles. In the next step, all the available images are matched to all projections and assigned to the projections with the greatest similarity. The resulting clusters of images are reconstructed into N new 3D models, and this process is repeated until convergence. The major disadvantage of this strategy is the bias towards the chosen initial models.

e2refine-split

Another method, that does not suffer from initialization bias and is part of the EMAN.2 software package, is e2refine-split [27]. This strategy starts with the class averages that are created after the first classification step in the pipeline (Fig. A.1). To each set of aligned images that belong to a class average, PCA is applied, which identifies the most prominent differences between the images (Fig. A.3b). The resulting factorial components are used to cluster the images of each class average into two groups, A and B, which correspond to two class averages per viewing direction. The next step is to assign each class average to either 3D model X or Y. This is done for each set of class averages, starting with the set containing the highest number of images. For each assignment, it is determined whether the average A or B should be matched to either model X or Y, by comparing the 3D Fourier volumes. After assigning all sets off averages to one of the models, two 3D models are reconstructed. This strategy is more robust and accurate than the previously described e2refinemulti algorithm since it does not require initial estimates of the different models.

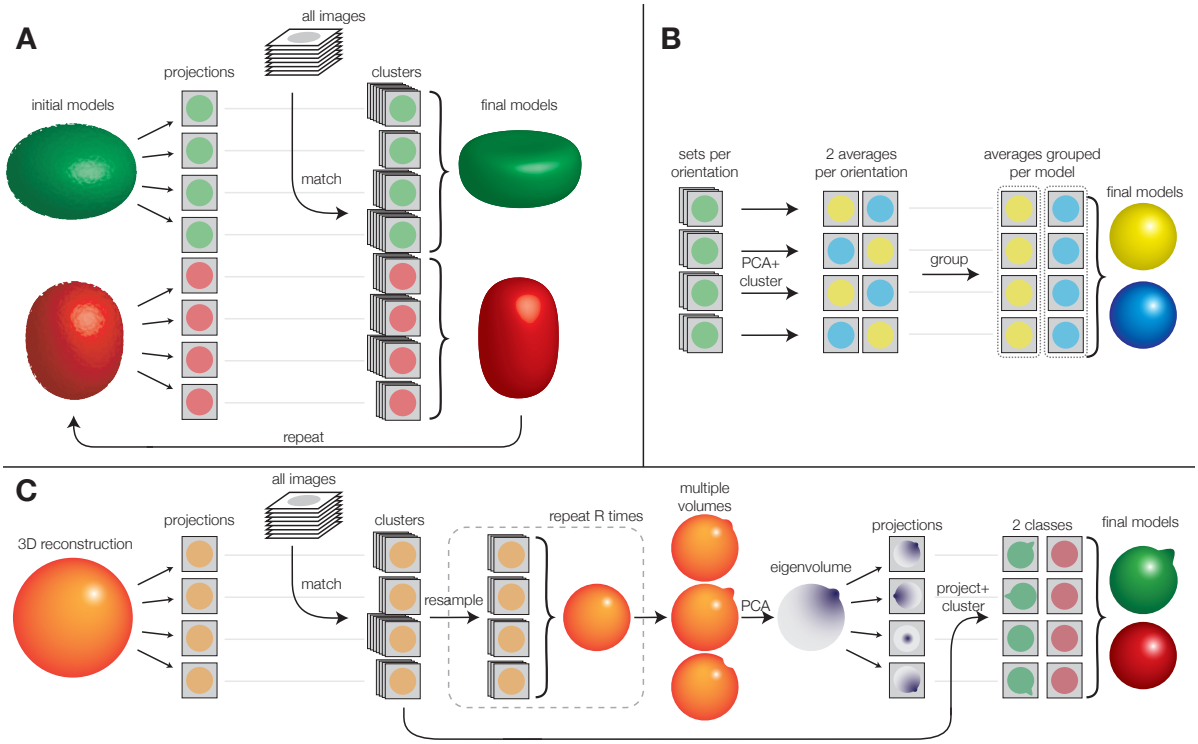


Figure A.3 | Classification of conformational states. A) The e2refinemulti algorithm starts with multiple initial 3D models (depicted in green and red). All images are matched to 2D projections of the initial models and the resulting groups of images are reconstructed to form new 3D models. This can be repeated till convergence. B) In e2refine-split, PCA is applied to each set of images representing a viewing direction. The resulting factorial components are used to cluster the images per orientation into two groups (either yellow or blue), which are grouped per model and reconstructed accordingly. C) The codimensional PCA algorithm starts with an initial 3D reconstruction and matches all the images to the 2D projections of the reconstruction. The clusters of images are resampled into equally big groups and reconstructed. This is done for multiple rounds of resampling resulting in different 3D reconstructions. PCA on the 3D volumes gives an eigenvolume that indicates the variation between the different volumes (illustrated by the bulge on the top-right side of the sphere). The 2D projections of the eigenvolume are used to determine for each image in the clusters to which class they belong. The resulting classes are reconstructed into two different 3D models.

Codimensional PCA
One algorithm investigating the 3D variability of the structure is codimensional PCA [34], which combines the ideas of e2refinemulti and e2refine-split. To prevent that PCA selects on other factors than variations in the structure, for example, different defocus and noise levels, it is better to apply it on the 3D reconstruction than on the class averages. The procedure starts by matching all the images to 2D projections of one initial 3D model, which results in sets of images per projection of different sizes (Fig. A.3c). To eliminate artifacts due to a non-uniform distribution of the images over the viewing directions, a HyperGeometric Stratified Resampling scheme (HGSR) is used. It resamples the images per viewing direction to create classes with uniform sizes, which are used to reconstruct a 3D model. This procedure is repeated for different rounds of resampling, resulting in multiple 3D models. To detect the variability in 3D, PCA is applied to the 3D models to create eigenvolumes. Eigenvolumes represent the variability between the different 3D models. The first eigenvolume is reprojected onto the same viewing directions as in the first step of the algorithm. The initial clusters EM images are projected onto the reprojections of the eigenvolume and clustered into two classes per reprojection based on the factorial components. The resulting classes are reconstructed into two different 3D models.

Conclusion

We have shown different classification techniques that are used in the reconstruction pipeline of cryo-EM. In single-particle averaging, image classification is a required step to cluster the images per viewing direction before creating the class averages. Another step where classification is important is the identification of different 3D conformations of the structure of interest.
With the acquired knowledge of how image classification is performed in the cryo-EM field, we can start applying and adapting the aforementioned concepts to single-particle averaging in light microscopy, especially single-molecule localization microscopy (SMLM). Given the different ways of image acquisition, the nature of the acquired images in SMLM is totally different from EM images. In contrast to EM, the acquisitions in SMLM are lists of localization. Next to that, the acquisitions suffer from incomplete labeling and fewer images are available. These differences make that the classification strategies from cryo-EM cannot directly be applied to SMLM. However, the principles behind the algorithms can definitely serve as inspiration for the introduction of image classification into the field of localization microscopy.

Bibliography

[1] A. Auer, M.T. Strauss, S. Strauss, and R. Jungmann. nanotron: a picasso module for mlp-based classification of super-resolution data. *Bioinformatics*, 36(11):3620–3622, 2020.

[2] T. Bendory, A. Bartesaghi, and A. Singer. Single-particle cryo-electron microscopy: Mathematical theory, computational challenges, and opportunities. *arXiv preprint arXiv:1908.00574*, 2019.

[3] E. Betzig, G.H. Patterson, R. Sougrat, O.W. Lindwasser, S. Olenych, J.S. Bonifacino, M.W. Davidson, J. Lippincott-Schwartz, and H.F. Hess. Imaging intracellular fluorescent proteins at nanometer resolution. *Science*, 313(5793):1642–1645, 2006.

[4] J. Brink, S.J. Ludtke, Y. Kong, S.J. Wakil, J. Ma, and W. Chiu. Experimental verification of conformational variation of human fatty acid synthase as predicted by normal mode analysis. *Structure*, 12(2):185–191, 2004.

[5] J. Broeken, H. Johnson, D.S. Lidke, S. Liu, R.P.J. Nieuwenhuizen, S. Stallinga, K.A. Lidke, and B. Rieger. Resolution improvement by 3d particle averaging in localization microscopy. *Methods and applications in fluorescence*, 3(1):014003, 2015.

[6] A. Burgert, S. Letschert, S. Doose, and M. Sauer. Artifacts in single-molecule localization microscopy. *Histochemistry and cell biology*, 144(2):123–131, 2015.

[7] D. Chen, J. Song, D.T. Chuang, W. Chiu, and S.J. Ludtke. An expanded conformation of single-ring groel-groes complex encapsulates an 86 kda substrate. *Structure*, 14(11):1711–1722, 2006.

[8] M. Ester, H. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[9] J. Frank. Single-particle reconstruction of biological macromolecules in electron microscopy–30 years. *Quarterly reviews of biophysics*, 42(3):139–158, 2009.

[10] J. Frank and M. Van Heel. Correspondence analysis of aligned images of biological particles. *Journal of molecular biology*, 161(1):134–137, 1982.

[11] J. Frank, W. Goldfarb, D. Eisenberg, and T.S. Baker. Reconstruction of glutamine synthetase using computer averaging. *Ultramicroscopy*, 3(3):283–290, 1978.

[12] J. Frank, M. Radermacher, P. Penczek, J. Zhu, Y. Li, M. Ladjadj, and A. Leith. Spider and web: processing and visualization of images in 3d electron microscopy and related fields. *Journal of structural biology*, 116(1):190–199, 1996.

[13] R.D.M. Gray, C. Beerli, P.M. Pereira, K.M. Scherer, J. Samolej, C.K.E. Bleck, J. Mercer, and R. Henriques. Virusmapper: open-source nanoscale mapping of viral architecture through super-resolution microscopy. *Scientific reports*, 6:29132, 2016.

[14] S. Habuchi. Super-resolution molecular and functional imaging of nanoscale architectures in life and materials science. *Frontiers in bioengineering and biotechnology*, 2:20, 2014.

[15] S.W. Hell. Microscopy and its focal switch. *Nature methods*, 6(1):24–32, 2009.

[16] R. Henderson. Avoiding the pitfalls of single particle cryo-electron microscopy: Einstein from noise. *Proceedings of the National Academy of Sciences*, 110(45):18037–18041, 2013.

[17] H. Heydarian, F. Schueder, M.T. Strauss, B. Van Werkhoven, M. Fazel, K.A. Lidke, R. Jungmann, S. Stallinga, and B. Rieger. Template-free 2d particle fusion in localization microscopy. *Nature methods*, 15(10):781–784, 2018.

[18] H. Heydarian, A. Przybylski, F. Schueder, R. Jungmann, B. van Werkhoven, J. Keller-Findeisen, J. Ries, S. Stallinga, M. Bates, and B. Rieger. Three dimensional particle averaging for structural imaging of macromolecular complexes by localization microscopy. *bioRxiv*, 2019. doi: 10.1101/837575. URL `https://www.biorxiv.org/content/early/2019/11/10/837575`.

[19] J.E. Hinshaw and R.A. Milligan. Nuclear pore complexes exceeding eightfold rotational symmetry. *Journal of structural biology*, 141(3):259–268, 2003.

[20] M. Hohn, G. Tang, G. Goodyear, P.R. Baldwin, Z. Huang, P.A. Penczek, C. Yang, R.M. Glaeser, P.D. Adams, and S.J. Ludtke. Sparx, a new environment for cryo-em image processing. *Journal of structural biology*, 157(1):47–55, 2007.

[21] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3): 264–323, 1999.

[22] B. Jian and B.C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1633–1645, 2010.

[23] R. Jungmann, C. Steinhauer, M. Scheible, A. Kuzyk, P. Tinnefeld, and F.C. Simmel. Single-molecule kinetics and super-resolution microscopy by fluorescence imaging of transient binding on dna origami. *Nano letters*, 10(11):4756–4761, 2010.

[24] T. Klein, S. Proppert, and M. Sauer. Eight years of single-molecule localization microscopy. *Histochemistry and cell biology*, 141(6):561–575, 2014.

[25] A. Löschberger, S. van de Linde, M. Dabauvalle, B. Rieger, M. Heilemann, G. Krohne, and M. Sauer. Super-resolution imaging visualizes the eightfold symmetry of gp210 proteins around the nuclear pore complex and resolves the central channel with nanometer resolution. *Journal of cell science*, 125(3): 570–575, 2012.

[26] A. Löschberger, C. Franke, G. Krohne, S. van de Linde, and M. Sauer. Correlative super-resolution fluorescence and electron microscopy of the nuclear pore complex with molecular resolution. *Journal of cell science*, 127(20):4351–4355, 2014.

[27] S.J. Ludtke. Single-particle refinement and variability analysis in eman2. 1. In *Methods in enzymology*, volume 579, pages 159–189. Elsevier, 2016.

[28] E.M.M. Manders, F.J. Verbeek, and J.A. Aten. Measurement of co-localization of objects in dual-colour confocal images. *Journal of microscopy*, 169(3):375–382, 1993.

[29] A. Mead. Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 41(1):27–39, 1992.

[30] Nature Methods. Method of the year 2015. *Nature Methods*, 13(1), 2015.

[31] R.P.J. Nieuwenhuizen, K.A. Lidke, M. Bates, D.L. Puig, D. Grünwald, S. Stallinga, and B. Rieger. Measuring image resolution in optical nanoscopy. *Nature methods*, 10(6):557, 2013.

[32] A. Ori, N. Banterle, M. Iskar, A. Andrés-Pons, C. Escher, H. Khanh Bui, L. Sparks, V. Solis-Mezarino, O. Rinner, P. Bork, et al. Cell type-specific nuclear pores: a case in point for context-dependent stoichiometry of molecular machines. *Molecular systems biology*, 9(1):648, 2013.

[33] P. Penczek, M. Radermacher, and J. Frank. Three-dimensional reconstruction of single particles embedded in ice. *Ultramicroscopy*, 40(1):33–53, 1992.

[34] P.A. Penczek, M. Kimmel, and C.M.T. Spahn. Identifying conformational states of macromolecules by eigen-analysis of resampled cryo-em images. *Structure*, 19(11):1582–1590, 2011.

[35] M. Radermacher and J. Frank. Use of nonlinear mapping in multivariate image analysis of molecule projections. *Ultramicroscopy*, 17(2):117–126, 1985.

[36] B. Rieger, R. Nieuwenhuizen, and S. Stallinga. Image processing and analysis for single-molecule localization microscopy: Computation for nanoscale imaging. *IEEE Signal Processing Magazine*, 32(1):49–57, 2014.

[37] P.W.K. Rothemund. Folding dna to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.

[38] M.J. Rust, M. Bates, and X. Zhuang. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm). *Nature methods*, 3(10):793–796, 2006.

[39] S.H.W. Scheres. A bayesian view on cryo-em structure determination. *Journal of molecular biology*, 415 (2):406–418, 2012.

[40] C. Sieben, N. Banterle, K.M. Douglass, P. Gönczy, and S. Manley. Multicolor single-particle reconstruction of protein complexes. *Nature methods*, 15(10):777–780, 2018.

[41] F.J. Sigworth. A maximum-likelihood approach to single-particle image refinement. *Journal of structural biology*, 122(3):328–339, 1998.

[42] R.R. Sokal. A statistical method for evaluating systematic relationships. *Univ. Kansas, Sci. Bull.*, 38: 1409–1438, 1958.

[43] C.O.S. Sorzano, J.R. Bilbao-Castro, Y. Shkolnisky, M. Alcorlo, R. Melero, G. Caffarena-Fernández, M. Li, G. Xu, R. Marabini, and J.M. Carazo. A clustering approach to multireference alignment of single-particle projections in electron microscopy. *Journal of structural biology*, 171(2):197–206, 2010.

[44] A. Szymborska, A. De Marco, N. Daigle, V.C. Cordes, J.A.G. Briggs, and J. Ellenberg. Nuclear pore scaffold structure analyzed by super-resolution microscopy and particle averaging. *Science*, 341(6146):655–658, 2013.

[45] G. Tang, L. Peng, P.R. Baldwin, D.S. Mann, W. Jiang, I. Rees, and S.J. Ludtke. Eman2: an extensible image processing suite for electron microscopy. *Journal of structural biology*, 157(1):38–46, 2007.

[46] M. Van Heel. Multivariate statistical classification of noisy images (randomly oriented biological macromolecules). *Ultramicroscopy*, 13(1-2):165–183, 1984.

[47] M. Van Heel and J. Frank. Use of multivariates statistics in analysing the images of biological macromolecules. *Ultramicroscopy*, 6(1):187–194, 1981.

[48] M. van Heel and M. Stöffler-Meilicke. Characteristic views of e. coli and b. stearothermophilus 30s ribosomal subunits in the electron microscope. *The EMBO journal*, 4(9):2389–2395, 1985.

[49] Y. Xu, J. Wu, C. Yin, and Y. Mao. Unsupervised cryo-em data clustering through adaptively constrained k-means algorithm. *PloS one*, 11(12), 2016.

[50] Z. Zhao and A. Singer. Rotationally invariant image representation for viewing direction classification in cryo-em. *Journal of structural biology*, 186(1):153–166, 2014.