DOI:
10.1038/nprot.2016.149

Document Version
Peer reviewed version

A Bayesian cluster analysis method for single molecule localisation microscopy data

Juliette Griffié[1], Michael Shannon[1], Claire L Bromley[2], Lies Boelen[3], Garth L Burn[1], David J Williamson[1], Nicholas A Heard[4], Andrew P Cope[5], Dylan M Owen[1*+] and Patrick Rubin-Delanchy[6*+]

[1]Department of Physics and Randall Division of Cell and Molecular Biophysics, King's College London [2]MRC Centre for Developmental Biology, King's College London, [3]Faculty of Medicine, Imperial College London, [4]Department of Mathematics, Imperial College London and Heilbronn Institute for Mathematical Research, [5]Division of immunology, infection and inflammatory Disease, Academic Department of Rheumatology, King's College London [6]Department of Statistics, University of Oxford and Heilbronn Institute for Mathematical Research. *These authors contributed equally. [+]correspondence to dylan.owen@kcl.ac.uk or delanchy@stats.ox.ac.uk

Abstract Cell function is regulated by the spatio-temporal organization of signaling machinery and a key facet of this is molecular clustering. Here, methodology is presented for the analysis of clustering in data generated by 2D single molecule localisation microscopy (SMLM), for example, photoactivated localisation microscopy (PALM) or stochastic optical reconstruction microscopy (STORM). Three features of such data cause standard cluster analysis approaches to be ineffective: the data take the form of a list of points rather than a pixel array; there is a non-negligible unclustered background density of points which must be accounted for; every localisation has an associated uncertainty on its position. These issues are overcome using a Bayesian, model-based approach. Many possible cluster configurations are proposed and scored against a generative model, which assumes Gaussian clusters overlaid on a completely spatially random background, before every point is scrambled by its localisation precision. We present the process of generating simulated and experimental data which are suitable for our algorithm, the analysis itself, and the extraction and interpretation of key cluster descriptors such as the number of clusters, cluster radii and the number of localisations per cluster. Variations in these descriptors can be interpreted as arising due to changes in the organization of cellular nanoarchitecture. The protocol requires no specific programming ability and the processing time of one data set, typically containing 30 regions of interest, is ~18h with ~1h of user input.

Introduction

In recent years, Single Molecule Localisation Microscopy (SMLM) has become a widely used technique. Conventional microscopy is limited in resolution to around 200 nm, which is the

closest two fluorescent molecules can be before they cannot be distinguished. This limit is due to diffraction, and is only dependent on the numerical aperture of the microscope objective and the wavelength of light, meaning it cannot easily be improved. Despite the great utility of diffraction limited microscopy, there are many biological structures and processes which occur on smaller length-scales. There is therefore a strong scientific motivation to develop methods which circumvent this limit. One of the most widely used techniques is SMLM[1-3], which exploits the temporal separation of fluorescing molecules to achieve resolutions in the 20-30 nm range.

The overarching principle behind SMLM is that if only a sparse distribution of molecules can be imaged at a time, then their positions can be accurately estimated by calculating the centre of each individual point spread function. The most common way of generating such a sparse distribution is by exploiting the stochastic nature of photo-physical processes, obtaining randomly separated fluorescent signals of different molecules in time. There are numerous ways of achieving this, including using photo-switchable fluorescent proteins[1], long-lived non-fluorescent dark states[4, 5], or transient molecular binding[6]. Once a sparse subset has been imaged, the centroid of individual point spread functions is found, typically by fitting a two-dimensional Gaussian kernel[7-9]. The position of the molecule is estimated as the location of the peak, with localisation uncertainty (also known as localisation precision) determined by the quality of the fit[9, 10]. This random subset is then bleached, and a new random subset activated. Through repeated cycles of activation, imaging and bleaching, the locations of a large subset of the available molecules is eventually acquired.

The clustering of molecules in biological systems is often critical to their function and therefore cluster analysis on data obtained from microscopy is an important analytical method. However, unlike conventional microscopes, which produce images consisting of arrays pixels each with a numerical value proportional to the intensity at that location, SMLM generates pointillist data, specifically a set of x-y (and z in the case of 3D acquisitions) coordinates with associated localisation precisions. Thus, the image analysis tools developed for conventional microscopy are not applicable. Instead, the data must be treated within the framework of spatial point pattern (SPP) analysis. At the same time, cluster analysis tools developed in this field do not handle the uncertainty in the positions of the molecules. Options are further reduced by the typical presence of a non-negligible background of points which are not clustered. We have therefore developed a new cluster analysis technique for SMLM data, using a Bayesian approach, to address all these issues. Our algorithm produces a full clustering of the points[11], i.e., a vector allocating each observation to a cluster or the background. The method is model-based, assuming Gaussian clusters overlaid on a completely spatially random (CSR) background, takes full account of the localisation precisions and does not require any arbitrary user-supplied analysis parameters but instead requires Bayesian prior probabilities which have well-defined statistical interpretations.

Applications

Despite being designed for SMLM data, the method is, in principle, applicable to any pointillist data set to which the 2D circular, Gaussian cluster model is applicable. Such examples may arise in diverse fields such as astronomy or ecology.

Here, we provide a precise guide to using the technique on simulated and experimental SMLM data. We recommend the use of ThunderSTORM[8] – an ImageJ plugin to localise the fluorophores for analysis. Our tool is flexible however, and can be used with any of the common or commercial localisation software of which there are many[7, 12, 13].

Our motivating application is the analysis of the clustering behaviour of molecules in, or proximal to, the cell plasma membrane. Membrane proximal signalling molecules are extremely common targets of study as all intercellular communication ultimately results in signal transduction through the plasma membrane. There is a large body of literature which now shows that in a wide range of signal transduction pathways, the clustering of molecules (either directly in the membrane itself, or proximal to it) is a regulating factor[14-19]. One manifestation of this regulation for example, is that clustering can digitise signalling, producing rapid and discretised cellular outcomes[20, 21]. The mechanisms for generating such clusters are diverse. One example are protein-protein interactions either resulting from direct binding domains (oligomerisation) or simple van der Waals interactions resulting from a Lennard-Jones potential[22]. Another might be clustering due to interactions with ordered membrane microdomains (lipid rafts); areas of the cell membrane with differential lipid packing to which membrane proteins have differential affinity[23-26]. The cytoskeleton can also influence clustering – cortical actin has been shown to corral membrane proteins both theoretically and experimentally[27-29].
While the morphology of the resulting clusters of course depends on the generative mechanism, in many cases, they can be reasonably approximated as 2-dimensional Gaussian clusters. This includes the case of a 3D membrane-proximal cluster projected into two dimensions. It is this type of 2D morphology which we address and analyse here; our tool is only appropriate if, for example by visual inspection, clusters are found to beat least approximately circular.

Method

Conceptually, the algorithm is formed of two parts[11]; a full schematic of the analysis workflow is shown in Figure 1. The first proposes several thousands of potential clustering configurations, called cluster proposals, by direct point pattern analysis of the data. The second scores each cluster proposal according to a Bayesian generative model. This allows the highest scoring cluster proposal to be identified, which is the main output of the algorithm. Tools for extracting key cluster descriptors such as cluster radii, number of

clusters per ROI or number of localisations per cluster, are also provided in a post-processing step.

Cluster proposal generation

Let $V = (V_1, \ldots, V_N)$ denote the list of 2D localisations provided, with associated localisation uncertainties $s_1, \ldots, s_N$ (treated as standard deviations). A cluster proposal is an assignment of every localisation to a specific cluster or to the background. This is represented by a vector of non-negative integers $\ell = (\ell_1, \ldots, \ell_N)$, where $\ell_i = \ell_j$ indicates that V_i and V_j are either in the same cluster, if $\ell_i \geq 1$, or the background, if $\ell_i = 0$. Two parameters, r and T, allow a single cluster proposal to be generated. A number of proposals are then generated by separately varying r and T. The cluster proposal mechanism proceeds as follows[30, 31]. Each localisation is first assigned a density estimate (transformed such that its value scales with r) based on the number, say k, of other localisations that are within a distance r,

$$\sqrt{A\,k/[\pi(N-1)]},$$

where A is the area of the ROI. Localisations with a density below T are assigned to the background. Those that remain are divided into clusters by connecting any pair less than a distance $2r$ apart. By default, the ranges considered by our algorithm are $r = 5, \ldots, 300$ nm and $T = 5, \ldots, 500$, both in increments of 5.

Generative model

The generative model assumes that the true molecular positions $Z_1, \ldots, Z_N$ follow a hybrid distribution whereby a certain proportion are completely spatially random, forming a so-called background process, and the remainder are grouped into Gaussian clusters. To each true molecular position, Z_i, we add independent circular Gaussian noise with variance s_i^2 (taken from the localisation uncertainty of V_i) in each dimension. For experimental data, these uncertainties are calculated theoretically for each localization. There are a number of theoretical derivations of these values, each taking into account parameters such as the number of photons per PSF, the width of the PSF its local background variance and the camera pixel size. Two of the most common were derived by Thompson et al[9] and Quan et al[10] respectively.

Points are independently assigned to the background with fixed prior probability p_B. Remaining points group into clusters according to the Dirichlet process, with concentration parameter α. These two prior assumptions determine our prior distribution on ℓ, denoted $p(\ell)$. The full effect of varying the priors has been analyzed in detail and the analysis has been found to be robust[32].

Clusters are mutually independent of each other. True molecular positions within a cluster are conditionally independent, drawn from a 2D circular Gaussian distribution, conditional on the cluster centre, *a priori* uniformly distributed on the ROI, and the cluster standard

deviation, a priori drawn from a user-supplied histogram. Together these assumptions determine the (marginal) likelihood of the data given ℓ, denoted $p(V \mid \ell)$. The main computational burden of the method is calculating this term, as it is not analytically available. Actual formulae and derivations are available in Rubin-Delanchy et al[11].

Following the central equation of Bayesian inference, any cluster proposal ℓ can therefore be assigned a posterior probability $p(\ell \mid V) \propto p(V \mid \ell)p(\ell)$, allowing selection of the optimal proposal. We have demonstrated the reliability of the scoring mechanism by showing overwhelming improvements in estimation accuracy of key cluster descriptors such as the number of clusters per region or percentage of localizations in clusters, compared to using arbitrarily (but sensibly) chosen proposals based on fixed *r* and *T* values. We demonstrated the reliability of the scoring mechanism on real data by dividing the localizations from a representative data set into two and showing that the algorithm produces consistent estimates for each sub-population[11].

Alternative approaches

The first cluster analysis method to be applied to SMLM data used Ripley's K-function[30, 33, 34]. Unlike our method, Ripley's K-function does not provide a full clustering of the data, but instead measures the average level of clustering at different scales for the region of interest (ROI) as a whole. The K-function is calculated by drawing concentric circles around each point and counting the number of neighbours encircled. Its value is then normalised to the overall localisation density and linearised such that its value scales with the radius of the circles rather than their area. Higher values of the K-function at a particular circle radius imply greater clustering at that length-scale. The K-function provides a rapid and robust overview of clustering behaviour in an ROI, and has a strong theoretical underpinning. On the other hand, it does not generate a full clustering of the data, nor key cluster descriptors such as the number of molecules per cluster or the number of clusters. A very closely related technique which has also been applied to SMLM data is Pair Correlation (PC)[35, 36]. Here, the circles are replaced by tori, to mitigate the effect of artefacts occurring at specific length scales propagating to other length-scales[37]. An example of such an artefact which motivated the development of PC is multiple blinking, discussed in the Limitations section.

While the above methods produce high-level summaries, there are a number of approaches which do generate a full clustering of the data. Possibly the most popular among these is DBSCAN[38], which has also been applied to SMLM[39]. This algorithm first chooses a subset of core points based on their local density (using a radius, r, and threshold T), then generates clusters by connecting any two points that are within r of each other, where at least one is a core point. The algorithm is computationally efficient and makes no modelling assumptions, which might make it more suitable for datasets where our model assumptions are strongly inaccurate, for example, if there are markedly non-circular clusters. The key disadvantage of this approach is that it requires the user to supply values for r and T, which strongly affect

the outcome, and there is no theoretical guidance on how these should be selected. In a similar vein to DBSCAN, Voronoi tessellation chooses a subset of points to be clustered based on the area of control of each point (acting as a density estimate), and then clusters the subset by connecting adjacent areas[40]. As with DBSCAN, there are no model assumptions, meaning that the approach may be more robust to e.g. non-circular clusters, but the user is required to choose a threshold, and again, there is no theoretical guidance. In summary, if the observed molecular distributions cannot be closely approximated as circular clusters – in the case of fibres for example – then segmentation techniques such as DBSCAN or Voronoi tessellation are more appropriate. For ease of comparison, the analysis software MIiSR and VividSTORM are recommended[41, 42]. In addition, specialised software exists for 2-colour co-cluster analysis[43, 44] and analysis of 3D features[45].

Overall, we recommend the use of Ripley's K-function or Pair Correlation to give a high-level overview of the clustering behaviour in the data. These serve as a complementary approach to ours. We would advocate the use of DBSCAN or Voronoi tessellation in situations where the assumptions of our model are deemed (strongly) unrealistic.

Limitations

Model assumptions

An issue facing any model-based approach to data analysis is the validity of model assumptions, the most important of which here is the assumption that each localization has a fixed, independent probability of being a member of a cluster and that the cluster shapes are well represented by circular Gaussian distributions. While we expect the algorithm to be robust to morphologically similar distributions, for example flat top clusters or low aspect-ratio ellipses, the algorithm is certainly not designed for the analysis of fibrous structures, extremely elongated or non-convex clusters. The default prior stipulates that each localisation has a 50% probability of being clustered, meaning that completely spatially random (CSR) distribution is extremely unlikely *a priori.* Our position is that exact CSR is extremely unlikely to occur in a biological context. However, if such a distribution is expected, or conversely a completely clustered distribution, then the prior parameter p_B must be set accordingly. Keeping the prior set at 50%, we have found results to be robust with between 20% and 80% of molecules in clusters[11]. If the user wishes to statistically demonstrate clustering above CSR before setting the prior, we recommend other methods such as Ripley's K-function, discussed more extensively in the Alternative Approaches section. There are other prior parameters, such as the prior on the cluster radii (standard deviation) and the concentration parameter of the Dirichlet process, that can be altered. We have thoroughly tested our default choices on a wide variety of cluster scenarios, and found results to be largely insensitive to these parameters. However, we recognize there may be extreme examples, e.g. distributions with very large clusters, where these choices may need to be revisited. As with all Bayesian analyses, it should be remembered that the prior parameters should genuinely represent the analyst's (subjective) prior beliefs.

Statistical efficiency

Theoretically optimal estimates of cluster descriptors such as their radii, number of clusters per ROI would be calculated on the basis of a posterior sample, rather than the highest scoring cluster proposal. However, obtaining a posterior sample for this inference problem is known to be algorithmically difficult, due to the explosion of the space of possible solutions. Here, the highest scoring proposal is selected and we have shown that this gives higher accuracy than the current state-of-the-art, at a manageable computational cost.

Data limitations

It is well recognised in the field that fluorophores can undergo a process of multiple blinking[46-49]. This means that a single fluorescent molecule can generate multiple localisations in the resulting dataset. While some implementations of SMLM are more susceptible to this problem, it is likely that multiple blinking artefacts exist in all SMLM datasets. There are a number of methods which attempt to correct for multiple blinking, most of which have been implemented at the localisation stage of data analysis. Our algorithm assumes that the data has been pre-processed to remove such effects, and makes no attempt to be robust to the problem. Previously, when analysing experimental data, we have merged localisations which were estimated to have arisen from the same molecule using the method of Annibale et al., implemented in the Thunderstorm software[8, 11, 46].
As well as the problem of multiple blinking, the labelling efficiency, expression profile, detection efficiency and other sample parameters are frequently not known. Considering these potential artefacts, the number of localisations per region and the number of localisations per cluster cannot necessarily be directly equated to the number of real biological molecules present.

The algorithm assumes a rectangular ROI, and attempts to correct for edge effects. Because of this, best results are expected when the area is square, maximizing the ratio of area to perimeter. We recommend regions are chosen to be of the order 3 by 3 microns in size. The algorithm expects the background and clusters to be uniformly distributed over the rectangle, meaning that ROIs need to be selected carefully such that they do not intersect with cell boundaries. Additionally, the folding of the membrane at the cell boundary renders the 2D assumption of the analysis invalid.

We have tested the performance of the algorithm in detecting clusters in various conditions. For typical overall density of localisations, for example, 100-1000 per square micron, we find a lower detectability of six molecules per cluster. Our proposal generating algorithm is optimal in settings where the clusters are homogeneous in size within each ROI. This is due to the requirement of setting a single radius and threshold across the ROI. Therefore, regions that show extreme heterogeneity may be sub-optimally characterised.

Computational considerations

The computation time scales with the number of points, affecting how long it takes to score one proposal, and the number of cluster proposals, which is dependent on the range and increments of r and T. These are user-controllable settings, and therefore accuracy can be traded off against computation time. By default, the radius is varied over the range 5 to 300 in increments of 5, and the threshold is varied over 5 to 500 in increments of 5, resulting in 4000 proposals, which for a region containing 1000 points typically takes 30 minutes on a standard office desktop. Given typical limitations of processor power and memory, we recommend a maximum number of localisations per region of 15-20,000. The number of localisations in one SMLM acquisition are usually in the range 100,000 to 1,000,000. This means that typically, 5-10 regions should be selected to analyse the total area of a cell, resulting in computation times of several hours with default settings.

Sample preparation and data acquisition

In this section we will provide guidelines and key steps common to all SMLM sample and data preparation protocols, recognising that there are now a vast range of protocols available, each tailored to the study of specific biological processes. Ultimately, the goal is to generate a text file with x-y coordinates and associated localisation uncertainties, for input into our cluster analysis algorithm. For users, there are already published protocols on SMLM sample preparation and acquisition available[50-52]. In addition, Figure 1 shows a schematic of the overall workflow of the algorithm, illustrating where users can input their data if not following the preceding steps recommended here.

Two of the most common SMLM implementations are dSTORM[4, 5], which achieves the temporal separation of fluorophores using photoactivatable dyes, and PALM[1], which is based on photoswitchable proteins. Other SMLM implementations such as PAINT or transient binding also produce appropriate data for our algorithm[6]. Our algorithm is designed for clustering two dimensional data, for example as generated by the TIRF imaging configuration[53, 54] which PALM and dSTORM almost exclusively employ. The use of TIRF means that the maximal z-range of the data is in the region of 100-150 nm. Here, the acquisition results in the analysis of a 2D projection of this thin volume. For this reason we suggest to avoid cell edges where the membrane can turn perpendicular to the plane of imaging. Away from edges, projection artefacts are likely to be small as they follow a Cosine function with the angle of the membrane relative to the imaging plane.

More precisely, PALM imaging consists of stochastically switching the emission wavelength of a random subset of genetically encoded fluorescent proteins over time. mEos, PS-CFP2, Dronpa and Dendra are some of the routinely used photoactivatable fluorescent proteins, among many[47, 55-57]. In terms of sample preparation, cells are typically transfected with a plasmid encoding the fluorescent fusion construct around 24 hours before imaging. PALM therefore relies on the expression of exogenous plasmids, creating a subpopulation of the protein of interest tagged with a localisable fluorophore. The principal advantage of PALM over other SMLM imaging techniques is that the protein is directly imaged, without the

need for antibodies or permeabilisation. While not strictly required[58], cells are almost always fixed before imaging due to long acquisition times.

During the acquisition, two lasers must be used: one to facilitate the photoconversion process and one for conventional excitation of the converted form. The first laser is used to switch the emission wavelength of a small subset of fluorescent proteins and hence is typically used at very low powers. The second laser is then used to image and bleach this subset. A typical camera integration time is around 30 ms but this should be optimised for the particular system in use.

dSTORM[4, 5] relies on the immunostaining of the molecule of interest, avoiding cell transfection and over-expression of the molecule of interest, which is a common problem with PALM. The principle of the method is to target the molecule of interest with a primary antibody (with or without permeabilisation depending on the location of the studied protein) and to then target this primary antibody with a secondary antibody onto which the fluorophore is attached. The use of two antibodies decreases the resolution as together they can sum to an error of up to 40 nm in the estimate of the position of the protein of interest. Strategies have been developed to improve estimation accuracy include direct fluorophore conjugation of the primary antibody or the use of nanobodies[59]. The camera integration time is generally around 10 ms/frame due to higher photon counts (another advantage over PALM).

The raw data obtained by SMLM imaging consist of a sequence of raw frames containing diffraction limited point spread functions (PSFs) resulting from the emitting fluorophore subset. Software processing is then applied to localise the fluorophores in each frame and concatenate all localisations to reconstruct the final image. The result is a list of x-y coordinates with associated localisation uncertainties $(x_1, y_1, s_1), (x_2, y_2, s_2),\ldots$ Many software packages are available to perform this localisation process. For the remainder of this protocol, we assume the use of ThunderSTORM[8], a free and open source plugin for ImageJ.

Two common issues that arise from SMLM processing are the case of overlapping PSFs where a simple Gaussian distribution cannot be fitted to the complex intensity profile, and the multiple blinking phenomenon. ThunderSTORM offers the possibility to use multiple-emitter fitting (MEF), allowing for up to four overlapped PSFs. In the case of PALM, multiple blinking can be accounted for and corrected, by merging localisations in close spatial and temporal proximity.

ThunderSTORM allows the filtering of localisations according to a number of properties (e.g. localisation uncertainty)[8]. While our method takes explicit account of localisation uncertainty, meaning that such filters are not strictly required, they may help reduce processing times. We recommend the use of MEF for high PSF densities, drift correction, duplicate correction based on the uncertainty (an artefact associated with MEF), correction for multiple blinking in the case of PALM, and a photon count filter (above 2000 photons per localisation for Alexa 647 for instance). Note that ThunderSTORM requires, as input, several of the camera settings used during the acquisition process (often available in the user manual)[8]. Save the output as .csv files, keeping all descriptive parameters.

Materials

- ImageJ (http://imagej.nih.gov/ij/) with downloaded and installed plugins "Grid" (http://rsb.info.nih.gov/ij/plugins/grid.html) and ThunderSTORM (https://code.google.com/p/thunder-storm/)[8]
- R (https://cran.r-project.org/) and RStudio (https://www.rstudio.com/). Note that the code only requires the free, open source RStudio version.
- Two additional R libraries, "splancs" and "igraph" which can be installed directly from the RStudio interface via Tools>Install Packages.
- (Optional) Matlab (http://uk.mathworks.com/products/matlab/?refresh=true) with version no earlier than 2014b.

Procedure

1. In order to start analysis of an experimental dataset, proceed with option A. A simulated dataset can also be prepared for analysis by following option B.

(A) Formatting processed experimental data sets for analysis and defining regions of interest (ROI). Timing 1 hr

i. Create a parent folder in which all files associated with this analysis will be kept. In our example, we will call this parent folder "Condition i".
ii. Copy the files provided for the analysis (formatting.R, get_histograms.m, run.R, internal.R, postprocessing.R, simulate.R, formatting_params.txt, sim_params.txt, Coord.txt and config.txt files) into this folder.
iii. Open ImageJ.
iv. Click on "Plugins>ThunderSTORM>Import/Export>Import Results". A navigation window will appear.
v. Navigate in "file path" to the .csv file and select it. Make sure "live preview" is selected, keeping all other defaults, and click OK. Both the list of localisations and their visualisation will appear (Figure 2). In the supplementary information, we provide an example experimental .csv file for users to test their analysis procedure.
TROUBLESHOOTING
vi. (Optional) Perform any required filtering and post-processing (e.g. drift correction) using ThunderSTORM[8].
vii. Click on "Export". A dialog box will appear.
viii. Select .csv as the format for the file extension.
ix. Indicate the path to the parent folder, "Condition i" in our example. Give a numerical name to the file for further steps, e.g. 1.csv.
x. Tick only x, y and uncertainty boxes to indicate which columns to save.
xi. Click on OK.

xii. Critical Step: Verify that the parent folder now contains a file called 1.csv which contains 3 columns: x coordinates, y coordinates and uncertainty.

TROUBLESHOOTING

xiii. Click on the visualisation window (Figure 2).

xiv. In the ImageJ main interface, click on "plugins>Grid". A dialog box will appear.

xv. Type in the "Area" box the desired area of the ROI, for example, '2' for a 2 μm by 2 μm square ROI. Note that processing time is dependent on the number of points in the ROI. We recommend that the size of each resulting .csv file to be <250kb (representing roughly 15-20,000 localisations). The size of the ROI should therefore be adjusted depending on the density of localisations in the sample. We have found that a ROI of 2 μm x 2 μm to 3 μm x 3 μm is adequate. As the cluster analysis framework assumes homogeneous clustering in the x-y plane, if the sample displays large scale heterogeneity, a larger number of smaller regions may better generate locally homogeneous ROIs. However, the algorithm is fully robust to any ROI size and therefore very large regions can be selected by the user, depending on the constraints of computational time. Note that rectangular ROIs are also possible.

xvi. Click "Ok". A grid will appear over the visualisation (Figure 2).

xvii. Select the pointer icon in the ImageJ main interface (Figure 2).

xviii. Place the cursor in the middle of a grid square thus defining the centre of a ROI.

xix. Look for the corresponding coordinates of this point on the main ImageJ interface (Figure 2). The coordinates will be in μm. Avoid any grid squares which contain the boundary of a cell, as these will give sub-optimal cluster results. In addition, if the cellular sub-region of interest is small, smaller ROIs should be selected in order that the ROI illustrates the specific cluster characteristics of that sub-region. Note that there are other ImageJ plugins that may be useful for selecting regions and these can be used at the user's discretion[42].

xx. Type the coordinates of the centres of each ROI in the Coord.txt file provided, in the following format:

a. First column: the name (numerical value) of the .csv file containing the list of localisations from which the ROIs are extracted e.g. 1 for 1.csv
b. Second and third columns: the coordinates (x and y respectively).
c. Columns should be separated by a Tab symbol.

xxi. Repeat steps 1iv to 2viii for all other data sets of the same condition.

xxii. Save and close Coord.txt.

xxiii. Open RStudio (Figure 3).

xxiv. Critical Step: Set the correct working directory. In the console interface type: setwd("path to parent folder"). Under Windows, occurrences of the backslash, "\", in the path name must be replaced by forward slashes, "/". For example: setwd("C:/Users/Owen/Desktop/Condition i") and press Return.

xxv. Select the open icon in the RStudio interface and open formatting.R. The code will appear in the RStudio interface (Figure 3).

xxvi. Specify the name of the folder where the experimental data ROI subfolders will be stored (line 1), by default the folder name is set as "ROIs".

xxvii. Open the formatting_params.txt file contained in the parent folder using a standard text editor such as Notepad. Formatting parameters are stored in this file and can be modified.

xxviii. Critical Step: Enter the size of the region of interest in x and y in nanometres (Lines 1 and 2 of the code).

xxix. Critical Step: Enter the columns within the raw data (e.g. 1.csv) in which the x (col_x, line 3), y (col_y, line 4)and localisation precisions (col_unc_xy, line 5)

xxx. Save and close formatting_params.txt

xxxi. Click on the "source" icon in RStudio (Figure 3).

xxxii. Critical Step: Verify that the code has created a subfolder called ROIs within the Parent Folder and that ROIs contains subfolders sequentially labelled from 1 to the total number of ROIs as well as the copied config.txt. Each contains a text file called data.txt containing the x, y and uncertainty values (Figure 4). The coordinates of the ROI will have been reset to begin at coordinate 0,0 and therefore if the user is not using formatting.R, the coordinates must also be reset to the origin.
TROUBLESHOOTING

B. Generating simulated data sets Timing 10 min

i. Create a parent folder in which all files associated with this analysis will be kept. In our example, we will call this parent folder "Condition i".

ii. Copy the files provided for the analysis (formatting.R, get_histograms.m, formatting_params.txt, run.R, internal.R, postprocessing.R, simulate.R, sim_params.txt, Coord.txt and config.txt files) into this folder.

iii. Open RStudio (Figure 3).

iv. Critical Step: Set the correct working directory. In the console interface type: setwd("path to parent folder"). Under Windows, occurrences of the backslash, "\", in the path name must be replaced by forward slashes, "/". For example:
setwd("C:/Users/Owen/Desktop/Condition i") and press Return.

v. Select the open icon in the RStudio interface and open simulate.R. The code will appear in the RStudio interface (Figure 3).

vi. Specify the name of the folder where the simulated ROI subfolders will be stored (line 1), by default the folder name is set as "ROIs".

vii. Open sim_params.txt in a standard text editor, such as Notepad. User definable parameters are stored in this file and can be modified.

viii. Enter the desired number of localisations per cluster (line 1: molspercluster, default=100). Note that by setting this parameter = 1, a completely spatially random data set will be generated, which can be used as a control for experimental conditions.

ix. Enter the desired fraction of localisations in the background (line 2: background, default=.5).

x. Enter the desired number of clusters per ROI (line 3: nclusters, default=10)

xi. Enter the desired x and y limits of the ROI in nm (lines 4-5: xlim, ylim, default=0,3000 for both).

xii. Enter the desired parameters for the Gamma distribution. This is the distribution from which the localisation precisions will be generated. The parameters are given in shape α, rate β format. The mean of this distribution is α/β and the variance is α/β^2 (line 6: gammaparams, default=5, 0.166667 which generates a mean simulated localisation precision of 30 nm).

xiii. Enter in the desired number of independent ROIs to be simulated (line 7: nsim, default=2).

xiv. Enter the desired cluster radii (defined as the standard deviation of the positions of the points within a cluster, before scrambling by the localisation uncertainties) in nm (line 8: sdcluster, default=50).

xv. (Optional) This will save simulated parameters representing a standard clustering scenario representing randomly positioned circular Gaussian clusters overlaid with a CSR background. Save and close this text file here if this standard configuration is desired. Options for scenarios with uneven background, multimerisation and unequal cluster sizes are given below.

xvi. (Optional) If a variety of cluster radii are desired, enter a list of radii in line 8 (separated by commas). The list should contain the same number of entries as the number of clusters (specified in line 3).

xvii. (Optional) Enter the desired multimerisation. If this parameter is a positive integer *m* larger than one, then previous clustering parameters are ignored. Points are distributed completely spatially randomly on the background, and replicated m times. Every point is scrambled by an independent localisation precision drawn from the Gamma distribution described above. For example, to simulate dimers, *m*=2 (line 9: multimerisation, default=0). In the case of multimerisation, enter the desired number of distinct molecular positions in the ROI (line 10: mols_for_multimer_case, default=2000). Finally, specify the desired fraction of the molecules to be simulated in a multimerised state (line 11: propmultimered, default=.1). To illustrate, the total number of localisations in the ROI should be around mols_for_multimer_case + (mols_for_multimer_case x propmultimered) x (multimerisation -1)) (the rare molecules that are simulated outside the ROI due to the scrambling process are deleted).

xviii. (Optional) Enter the desired background distribution if the default of CSR is not desired. Background localisations are distributed uniformly in the y dimension, but according to a Beta distribution in the x dimension, with specified parameters a and b. The uniform distribution (CSR) is achieved with a=1 and b=1. The choice a=b=2, for example, induces a moderate increase in background density in the centre of the ROI, whereas a=5,b=1 induces an extreme increase in background density at the right side of the ROI. (line 12: ab, default=1,1)

xix. Save and close the text file.

xx. Critical Step: In RStudio, click on “Source”. This will create a folder with subfolders containing the simulated ROIs in the specified path (Figure 4). Verify that the code has created a subfolder called ROIs within the Parent Folder and that ROIs contains subfolders sequentially labelled from 1 to the total number of ROIs. Each contains a text file called data.txt containing the x, y, localisation uncertainty and cluster label values (Figure 4). Clusters are labelled sequentially up to the number of clusters. Unclustered localisations have a unique label (singletons).
TROUBLESHOOTING

Setting up the config.txt for analysis Timing 5 min

2. Open the config.txt file contained in “ROIs” (not the version which exists in the parent folder which will always retain the default analysis parameters), using a standard text editor such as Notepad. Analysis parameters are stored in this file and can be modified.

3. Critical Step: enter the ROI limits in nm (lines 2-3: xlim, ylim, default=0,3000 for both). The size of the ROI should always match the size specified in section 2 iii.
4. Critical Step: default parameters provided below are appropriate for cluster analysis in micron-sized regions in which 20-80% of localisations are in clusters, and the radius of a cluster is expected to fall in the range 10-500 nm. If these parameters are not expected to be appropriate for the data under analysis, then modify the values as detailed below.
5. (Optional) Enter the prior probability distribution on the cluster radii (defined as the standard deviation of the positions of the points within a cluster) in a histogram format, giving first the bin locations in nm (line 4: histbins) and then the bin frequencies (line 5: histvalues). The code converts the histogram into a probability density function by linear interpolation followed by re-standardization. For example, setting histbins=1,100 and histvalues=1,1 is equivalent to assuming a uniform prior distribution between 1 nm and 100 nm: *a priori*, each cluster could have a standard deviation between 1 nm and 100 nm, and any value within that range is equally probable.
6. (Optional) Set the concentration coefficient of the Dirichlet process (line 6: alpha, default 20). This prior affects how clustered points are assumed to organise into groups. Lower (higher) values induce fewer (more) clusters.
7. (Optional) Set the prior on the proportion of localisations in the background (line 7: pbackground, default=.5). The default has been shown to be robust to true proportions between 0.2 and 0.8[11].
8. (Optional) Set the range and increment of radii (rseq) and threshold (thseq) to be used to generate cluster proposals in the format min, max, increment (lines 8-9: rseq, default 5,300,5, thseq, default 5,500,5). The minimum should always be set below the minimum possible expected cluster radius and the maximum set to be above the expected maximum possible cluster radius. Increasing the overall range increases processing time accordingly. Lower values of increment increase the accuracy of the generated cluster proposals at the expense of computational speed.
9. (Optional) Set the value of makeplot (1=true, 0=false). This parameter determines whether cluster maps will be generated (as opposed to simply extracting cluster descriptors) for each ROI (line 10: default 1).
10. (Optional) Set the value of superplot (1=true, 0=false). This parameter determines whether a montage of the cluster maps for each ROI will be generated as a single .pdf (line 11: default 1).
11. (Optional) Set the value of skeleton (1=true, 0=false). This parameter determines whether a copy of the ROIs folder will be generated (as R_ROIs) which will only retain the highest scoring label proposal (for ease of storage and transport) (line 12: default 0).
12. Save and close config.txt.

Running the cluster analysis. Timing 45 min per ROI

13. Open RStudio (Figure 3).
14. Critical Step: Set the correct working directory. In the console interface type: setwd("path to parent folder"). Under Windows, occurrences of the backslash, "\", in the path name must be replaced by forward slashes, "/". For example:
setwd("C:/Users/Owen/Desktop/Condition i") and press Return.

15. Select the open icon in the RStudio interface and open run.R. The code will appear in your RStudio interface (Figure 3).
16. On line 2 of run.R, set the folder name to match the folder containing the regions of interest. In our example “ROIs”. Additional folders can be added within the parent folder (e.g. ROIs2, ROIs3). If this is the case, these can be listed on line 2, separated by commas, in order to allow batch processing e.g. foldernames=c(“ROIs”, "ROIs2", “ROIs3”).
17. Critical Step: Click on Source to launch the analysis (Figure 3). The progress of the analysis will appear on the console section of your RStudio interface. Once “>” appears in the console, the analysis is complete. Note that the processing time depends on the number of ROIs as well as the size of the data.txt files associated with each ROI. In each numbered subfolder within ROIs, the code will have generated a text file called r_vs_thresh.txt and a new subfolder called “labels” containing all the tested cluster proposals.
    TROUBLESHOOTING

Post-processing. Timing 5 min

18. Click on the open icon of the RStudio interface and select postprocessing.R.
19. On line 2 of postprocessing.R, set the folder name to match the folder containing the regions of interest. In our example “ROIs”. Additional folders can be added within the parent folder (e.g. ROIs2, ROIs3). If this is the case, these can be listed on line 2, separated by commas, in order to allow batch processing e.g. foldernames=c(“ROIs”, "ROIs2", “ROIs3”).
20. Click on “Source” to run the code (Figure 3). The code extracts the best proposal, obtaining key cluster descriptors (radii of the clusters: radii.txt, number of localisations per cluster: nmols.txt, number of clusters per ROI: nclusters.txt, percentage of localisations in clusters per ROI: pclustered.txt, total number of localisation per ROI: totalmols.txt and relative density: reldensity.txt) and saving them in the ROIs folder, along with .pdfs containing a histogram of each descriptor and, if specified, the superplot (Figure 5). The relative density is defined as the density of localisations within clusters (localistions per square micron) divided by the density outside of clusters. Note that in terms of biological interpretation, the number of localisations per cluster and per region cannot necessarily be directly equated to the real number of molecules due to the problems of labelling efficiency, endogenous protein expression, detection efficiency and fluorophore multiple blinking. Nevertheless, a modification of these descriptors between two conditions illustrates a relative change in clustering. The code saves summary information about each specific region, including the best scoring proposal (e.g. in “ROIs/1/summary.txt” (Skeleton=0) or “R_ROIs/1/summary.txt” (Skeleton=1) in our example) and a .pdf image (e.g. in “ROIs/labels/1/plot.pdf in our example) of the resulting cluster map (Figure 5). If simulated data was analysed, the .pdf file will contain two cluster maps, the first corresponding to the true, simulated labelling and the second to the analysed labelling. Finally, the code saves the x, y positions of each detected cluster (cluster_statistics.txt). If the user wishes to analyse multi-scale clustering (cluster of clusters), the analysis could be re-run on the cluster centres, for example using Ripley’s K-function[30, 33].
    TROUBLESHOOTING

(optional) Displaying cluster descriptors Timing 10 min

The Get_histograms.m code provided allows the generation of user definable histograms of all six key cluster descriptors, saved as .fig files to facilitate visualisation and interpretation of the data. This step is purely aesthetic.

21. Open Matlab.
22. Click on the "current folder" icon. Select the parent folder ("Condition i" folder in our example) (Figure 6). The files contained in this folder, including the .m files, will appear in the current folder section (on the left of the Matlab interface).
23. Double click on the Get_histograms.m file to open the function in the editor.
24. Enter the folder name containing the postprocessed data on line 4 (ROIs in our example)
25. Enter the bin width which will be used to generate the histograms of each descriptor (lines 7-12).
26. Click Run. The code will generate formatted histograms of each descriptor.
    TROUBLESHOOTING

Timing

- Selecting and formatting regions of interest: 60 mins (for 30 regions)
- Preparing analysis: 5 mins
- Analysis processing time 45 mins (per region)
- Post-processing and visualisation: 15 mins

Troubleshooting

Steps	Problem	Possible reason	Solution
1A.v	Localisations within the live preview are very sparse or there are significant background localisations outside of the cell	Low transfection efficiency or expression of the fluorescent construct, low labelling efficiency of the antibody or high levels of non-specific binding	Optimise transfection protocol or immunolabelling procedure to produce higher signal and lower or lower background
1A. xii	.csv data file does not exist in the parent folder	Incorrect path was entered when exporting the .csv file	Export again with correct path to the parent folder
1A. xxxii Or 1B. xx	When running formatting.R, the correct files are not created in the folder structure (Figure 4)	Either Coord.txt or config.txt are missing from the parent folder.	Make sure config.txt and Coord.txt are in the parent folder. Ensure the working directory has been set correctly

1A. xxxii Or 1B. xx	When running formatting.R, The code did not generate a subfolder within the parent folder containing -regions of interest or the data.txt files do not contain localisation coordinates or uncertainties.	The Coord.txt file is empty or formatted incorrectly, or the regions did not contain any localisations, or incorrect columns have been exported into the .csv.	Check that the selected regions of interest contain localisations and that the coordinates of the centre of each ROI have been saved in a correctly formatted Coord.txt file. Also ensure that only the x,y and uncertainty columns were exported into the .csv file.
17	When run.R is launched, the error message "cannot open file 'internal.R': No such file or directory" appears in the RStudio console interface .	The wrong working directory has been set in RStudio.	Set the correct working directory in the Console interface by typing setwd("path to parent folder") and press Return. Under Windows, occurrences of the backslash, "\", in the path name must be replaced by forward slashes, "/".
17	When run.R is launched, the error message "cannot open file '…config.txt': No such file or directory" appears in the RStudio console interface.	The target folder of run.R (line 2) has been set incorrectly.	Specify the name of the folder where the simulated ROI subfolders are stored (line 2 of run.R). This folder must match the name of the folder in which the regions of interest are stored.
17	During processing, an error message appears indicating an out of memory error.	The allocated memory limit has been reached.	Make sure your data.txt files are under 250kb. This can be achieved by further filtering in ThunderSTORM (e.g. by photon count), or specifying smaller regions of interest in formatting_params.txt and config.txt.
20 And 26	The cluster maps fail to find obvious clusters or detect clusters in visually random point patterns.	1. Priors or 2. the range of radius and threshold used to generate proposals were not set appropriately, or 3. The spatial configuration of the localisations does not conform closely to the assumed Bayesian model of circular Gaussian clusters overlaid with a CSR background.	1. If a visual inspection of the generated cluster map reveals extreme clustering or near-CSR, the prior on pbackground can be modified accordingly in config.txt and the analysis re-run. If there are extremely small or large clusters the prior histogram on the size distribution can be modified in config.txt. 2. Increase the range of radius and threshold (note processing time

			will increase). 3. The data is not appropriate for this analysis method.
20 And 26	Interpretation of resulting cluster maps is problematic due to issues surrounding labelling, sample preparation and imaging conditions	Non-specific antibody binding, artefacts from permeabilisation, protein over-expression, fluorophore multiple-blinking	Repeat experiments using different antibodies, fluorophore or fluorescent proteins and check that clustering behaviour is consistent

Expected Outcomes

Figure 7 shows expected outcomes from our cluster analysis. In this case, the data was generated by dSTORM of ZAP-70 in primary human T cells forming a T cell immunological synapse[60] on activating anti-CD3 and anti-CD28 coated glass coverslips as previously described for super-resolution[11, 14, 15, 61, 62]. Cells were left to form synapses, fixed and then immunostained with primary and secondary antibodies labelled with Alexa-647 and imaged in a standard dSTORM buffer. Images were acquired on a Nikon N-STORM microscope operated in a TIRF configuration and pre-processed using ThunderSTORM as described, including MEF.

Figure 7a shows a representative reconstructed image from which ROIs were selected. The data set for this example analysis is available for download as Supplementary Information. The image shows the approximate localization density and level of background to be suitable for cluster analysis. Figure 7b shows an example of the generated cluster maps where each cluster is pseudo-coloured in an arbitrary colour. Figures 7c-h shows generated histograms for the 6 key cluster descriptors, with the means indicated by the dashed lines. The number of localisations per cluster and per region is only semi-quantitatively interpreted as the number of true molecules due to the unknown labelling efficiency, detection efficiency and multiple-blinking. Unclustered localisations are true localisations that are not clustered and should not be interpreted as experimental background noise. The cluster radius is defined as the standard deviation of the positions of the localisations associated with that cluster.

In this example, the output indicates that ZAP-70 is clustered at the membrane at the T cell immunological synapse. It is well known that many signaling molecules cluster following stimulation through the T-cell receptor pathway. Other such examples include the TCR itself, Lck, LAT and SLP-76. Studies have shown that clustering can digitize cell signaling.

Mechanisms for clustering are hypothesized to be protein-protein interactions, docking at newly available phosphorylated sites, membrane lipid microdomains or interactions with the dynamic cortical actin meshwork.

Figure Captions

Figure 1: Schematic of the Bayesian cluster analysis workflow. The schematic indicates the required data formats and inputs for each stage from localization, through data formatting and cluster analysis to post-processing and visualization.

Figure 2: **Schematic of the ImageJ interface while displaying the output of ThunderSTORM**. The visualization window and controls required to select regions of interest for analysis have been highlighted.

Figure 3: Schematic of the RStudio interface. Here, the main processing program, run.R, is open and ready for analysis. Major controls and windows are highlighted.

Figure 4: **Format of the folder structure ready for the Bayesian cluster analysis algorithm**. This the folder and file structure after the running of formatting.R to define ROIs for experimental data. The parent folder (Condition i), contains (in this example) 5 sequentially named SMLM data sets, the user-generated Coord.txt file to specify which regions will be analysed and all the other necessary, supplied files. A subfolder (in this example "ROIs") in turn contains a folder for each of the individual regions to be analysed, with each folder containing a file, data.txt, of the localisation coordinates and associated uncertainties.

Figure 5: Format of the folder and file structure after processing and post-processing are complete.. The ROIs folder now contains text files listing the aggregated cluster descriptors from all analysed regions as well as basic histograms of these parameters. Each individual region folder also contains a generated cluster map (plot.pdf), and two a files, summary.txt and cluster-statistics.txt, which list key cluster descriptors for that specific region. The folder "labels" contains every tested cluster proposal and r_vs_thresh.txt contains the full set of scores for every tested cluster proposal. The highest scoring proposal is given in summary.txt. Schematic is in the case of skeleton=0 (FALSE).

Figure 6. Schematic of the Matlab interface for generating optional histograms of the output data. The open and Run icons have been highlighted, along with the code itself.

Figure 7: **Expected outcome of the analysis algorithm when used on experimental dSTORM data.** Cluster analysis of ZAP-70 at the T cell immunological synapse imaged by dSTORM, from a total of n = 28 selected regions from n = 12 cells. a) reconstructed

ThunderSTORM image of a representative cell. b) representative 3 x 3 μm cluster map generated by our algorithm. c) histogram of the number localisations per cluster. d) histogram of the radii of the detected clusters. e) histogram of the percentage of localisations found in clusters for each ROI. f) histogram of the relative density of molecules inside clusters as compared to outside for each ROI. g) histograms of the number of detected clusters per ROI. h) histogram of the total number of localisations per ROI.

Acknowledgements

D.M.O. acknowledges funding from the European Research Council (FP7 starter grant 337187) and Marie Curie Career Integration grant 334303. A.P.C. is funded by Arthritis Research UK grants 19652 and 20525. We acknowledge the use of the King's College Nikon Imaging Centre (NIC).

Author Contributions

JG, MS, CLB, LB, NAH, DMO and PR-D developed and tested the protocol. JG and PR-D wrote the analysis code. GLB, DJW and APC provided samples. JG, MS, GLB, DMO and PR-D performed imaging, simulations and analysis. JG, DMO and PR-D wrote the manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

References

1. Betzig, E., Patterson, G.H., Sougrat, R., Lindwasser, O.W., Olenych, S., Bonifacino, J.S., Davidson, M.W., Lippincott-Schwartz, J. & Hess, H.F. Imaging Intracellular Fluorescent Proteins at Nanometer Resolution. *Science* **313**, 1642-1645 (2006).
2. Hess, S.T., Girirajan, T.P.K. & Mason, M.D. Ultra-High Resolution Imaging by Fluorescence Photoactivation Localization Microscopy. *Biophys. J.* **91**, 4258-4272 (2006).
3. Rust, M.J., Bates, M. & Zhuang, X. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nat. Meth.* **3**, 793-796 (2006).
4. Heilemann, M., van de Linde, S., Schüttpelz, M., Kasper, R., Seefeldt, B., Mukherjee, A., Tinnefeld, P. & Sauer, M. Subdiffraction-Resolution Fluorescence Imaging with Conventional Fluorescent Probes. *Angew. Chem. Int. Ed.* **47**, 6172-6176 (2008).
5. Heilemann, M., van de Linde, S., Mukherjee, A. & Sauer, M. Super-Resolution Imaging with Small Organic Fluorophores. *Angew. Chem. Int. Ed.* **48**, 6903-6908 (2009).
6. Kiuchi, T., Higuchi, M., Takamura, A., Maruoka, M. & Watanabe, N. Multitarget super-resolution microscopy with high-density labeling by exchangeable probes. *Nat. Meth.* **12**, 743-746 (2015).
7. Henriques, R., Lelek, M., Fornasiero, E.F., Valtorta, F., Zimmer, C. & Mhlanga, M.M. QuickPALM: 3D real-time photoactivation nanoscopy image processing in ImageJ. *Nat. Meth.* **7**, 339-340 (2010).
8. Ovesný, M., Křížek, P., Borkovec, J., Švindrych, Z. & Hagen, G.M. ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging. *Bioinformatics* **30**, 2389-2390 (2014).

9. Thompson, R.E., Larson, D.R. & Webb, W.W. Precise Nanometer Localization Analysis for Individual Fluorescent Probes. *Biophys. J.* **82**, 2775–2783 (2002).
10. Quan, T., Zeng, S. & Huang, Z.-L. Localization capability and limitation of electron-multiplying charge-coupled, scientific complementary metal-oxide semiconductor, and charge-coupled devices for superresolution imaging. *J. Biomed. Opt.* **15**, 066005-066005-066006 (2010).
11. Rubin-Delanchy, P., Burn, G.L., Griffie, J., Williamson, D.J., Heard, N.A., Cope, A.P. & Owen, D.M. Bayesian cluster identification in single-molecule localization microscopy data. *Nat. Meth.* **12**, 1072-1076 (2015).
12. Wolter, S., Loschberger, A., Holm, T., Aufmkolk, S., Dabauvalle, M.-C., van de Linde, S. & Sauer, M. rapidSTORM: accurate, fast open-source software for localization microscopy. *Nat. Meth.* **9**, 1040-1041 (2012).
13. Sage, D., Kirshner, H., Pengo, T., Stuurman, N., Min, J., Manley, S. & Unser, M. Quantitative evaluation of software packages for single-molecule localization microscopy. *Nat. Meth.* **12**, 717-724 (2015).
14. Williamson, D.J., Owen, D.M., Rossy, J., Magenau, A., Wehrmann, M., Gooding, J.J. & Gaus, K. Pre-existing clusters of the adaptor Lat do not participate in early T cell signaling events. *Nat. Immunol.* **12**, 655-662 (2011).
15. Sherman, E., Barr, V., Manley, S., Patterson, G., Balagopalan, L., Akpan, I., Regan, Carole K., Merrill, Robert K., Sommers, Connie L., Lippincott-Schwartz, J. & Samelson, Lawrence E. Functional Nanoscale Organization of Signaling Molecules Downstream of the T Cell Antigen Receptor. *Immunity* **35**, 705-720 (2011).
16. Garcia-Parajo, M.F., Cambi, A., Torreno-Pina, J.A., Thompson, N. & Jacobson, K. Nanoclustering as a dominant feature of plasma membrane organization. *J. Cell Sci.* **127**, 4995-5005 (2014).
17. Pageon, S.V., Cordoba, S.-P., Owen, D.M., Rothery, S.M., Oszmiana, A. & Davis, D.M. Superresolution Microscopy Reveals Nanometer-Scale Reorganization of Inhibitory Natural Killer Cell Receptors upon Activation of NKG2D. *Sci. Signal.* **6**, ra62- (2013).
18. Lin, W.-C., Iversen, L., Tu, H.-L., Rhodes, C., Christensen, S.M., Iwig, J.S., Hansen, S.D., Huang, W.Y.C. & Groves, J.T. H-Ras forms dimers on membrane surfaces via a protein–protein interface. *Proc. Natl. Acad. Sci.* **111**, 2996-3001 (2014).
19. Lewitzky, M., Simister, P.C. & Feller, S.M. Beyond 'furballs' and 'dumpling soups' – towards a molecular architecture of signaling complexes and networks. *FEBS Lett.* **586**, 2740-2750 (2012).
20. Kenworthy, A.K. Nanoclusters digitize Ras signalling. *Nat. Cell Biol.* **9**, 875-877 (2007).
21. Roob, E., III, Trendel, N., Rein ten Wolde, P. & Mugler, A. Cooperative Clustering Digitizes Biochemical Signaling and Enhances its Fidelity. *Biophys. J.* **110**, 1661-1669 (2016).
22. Stone, M.B. & Veatch, S.L. Steady-state cross-correlations for live two-colour super-resolution localization data sets. *Nat. Commun.* **6** (2015).
23. Simons, K. & Ikonen, E. Functional rafts in cell membranes. *Nature* **387**, 569-572 (1997).
24. Brown, D.A. Lipid Rafts, Detergent-Resistant Membranes, and Raft Targeting Signals. *Physiology* **21**, 430-439 (2006).
25. Owen, D.M., Williamson, D.J., Magenau, A. & Gaus, K. Sub-resolution lipid domains exist in the plasma membrane and regulate protein diffusion and distribution. *Nat. Commun.* **3**, 1256 (2012).

26. Lingwood, D. & Simons, K. Lipid Rafts As a Membrane-Organizing Principle. *Science* **327**, 46-50 (2010).
27. Gowrishankar, K., Ghosh, S., Saha, S., C, R., Mayor, S. & Rao, M. Active Remodeling of Cortical Actin Regulates Spatiotemporal Organization of Cell Surface Molecules. *Cell* **149**, 1353-1367 (2012).
28. Goswami, D., Gowrishankar, K., Bilgrami, S., Ghosh, S., Raghupathy, R., Chadda, R., Vishwakarma, R., Rao, M. & Mayor, S. Nanoclusters of GPI-Anchored Proteins Are Formed by Cortical Actin-Driven Activity. *Cell* **135**, 1085-1097 (2008).
29. Köster, D.V., Husain, K., Iljazi, E., Bhat, A., Bieling, P., Mullins, R.D., Rao, M. & Mayor, S. Actomyosin dynamics drive local membrane component organization in an in vitro active composite layer. (2016).
30. Ripley, B.D. Modelling spatial patterns. *J. R. Stat. Soc. Series B Stat. Methodol.* **39**, 172-192 (1977).
31. Getis, A. & Franklin, J. Second-Order Neighborhood Analysis of Mapped Point Patterns. *Ecology* **68**, 473-477 (1987).
32. Juette, M.F., Terry, D.S., Wasserman, M.R., Altman, R.B., Zhou, Z., Zhao, H. & Blanchard, S.C. Single-molecule imaging of non-equilibrium molecular ensembles on the millisecond timescale. *Nat. Meth.* **13**, 341-344 (2016).
33. Owen, D.M., Rentero, C., Rossy, J., Magenau, A., Williamson, D., Rodriguez, M. & Gaus, K. PALM imaging and cluster analysis of protein heterogeneity at the cell surface. *J. Biophoton.* **3**, 446-454 (2010).
34. Lagache, T., Lang, G., Sauvonnet, N. & Olivo-Marin, J.-C. Analysis of the Spatial Organization of Molecules with Robust Statistics. *PLoS ONE* **8**, e80914 (2013).
35. Sengupta, P., Jovanovic-Talisman, T., Skoko, D., Renz, M., Veatch, S.L. & Lippincott-Schwartz, J. Probing protein heterogeneity in the plasma membrane using PALM and pair correlation analysis. *Nat. Meth.* **8**, 969-975 (2011).
36. Sengupta, P. & Lippincott-Schwartz, J. Quantitative analysis of photoactivated localization microscopy (PALM) datasets using pair-correlation analysis. *BioEssays* **34**, 396-405 (2012).
37. Veatch, S.L., Machta, B.B., Shelby, S.A., Chiang, E.N., Holowka, D.A. & Baird, B.A. Correlation Functions Quantify Super-Resolution Images and Estimate Apparent Clustering Due to Over-Counting. *PLoS ONE* **7**, e31457 (2012).
38. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *KDD-96 Proceedings*, 226-231 (1996).
39. Dudok, B. et al. Cell-specific STORM super-resolution imaging reveals nanoscale organization of cannabinoid signaling. **18**, 75-86 (2015).
40. Levet, F., Hosy, E., Kechkar, A., Butler, C., Beghin, A., Choquet, D. & Sibarita, J.-B. SR-Tesseler: a method to segment and quantify localization-based super-resolution microscopy data. *Nat. Meth.* **12**, 1065-1071 (2015).
41. Caetano, F.A., Dirk, B.S., Tam, J.H.K., Cavanagh, P.C., Goiko, M., Ferguson, S.S.G., Pasternak, S.H., Dikeakos, J.D., de Bruyn, J.R. & Heit, B. MIiSR: Molecular Interactions in Super-Resolution Imaging Enables the Analysis of Protein Interactions, Dynamics and Formation of Multi-protein Structures. *PLoS Comput. Biol.* **11**, e1004634 (2015).
42. Barna, L., Dudok, B., Miczan, V., Horvath, A., Laszlo, Z.I. & Katona, I. Correlated confocal and super-resolution imaging by VividSTORM. *Nat. Protocols* **11**, 163-183 (2016).

43. Malkusch, S., Endesfelder, U., Mondry, J., Gelléri, M., Verveer, P. & Heilemann, M. Coordinate-based colocalization analysis of single-molecule localization microscopy data. *Hist. Cell. Biol.* **137**, 1-10 (2012).
44. Rossy, J., Cohen, E., Gaus, K. & Owen, D.M. Method for co-cluster analysis in multichannel single-molecule localisation data. *Hist. Cell Biol.* **141**, 605-612 (2014).
45. Owen, D.M., Williamson, J., Boelen, L., Magenau, A., Rossy, J. & Gaus, K. Quantitative Analysis of Three-Dimensional Fluorescence Localization Microscopy Data. *Biophys. J.* **105**, L05-L07 (2013).
46. Annibale, P., Vanni, S., Scarselli, M., Rothlisberger, U. & Radenovic, A. Quantitative Photo Activated Localization Microscopy: Unraveling the Effects of Photoblinking. *PLoS ONE* **6**, e22678 (2011).
47. Lee, S.-H., Shin, J.Y., Lee, A. & Bustamante, C. Counting single photoactivatable fluorescent molecules by photoactivated localization microscopy (PALM). *Proc. Natl. Acad. Sci.* **109**, 17436-17441 (2012).
48. Dempsey, G.T., Vaughan, J.C., Chen, K.H., Bates, M. & Zhuang, X. Evaluation of fluorophores for optimal performance in localization-based super-resolution imaging. *Nat. Meth.* **8**, 1027-1036 (2011).
49. Annibale, P., Scarselli, M., Kodiyan, A. & Radenovic, A. Photoactivatable Fluorescent Protein mEos2 Displays Repeated Photoactivation after a Long-Lived Dark State in the Red Photoconverted Form. *J. Phys. Chem. Lett.* **1**, 1506-1510 (2010).
50. Gould, T.J., Verkhusha, V.V. & Hess, S.T. Imaging biological structures with fluorescence photoactivation localization microscopy. *Nat. Protocols* **4**, 291-308 (2009).
51. van de Linde, S., Loschberger, A., Klein, T., Heidbreder, M., Wolter, S., Heilemann, M. & Sauer, M. Direct stochastic optical reconstruction microscopy with standard fluorescent probes. *Nat. Protocols* **6**, 991-1009 (2011).
52. Sengupta, P., Jovanovic-Talisman, T. & Lippincott-Schwartz, J. Quantifying spatial organization in point-localization superresolution images using pair correlation analysis. *Nat. Protocols* **8**, 345-354 (2013).
53. Axelrod, D. Cell surface contacts illuminated by total internal refection fluorescence. *J. Cell Biol.* **89**, 141-145 (1981).
54. Axelrod, D. Total internal reflection fluorescence microscopy in cell biology. *Traffic* **2**, 764-774 (2001).
55. Wang, S., Moffitt, J.R., Dempsey, G.T., Xie, X.S. & Zhuang, X. Characterization and development of photoactivatable fluorescent proteins for single-molecule–based superresolution imaging. *Proc. Natl. Acad. Sci.* **111**, 8452-8457 (2014).
56. Subach, O.M., Patterson, G.H., Ting, L.-M., Wang, Y., Condeelis, J.S. & Verkhusha, V.V. A photoswitchable orange-to-far-red fluorescent protein, PSmOrange. *Nat. Meth.* **8**, 771-777 (2011).
57. Brakemann, T. et al. A reversibly photoswitchable GFP-like protein with fluorescence excitation decoupled from switching. *Nat. Biotechnol.* **29**, 942-947 (2011).
58. Shroff, H., Galbraith, C.G., Galbraith, J.A. & Betzig, E. Live-cell photoactivated localization microscopy of nanoscale adhesion dynamics. *Nat. Meth.* **5**, 417-423 (2008).
59. Ries, J., Kaplan, C., Platonova, E., Eghlidi, H. & Ewers, H. A simple, versatile method for GFP-based super-resolution microscopy via nanobodies. *Nat. Meth.* **9**, 582-584 (2012).

60. Bromley, S.K., Burack, W.R., Johnson, K.G., Somersalo, K., Sims, T.N., Sumen, C., Davis, M.M., Shaw, A.S., Allen, P.M. & Dustin, M.L. The Immunological Synapse. *Annu. Rev. Immunol.* **19**, 375-396 (2001).
61. Rossy, J., Owen, D.M., Williamson, D.J., Yang, Z. & Gaus, K. Conformational states of the kinase Lck regulate clustering in early T cell signaling. *Nat. Immunol.* **14**, 82-89 (2013).
62. Lillemeier, B.F., Mortelmaier, M.A., Forstner, M.B., Huppa, J.B., Groves, J.T. & Davis, M.M. TCR and Lat are expressed on separate protein islands on T cell membranes and concatenate during activation. *Nat. Immunol.* **11**, 90-96 (2010).