Seed-Point Detection of Clumped Convex Objects by Short-Range Attractive Long-Range Repulsive Particle Clustering

James Kapaldo, Xu Han, and Domingo Mery, *Member, IEEE*

Abstract—Locating the center of convex objects is important in both image processing and unsupervised machine learning/data clustering fields. The automated analysis of biological images uses both of these fields for locating cell nuclei and for discovering new biological effects or cell phenotypes. In this work, we develop a novel clustering method for locating the centers of overlapping convex objects by modeling particles that interact by a short-range attractive and long-range repulsive potential and are confined to a potential well created from the data. We apply this method to locating the centers of clumped nuclei in cultured cells, where we show that it results in a significant improvement over existing methods (8.2% in F_1 score); and we apply it to unsupervised learning on a difficult data set that has rare classes without local density maxima, and show it is able to well locate cluster centers when other clustering techniques fail.

Index Terms—seed-point detection; data mining; nuclei de-clumping; short-range attractive long-range repulsive

1 Introduction

Locating object and distribution centers is a fundamental step of both image processing and unsupervised machine learning or exploratory data mining. The automated analysis and classification of cells from microscopy images is an important field that includes both of these disciplines [1], [2], [3], [4], [5], [6]. In this field, the first step is often locating nuclei centers using image processing techniques, which allows for seed-point based segmentation algorithms [7], [8], [9], [10] to determine the nuclei boundaries. Using the segmentation results, features of each nucleus or cell are measured and used to classify the cells into different groups (e.g. cancer/non-cancer, different phenotypes/cell-phase [5], [11], [12], [13], [14], [15]). When searching for new groups (new phenotypes, new effects, ...) or when it not possible for an expert to create a labeled data set, unsupervised machine learning, where groups are determined by looking for similarities in the data, must be used.

Locating nuclei centers and using unsupervised learning to locate cluster centers are, abstractly, very similar: they both try to locate the centers of partially-overlapping convex objects or distributions. However, the techniques for solving each of these problems are quite different and are not ideal for several reasons. The current methods for locating nuclei centers (radial voting [10], [16], [17], [18], gradient convergence and sliding band filters [19], [20], Laplacian of Gaussian filters [21], [22], [23], [24]) try to compute a surface, named the *voting landscape*, that has local extrema at the nuclei centers. The fundamental problem with these methods is that the extrema of the voting landscapes are not at the nuclei centers; further, there can be extra extrema, where false nuclei are detected, or missing extrema, where true nuclei are not detected. Since the segmentation methods will return one region per seed-point, it is critical that each nuclei have only one seed-point as close as possible to the nuclei center [10], [22]. For unsupervised learning, there are many well known clustering techniques (K-means/fuzzy C-means [25], [26], [27], expectation maximization/mixture of Gaussians [28], hierarchical clustering [29], mean-shift [30]), but a general drawback of most of these methods is that they produce cluster centers that are biased towards high density regions in the data, and many of them require the number of clusters (which is normally not known beforehand) as an input. Mean-shift clustering can be better in this regard; however, each cluster must have a local maximum in the point density.

In this work, we develop a new clustering method (*SALR clustering*) that can be used for locating both nuclei centers and the cluster centers in scatter point data, while at the same time addressing the above issues. The premise of our method, which is graphically described in Figure 1 and which takes cues from the fundamental physics of modeling classical Wigner crystals [31] and modeling the formation of clusters [32], [33], is that we can find the centers of overlapping convex regions by modeling the dynamics of particles trapped in an appropriate confining well.

Consider a set of repulsive particles confined to a region; these particles will try to move as far apart from each other as possible. Figure 1a shows this for 10 particles confined to a parabolic potential well. If the region the particles are confined to has several local minima, then the particles will preferentially locate as close as possible to these minima to lower the total energy. Assuming our goal is to have particles near each local minima and nowhere else, we

- *This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.*
- *J. Kapaldo and X. Han are with the Department of Physics, University of Notre Dame, Notre Dame, IN, 46545. (e-mail: jkapaldo@nd.edu, xhan1@nd.edu)*
- *D. Mery is with the Department of Computer Science, Pontificia Universidad Católica de Chile, Santiago 7820436, Chile. (email: dmery@ing.puc.cl)*

Manuscript submitted Aug. 26, 2017.

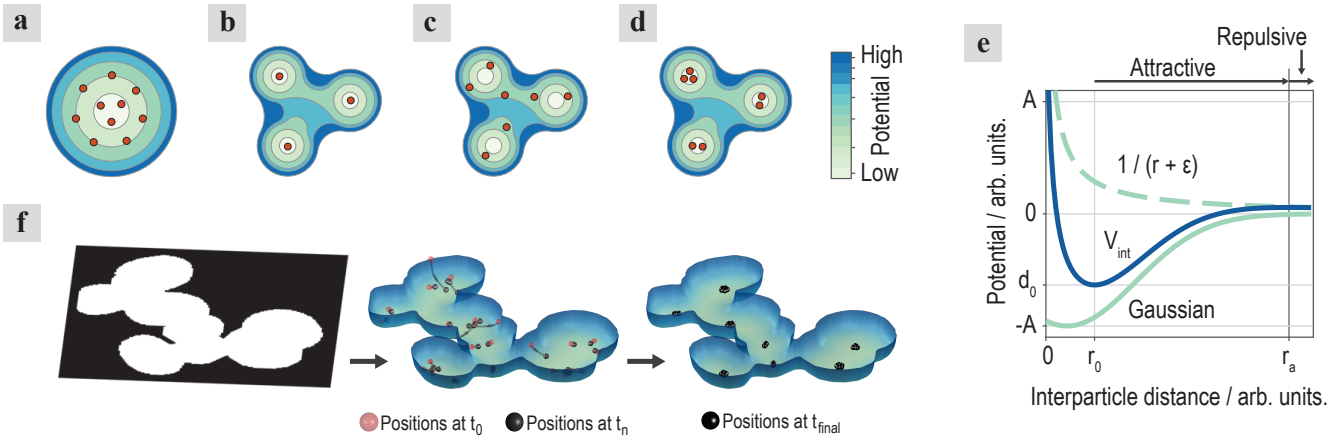


Fig. 1. Premise of SALR clustering. **a–d**, Intuition: schematic of the lowest energy configuration for **a**, 10 repulsive particles in a parabolic well (this is one of the two isomeric ground states [31]); **b**, three repulsive particles with three potential minima; **c**, seven repulsive particles with three potential minima; and **d**, seven SALR interacting particles with three potential minima. Contour levels represent potential energy. **e**, SALR interaction potential formed with a Gaussian attractive term and a $1/r$ repulsive term. The distance r_a, named the attractive extent, is the distance where particles switch from being attractive to repulsive. **f**, Starting from a region of overlapping convex objects, a confining potential is created (third dimension corresponds to potential value); particles are randomly placed and the dynamics modeled; at the end the particles will form clusters located approximately at the center of each convex object.

must have the same number of particles as the number of potential minima, since two particles cannot be near each other, see Figure 1b,c. This is a problem as the number of potential minima is normally not known before hand, but it can be solved by modifying how the particles interact with each other such that particles near each other are attracted to each other instead of repulsed—this short-range attractive long-range repulsive (SALR) particle interaction potential can be seen in Figure 1e. Now, we can use (many) more particles than we expect there to be potential minima, and we can have a cluster of particles located at each potential minima, see Figure 1d.

Using this premise, we create a confining potential from the region of overlapping convex objects that has a higher energy at the region's edges and a lower energy (valley) near the object centers, we randomly place many SALR particles in this region, and then we model the particle dynamics while slowly decreasing their speed; the final position of each cluster of particles will approximately give the center of each locally convex region, see Figure 1f and Supplemental Video 1.

We validate SALR clustering by applying it to the problem of locating nuclei centers, and show that it can significantly improve the location performance compared with previous methods and discuss how/why it is able to improve the performance. We also demonstrate the application of SALR clustering to unsupervised machine learning, and show that it can determine the correct number and position of clusters and better locate rare clusters that do not have local density maxima.

2 Modeling Particle Dynamics

The equations governing the particle dynamics are derived from the Hamiltonian of a set of N interacting charged particles,

$$\mathcal{H} = \sum_i \left(\frac{1}{2m_i} \mathbf{p}_i^2 + V(\mathbf{r}_i) \right) + \sum_{\substack{i \\ j \neq i}} \frac{k q_i q_j}{2} V_{\text{int}}(|\mathbf{r}_i - \mathbf{r}_j|), \quad (1)$$

where m, $\mathbf{p}$, $\mathbf{r}$, and q represent the particle's mass, momentum, position, and charge, respectfully, k is a coupling constant, and $|\mathbf{r}_i - \mathbf{r}_j|$ is the distance between $\mathbf{r}_i$ and $\mathbf{r}_j$. $V_{\text{int}}(\cdot)$ is the SALR particle interaction potential created with a Gaussian attractive term and a $1/r$ repulsive term

$$V_{\text{int}}(r) = \frac{1}{r + \epsilon} - A \exp\left(-\frac{(r - \mu)^2}{2\sigma^2} \right), \quad (2)$$

where r is the distance between two of the particles and ϵ is a small constant value ($\epsilon = 0.2$) to prevent the divergence at $r = 0$. An example of this potential is shown in Figure 1e. The values A, μ, and σ in (2) are set by solving a least-squares problem so that the depth d_0, the location of the potential minimum r_0, and the distance at which the potential goes from being attractive to repulsive, referred to as the attractive extent, r_a may be directly specified, see Figure 1e. For example, the correspondence between two sets of these parameters $(d_0, r_0, r_a) \rightarrow (A, \mu, \sigma)$ are $(-1, 2, 10) \rightarrow (1.58, 0.87, 2.82)$ and $(-1, 2, 15) \rightarrow (1.84, -1.32, 4.84)$.

The particles' equations of motion can be found from the Hamiltonian (1) to be

$$\frac{d\mathbf{p}_i}{dt} = -\boldsymbol{\nabla}_i V(\mathbf{r}_i) - \sum_{j \neq i} k q_i q_j \boldsymbol{\nabla}_i V_{\text{int}}(|\mathbf{r}_i - \mathbf{r}_j|) \quad (3)$$

$$\frac{d\mathbf{r}_i}{dt} = \frac{1}{m_i} \mathbf{p}_i, \quad (4)$$

where $\boldsymbol{\nabla}_i$ is the gradient operator acting on vector $\mathbf{r}_i$.

The system described so far conserves energy, which means that the particles/clusters will never stop moving. To find the lowest (or a low lying) energy state of the

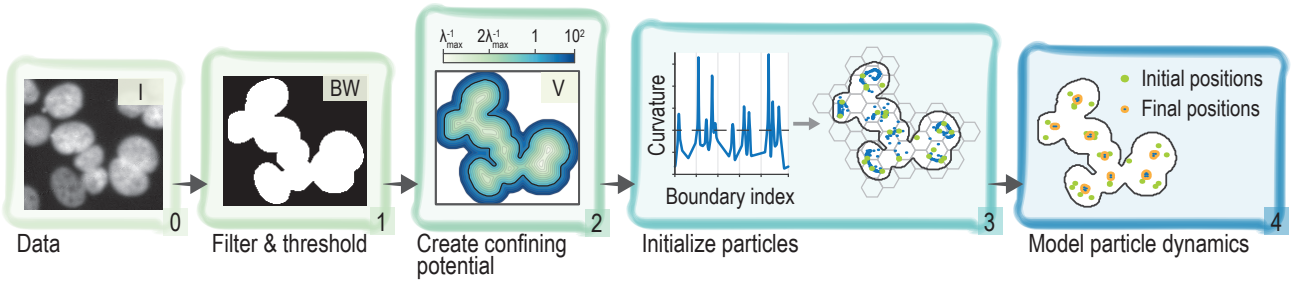


Fig. 2. Locating nuclei centers method. 1) create a binary mask by filtering and thresholding the image. 2) create a confining potential (a 3D version of this potential is shown in Figure 1f). 3) initialize particle positions: create possible set of points (blue dots) based on the boundary curvature and choose one point per grid element of an overlaid grid (green points). 4) model the particle dynamics.

system, we slowly decrease the temperature of the system by damping the particles:

$$\frac{d\mathbf{p}_i}{dt} = -\boldsymbol{\nabla}_i V(\mathbf{r}_i) - \frac{\alpha(t)}{m_i}\mathbf{p}_i - \sum_{j\neq i} kq_iq_j\boldsymbol{\nabla}_i V_{\text{int}}(|\mathbf{r}_i-\mathbf{r}_j|), \quad (5)$$

where $\alpha(t)$ is a time dependent damping constant. $\alpha(t)$ is not required to be time dependent, but faster convergence can be achieved by increasing the value of α as the simulation goes on, e.g. $\alpha(t) \propto t$.

Using (4) and (5) we can model the dynamics of the particles until the solution converges. We solve the set of $2N$ differential equations using a Runge-Kutta (2,3) solver [34] (implemented with Matlab), which provides high enough tolerance to solve the problem and is faster than higher order methods. Convergence is determined by looking at the average particle speeds over a short time span (the last few time steps of the ODE solver), and if this average speed is below some threshold, then we stop the simulation. Averaging over a time span is used because the clusters of particles can have vibrational or rotational modes [35] that take a long time to die out (perhaps 30% more computation time), and using the average speed allows these modes to be effectively ignored.

In Supplemental Note 1, we make connections between our optimization method and simulated annealing and gradient descent that could lead to faster convergence.

3 Locating Nuclei Centers

3.1 Method

The process we developed for applying our method to locating nuclei centers is shown in Figure 2. We go through each step below, and recommend viewing Supplemental Video 1 for a visual description.

3.1.1 *Filter & Threshold*

After reading in the data, the (overlapping) foreground objects should be separated from the background with the goal of creating a binary mask $BW(\mathbf{r})$ such that $BW(\mathbf{r}) = 1$ for all $\mathbf{r}$ in an object and $BW(\mathbf{r}) = 0$ for all $\mathbf{r}$ not in an object. For the nuclei images used in this work, we filtered the images with a Gaussian blur ($\sigma = 1$) and then used an adaptive log-weighted Otsu threshold [36].

3.1.2 *Construct confining potential*

We use the binary mask to create a confining potential well, $V(\mathbf{r})$. Let the set of coordinates belonging to the object be denoted by $\Omega = \{\mathbf{r} \mid BW(\mathbf{r}) = 1\}$. The confining potential is given by

$$V(\mathbf{r}) = \begin{cases} \text{dt}(\neg BW)^{-1}, & \forall\, \mathbf{r} \in \Omega \\ \text{dt}(BW)^2 + 1, & \forall\, \mathbf{r} \notin \Omega \end{cases} \quad (6)$$

where $\text{dt}(\cdot)$ is the distance transform, and $\neg$ is the negation operator ($0 \leftrightarrow 1$). The distance transform takes a binary image and assigns to each pixel the Euclidean distance to the nearest non-zero pixel in the binary image. The first line of (6) states that the potential inside the object is given by 1 over the distance to the nearest object boundary pixel, and the second line states that the confining potential increases quadratically outside of the object. Note that the potential is < 1 inside the object and > 1 outside. After forming $V(\mathbf{r})$, it is smoothed with a Gaussian with $\sigma = 1$ pixel.

This confining potential is not scale invariant: as a object becomes larger, the potential will become more *flat-bottomed*, which will cause the particles to push each other closer to the object boundary, as is shown in Supplemental Figure 1a. Scale invariance can be added by implicitly scaling all objects so that their maximum distance transform values, $\lambda = \max(\text{dt}(\neg BW))$, are equal to the same fixed value, $\lambda_{\max}$ (see Supplemental Figure 1b). This changes the confining potential on the interior of the object to be

$$V_{\mathbf{r}\in\Omega} = \left[1 + \frac{\lambda_{\max} - 1}{\lambda - 1}\Big(\text{dt}(\neg BW) - 1\Big)\right]^{-1}. \quad (7)$$

Note that for a single ellipsoidal object, λ represents the semi-minor axis of the object; scaling the distance transform effectively means we are resizing each object to have the same semi-minor axis, $\lambda_{\max}$.

3.1.3 *Particle initialization*

The next step is initializing the particles: determine the density/number of particles to use, set their mass and charge, and set their initial velocities and positions.

Number: The number of particles is determined by setting the particle density. The parameter we use to set the particle density is the Wigner-Seitz radius r_s, which

describes the amount of space taken up by each particle. The number of particles N, in 2D, is then approximately

$$N = \frac{A}{\pi r_s^2}, \tag{8}$$

where A is the area of the object (the number of elements in set Ω) and πr_s^2 is the amount of space per particle.

MASS & CHARGE: We set the mass of the particles to one and the particle charge to decrease with N as $q = N^{-\beta}$. Larger values of β result in particle clusters that interact with each other less and can move around more, smaller values of β result in particle clusters that strongly interact and push each other closer to the object boundary.

VELOCITY: The particles are given random initial velocities according to $\mathbf{v}_0 = s_0\hat{\mathbf{u}}$, were s_0 is the initial speed and $\hat{\mathbf{u}}$ is a random unit vector.

POSITION: There are several possible methods of selecting the particles' initial positions; we describe three methods which would each have their own benefit depending on the data.

i) Random: The most simple method, choose N random points out of the set Ω.

ii) Uniform random: To ensure we can find all of the seed-points, the particles' initial locations should uniformly cover the domain of interest (Ω). This can be accomplished by overlaying a lattice across the domain and placing one particle in each lattice unit cell. In order to have approximately N particles, the lattice constant is chosen such that the lattice unit cell's area is approximately the size of particle, πr_s^2 in 2D. In 2D, we use a hexagonal lattice with lattice constant $2r_s$, and in higher dimensions we use a simple (hyper)cubic lattice. The location of the particle within each lattice cell is selected by choosing one random point from the points in Ω that are also in each lattice cell, $\Omega \cap \mathrm{U}_i$, where U_i is the set of all coordinates in the i'th lattice cell.

iii) Convex hull transformed center of curvature (CvxHll CoC): Better results can be obtained if we can choose points that are approximately where the true seed-points are expected. For this, we compute a set of possible positions, R_0, using the centers of curvature of each boundary vertex of the region Ω, see Supplemental Note 2. For each lattice cell, we then choose one random point from the set, $\mathrm{R}_0 \cap \mathrm{U}_i$. Note that the centers of curvature can be far from the geometric center of elliptical regions since the long side of an ellipse has a smaller curvature (larger radius). This is addressed by modifying the boundary curvature to be the convex hull of each negative curvature region (See Supplemental Figure 2 for an example of this method.)

As a final note on position initialization, the confining potential can be used to tighten the constraint on where particles can be located. At the least, we require all initial positions to have a confining potential value less than 1 (the particles must start inside the object); however, we can further require that the confining potential value at all initial positions be in some range $[V_{\min}, V_{\max}]$, which can be useful for both improving control over initial positions and reducing the number of particles used in the simulation.

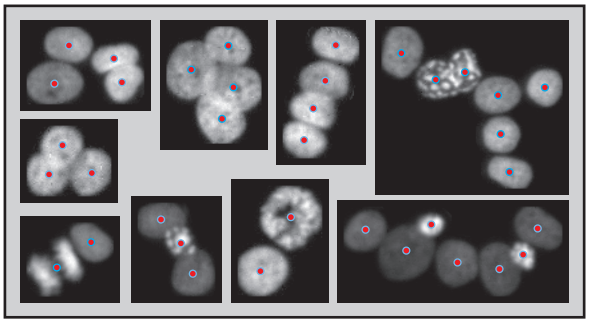


Fig. 3. Example nuclei clumps. Examples of the nuclei clumps used for validation; the contrast for each object has been enhanced. The red dots represent the labeled nuclei centers, and the edge of the markers represent our expected confidence in their location (~ 3 pixel radius).

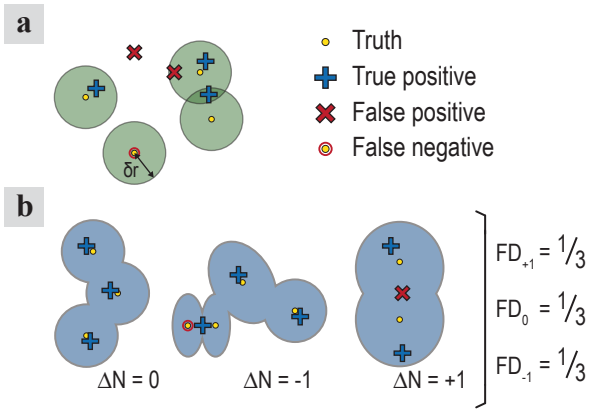


Fig. 4. Comparison metrics. **a**, Definition of true positive and false positive calculated seed-points and false negative seed-points. **b**, Example showing the calculation of the fractional distribution $\mathrm{FD}_{\Delta \mathrm{N}}$ measure for three objects: one with the correct number (left), one with one too few (middle), and one with one too many (right) computed seed-points.

3.1.4 *Model particle dynamics*

The particles are modeled using (4) and (5) until the solution converges. Box 4 in Figure 2a shows the particle positions after convergence; you can see clusters of particles are located at each of the nuclei centers. After the model has converged, we extract the center of each cluster by grouping together all particles within some distance of each other $(0.7\, r_0 + 0.3\, r_a)$ and then finding the mean location of each group. (This distance is empirically set and could be any distance $< r_a$ and $> r_0$.) These cluster centers are the final seed-points returned by our method.

3.1.5 *Reduction of work*

It is not necessary to model the particle dynamics on all objects; if an object is convex, or it is smaller than a particle (πr_s^2), or only one particle would be modeled, then we do not run the simulation, but simply return the centroid of the object as the seed-point. We determine if an object is convex by testing that the all boundary curvature is less than some threshold, $0.25/\lambda$.

3.2 Results

The data we use for validating our method is a set of 2,420 images of nuclei clumps with a total of 7,789 nuclei. The

cells are epithelial cancer cells from the tongue (squamous cell carcinoma, SCC25) that we cultured, stained with 4',6-diamidino-2-phenylindole (DAPI) to label their DNA, and then had imaged using a whole slide fluorescence reader, see Supplemental Note 3 for more details. The center of each of these 7,789 nuclei were manually labeled to create the truth data. Figure 3 shows an example of nine nuclei clumps with the labeled nuclei centers (more examples can be seen in Supplemental Figure 3).

We use two metrics to determine the performance of the computed nuclei center locations. The first is the F_1 score, which ranges between 0 and 1 and is the harmonic mean of the precision and recall; a F_1 value of 1 means that all seed-points (nuclei centers) were calculated perfectly, and a F_1 value of 0 means not a single seed-point was calculated correctly. It is computed as

$$F_1 = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{9}$$

where TP, FN, and FP are the number of true positives, false negatives, and false positives, respectively. A graphical definition of these terms is shown in Figure 4a: a calculated seed-point is TP if it is within a distance of δr to a truth point and it is the closest calculated seed-point to that truth point, otherwise it is FP; a FN point is a truth point that does not have a calculated seed-point assigned to it. For example, the rightmost red X in Figure 4a is FP because there is another computed seed-point closer to the truth location than it.

The second metric we use measures the fractional distribution, $\mathrm{FD}_{\Delta \mathrm{N}}$, of objects (nuclei clumps) with the correct, too many, or too few calculated seed-points. For example, if $\mathrm{FD}_{-1} = 0.1$, then 10% of all the objects have one less calculated seed-point than the true number. This metric is important as it directly measures the ability of a method to calculate the correct number of seed-points, even if the seed-points are not in the correct location. The fractional distribution $\mathrm{FD}_{\Delta \mathrm{N}}$ of ΔN is given by

$$\mathrm{FD}_{\Delta \mathrm{N}} = \frac{1}{M} \sum_{i=1}^{M} \delta_{\Delta \mathrm{N},\,(\mathrm{FP}_i - \mathrm{FN}_i)} \tag{10}$$

where M is the number of objects (2,420), $\delta_{\cdot,\cdot}$ is the Kronecker delta, and FP_i and FN_i are the number of false positives and the number of false negatives for the i'th object. An example calculating $\mathrm{FD}_{\Delta \mathrm{N}}$ is shown in Figure 4b. Depending on how the seed-points will be used, either F_1 or $\mathrm{FD}_{\Delta \mathrm{N}}$ could be more important.

Using these two metrics, we compare the results of our method to seven other standard and leading methods for locating nuclei centers: multi-pass voting (MPV) by [16], two versions of single-pass voting by [10] ($\mathrm{SPV}_{\mathrm{Qi}}$) and by [18] ($\mathrm{SPV}_{\mathrm{Xu}}$), sliding-band filter (SBF) by [19], generalized Laplacian of Gaussian (gLoG) by [24], maximally stable extremal regions (MSER) by [37], and directly using the local maximum of the distance transform (Dist. Trans.). We optimized the parameters of each method to maximize the sum of $F_{1,\,\delta r=3}$ and FD_0; descriptions of the methods along with details on the parameters we optimized over and the final parameters used for each method can be seen in Supplemental Note 4. The parameters used for our method

TABLE 1
Model parameter values used for our method in computing the results of Figure 5.

Parameter name	Symbol	Default value
Particle initialization		CvxHll CoC
Wigner-Seitz radius	r_s	5
Confining potential depth	$\lambda_{\max}$	18
V_{int} attractive extent	r_a	13
V_{int} minimum location	r_0	2
V_{int} depth	d_0	-1
Max init. particle potential	$V_{\max}$	1/5
Charge normalization	β	1/3
Damping rate	$\alpha(t)$	$5 \cdot 10^{-4}\,t$
Initial speed	s_0	0.01
Mass	m	1
Coupling constant	k	1

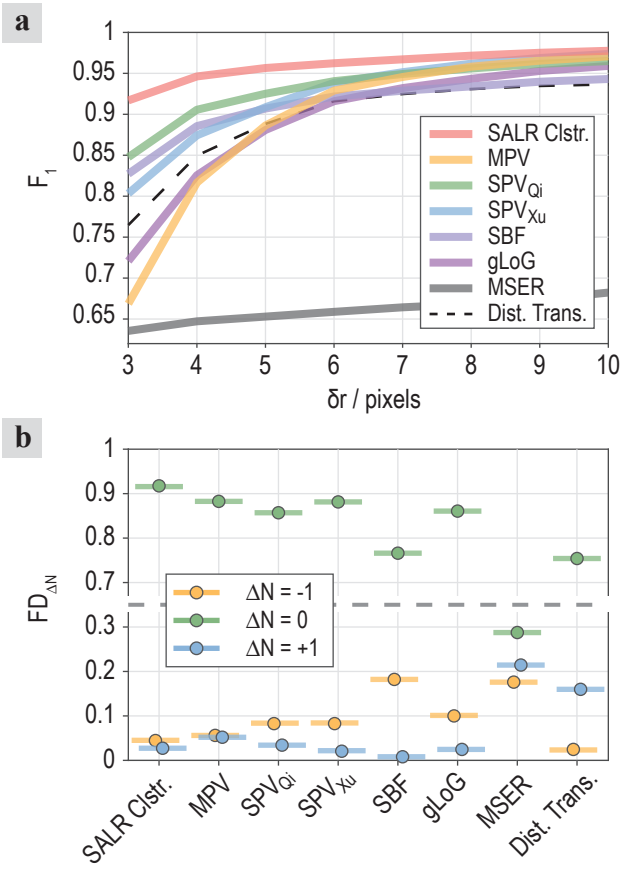


Fig. 5. Methods comparison. **a**, F_1 versus δr for the seven methods. **b**, $\mathrm{FD}_{\Delta \mathrm{N}}$ for $\Delta N = \{-1, 0, +1\}$ for the seven methods.

are shown in Table 1. Many of these parameters are set empirically, and several were set by explicit optimization and are discussed in Supplemental Note 5.

We show the results of the comparison in Figure 5. Our method (SALR Clstr.) has both the best F_1 score (0.069 higher, 8.2%, than the next best method at $F_{1,\,\delta r=3}$ and 0.028 higher, 3.0%, than the next best average F_1 score) and the best FD_0 value (0.033 higher, 3.8%, than the next best method).

In particular, note that our method does much better than the distance transform; this is important because our confining potential is proportional to one over the distance

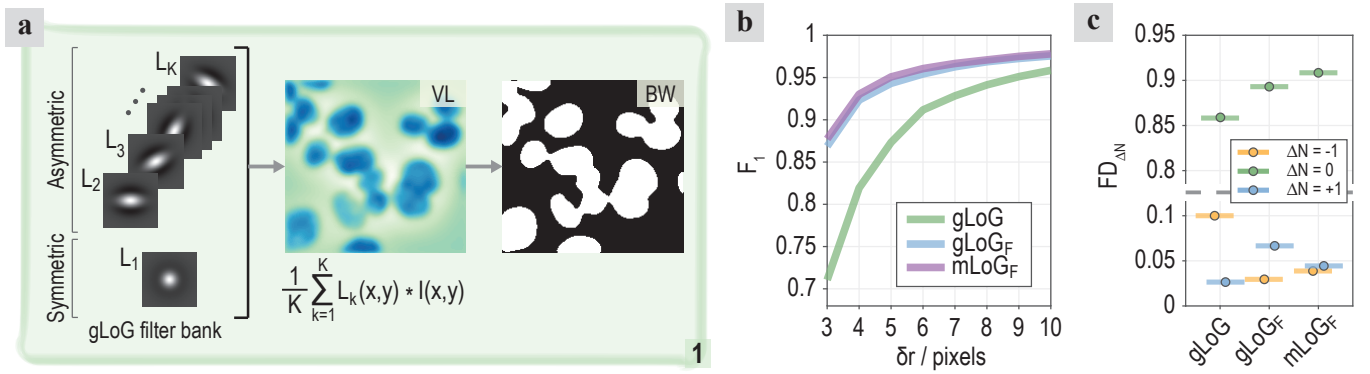


Fig. 6. Advanced filtering. **a**, *Filter & threshold* step of method: convolve a gLoG filter bank with an image to create a voting landscape, then threshold to create the binary mask. **b**, F_1 score using the previous gLoG method (shown in Fig. 5), using the entire gLoG filter bank (gLoG_F), and using only the symmetric gLoG filter (gLoG_SF). **c**, $\text{FD}_{\Delta\text{N}}$ results for the same three methods.

transform. This means our method is not simply locating the local maxima of the distance transform, see Discussion.

Additional validation of our method's scale invariance as well as the performance and computation time dependence on the initial particles starting locations and density are discussed/shown in Supplemental Note 6/Supplemental Figure 5 and Supplemental Note 7/Supplemental Figure 6.

3.3 Advanced filtering

The previous methods we compared to all use a voting landscape to determine the nuclei centers. We surmised that thresholding their voting landscape to create a binary mask and then using our method, instead of using the local maxima of the voting landscape, could improve the performance of the methods. We tried this using LoG filtering as it is likely the most easy to implement as well as the fastest.

In Figure 6a we show the *Filter & Threshold* step using gLoG. A gLoG filter bank is created that consists of a symmetric multiscale LoG filter (mLoG) and several asymmetric multiscale LoG filters. The voting landscape is formed by summing the convolution of each of these filters with the input image, and the binary mask is created with a simple threshold. In Figure 6b and 6c we show the F_1 and $\text{FD}_{\Delta\text{N}}$ results of the original gLoG method (the same results as shown in Figure 5) and the results of using gLoG as the filter for our method (gLoG_F). You can see that using gLoG with our method results in a significant improvement in $F_{1,\,\delta r=3}$, by ~ 0.15, and a large improvement in FD_0, by $\sim 4\%$. Additionally, using only the symmetric mLoG as the filter for our method (mLoG_F) results in even better performance than the gLoG_F, with about a 2% increase in FD_0.

These results are important because much research does not use images where the nuclei can be easily segmented by thresholding (like the cultured cells in this work), but use colored histopathological (tissue) images, which are not easily thresholded. Our results show that the nuclei regions can be detected using gLoG filters and then accurate nuclei center locations and accurate nuclei count can be obtained by using our method.

4 Application to Data Clustering

In unsupervised machine learning and data mining, data often comes in the form of scatter point data where each point is an observation and each dimension is a different measured quantity. An example of 2D scatter point data is shown in the top of Figure 7a. The easiest way to apply our seed-point detection method to this type of data is by *binning* the data, where a grid is overlaid on the data and the number of points in each bin is counted. The result, shown in the bottom of Figure 7a, can be thought of as an image, and we can threshold it to create a binary mask and then proceed as before.

4.1 Simple 3D data

We first validate the use of our method for detecting the cluster centers of scatter point data by using a simple 3D distribution where we can use the well known k-means clustering. The point-count isosurfaces of the 3D distribution can be seen along with the projection of the isosurfaces to the 2D axis planes in Figure 7b. We first scale the z-axis so that the objects are approximately the same size along any direction (see Supplemental Note 8), we then threshold the point-count and create the confining potential, which is shown in Figure 7c.

We initialize particles for the simulation using a uniform random distribution with $r_s \approx 9$ (resulting in 79 particles); all parameters other than particle density are the *same* as we used in the 2D images above (values shown in Table 1). We simulated the particles 20 times (each time with new initial positions) and show the computed seed-points from each iteration as the black points in the 3D view of Figure 7c. The seed-points are well located in seven primary locations; the few points not at the primary locations can be reduced by tuning the particle damping rate (Supplemental Note 9); however, it is computationally faster to simply cluster the results of these 20 iterations and take all clusters with more than a few points as the final seed-points, which are shown as the large red dots in the 3D view. This process of clustering the results of several iterations is very robust; the 2D view in Figure 7c shows the results of repeating this process 20 times, and you can see the computed seed-points are all in the same location.

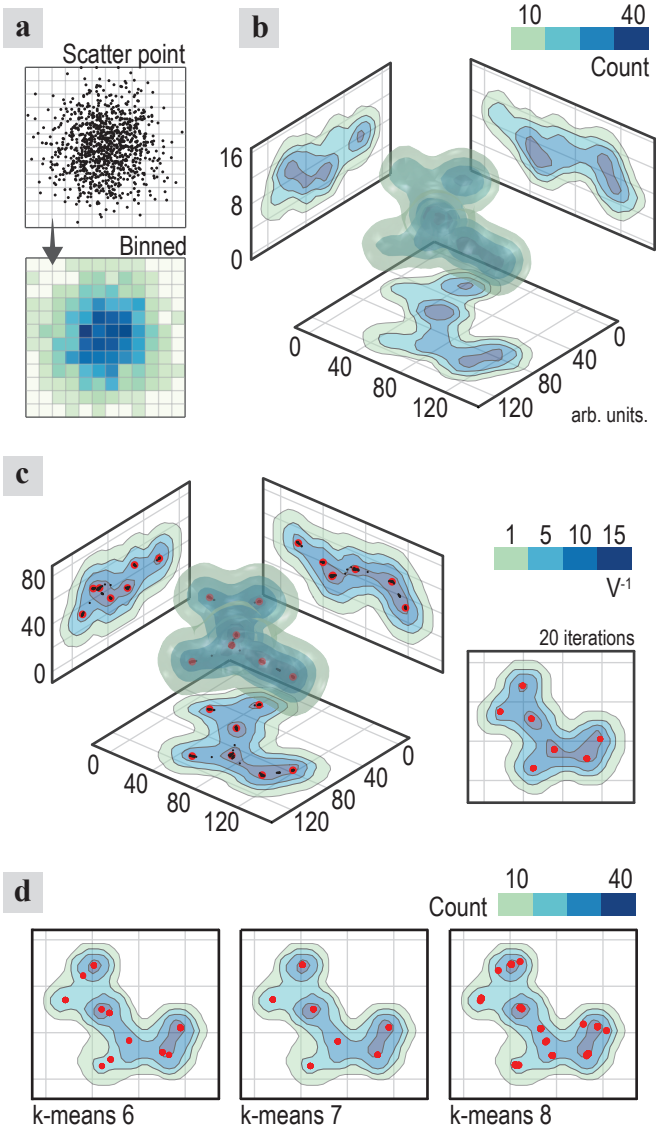


Fig. 7. SALR clustering for scatter point data. **a**, Example showing how scatter point data is binned. **b**, Isosurface plot of the point count in our 3D example; the axis planes show the projections of the isosurfaces to 2D. **c**, The confining potential created by thresholding the point count at 5 and then using (7). The black points are the results of running our seed-point calculation method 20 times, and the red points are the clusters formed from these results. The 2D view of the x-y plane shows the results of repeating this process 20 times. **d**, The red points show the result of running k-means clustering 20 times using 6, 7, and 8 clusters.

We check the validity of our seven seed-points by running k-means clustering with two replicates (as implemented by Matlab's k-means++ routine) using 6, 7, and 8 clusters. We ran the clustering 20 times in each case, and show the results in Figure 7d. The stability of the results using 7 clusters (all 20 iterations produce the same seven positions), as compared with 6 or 8, imply that this 3D distribution does have seven clusters; and, the strong agreement in seed-point location between the k-means results and the results of our method imply that these are likely the correct locations.

These results indicate that our method can determine the correct number of seed-points and their correct location, and, in particular, could be a good choice for locating the centers of nuclei in 3D images, which this 3D distribution resembles.

4.2 More complex 5D data

We next turn to a real data set with five dimensions and 2.7 million points; each point represents one cell' nucleus (the same SCC25 cells as above) and the five features give a measure of the damage done to the nuclei. The data is scaled in each dimension by its standard deviation and then binned (about 33 bins in each dimension). Figure 8a shows point-count isosurfaces for two (of the ten possible) 3D projections of the 5D data; the data has one very dense region in the center (which are undamaged nuclei) and maybe three long/thin low-density regions extending outwards from it. The long/thin regions mean that the distance transform cannot be used to create the confining potential (Supplemental Note 8). Instead, we use one over the point density for the confining potential; however, we must scale the magnitude of the potential gradient $|\nabla V|$ so that it is approximately the same order of magnitude as the particle repulsion (see Discussion below). In addition to this change, we introduce two generalizations to our method.

- *Distance metric*: In higher dimensions, we found that we can get more stable results using the Minkowski distance metric where the distance between two points, $\mathbf{x}$ and $\mathbf{y}$, is defined as
$$d = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{11}$$
where n is the dimension of the space and $p \geq 1$. If we set $p = 2$ then we have standard Euclidean distance; and if we set $p \to \infty$, then $d = \max_i |x_i - y_i|$, which, in our case, would mean that two particles are only attracted to each other if they are within r_a in all dimensions. (Note that, in general, any distance metric can be used as long as the interaction potential is correctly defined.)
- *Isotropic scaling*: We make a distinction between the data space (the N-D space the data is defined in) and the solver space (the N-D space we solve the equations of motion). We map between the two spaces with a simple scale factor defined to be ℓ/r_a, where r_a is the attractive extent in data space and ℓ is the *characteristic distance* of the solver space, which, for this paper, can be directly interpreted as the attractive extent in the solver space.
Introducing the solver space has two primary benefits: 1) If the attractive extent in data space should be $r_a \lesssim 2$, then it is not possible to accurately solve for the interaction potential parameters A, μ, and σ (assuming $r_0 \approx 0.2 r_a$ and $d_0 = -1$). The solver space lets us scale up the problem to use an attractive extent of ℓ at which we can accurately solve for the parameters. 2) Changing the value of ℓ can precisely control the magnitude of the particle interaction (this could also be achieved using the coupling constant, k, if one wanted).
In Supplemental Figure 8, we show how the interaction potential and force in data space change as ℓ goes from 10 to 50. The interaction force is strongest at $\ell = 10$ and decreases as ℓ increases.

We model particles five times (small black points in Figure 8a) and cluster the results (red points in Figure 8a).

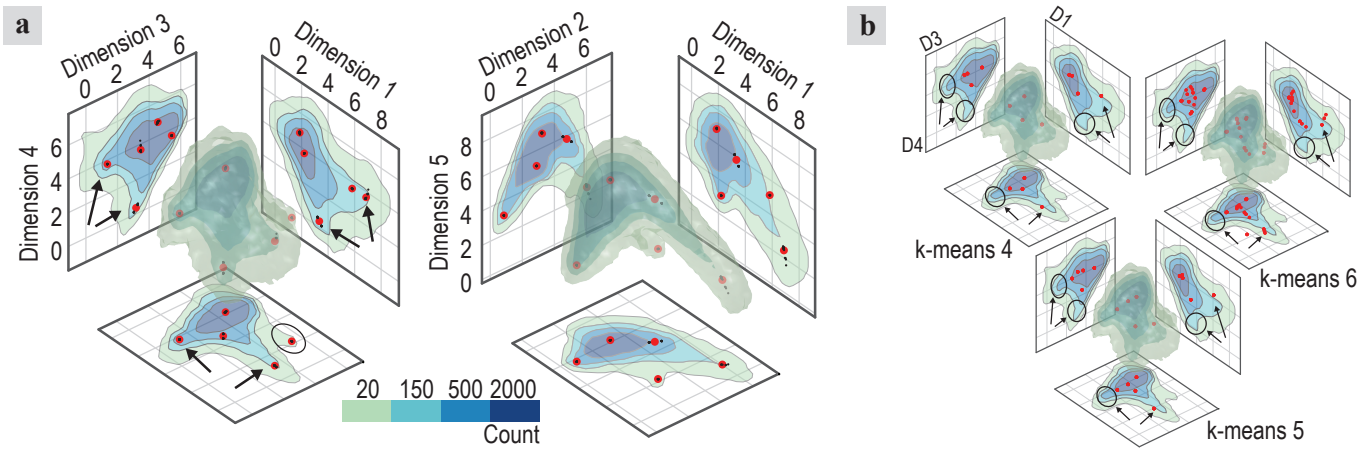


Fig. 8. SALR clustering for real data. **a**, projections of the 5D point count down to dimensions 1, 3, and 4 and dimensions 1, 2, and 5. Black points are the results from each of the 5 iterations, red points are the final seed-points. Arrows label the three low density regions. **b**, results using k-means with 4, 5, and 6 clusters. The black arrow labels the one seed-point that approximately locates a low-density region; the circles show the two missing seed-points.

You can see that our method located the three low density extrusions (labeled by black arrows) very well, and our method even finds the very small, but distinct, cluster marked by the circle in right plot of Figure 8a. (Supplemental Video 2 shows an animation of the particle dynamics along with the comparison to k-means with 5 clusters.) The parameter values used in these simulations were as follows; uniform random distribution, $r_s \approx 1.5$, $[V_{\min}, V_{\max}] = [1/4, 1/6]$, $r_a = 2$, $r_0 = 0.15\, r_a$, $d_0 = -1$, $\ell = 12$, $p = 4$, and the potential force was scaled so that the 99% value is 0.4. (The potential bound requirements helps to reduce the number of particles, which is 147, and to position the particles near the outskirts of the region, which is better, in general.)

As means of comparison, we used k-means clustering with two repetitions for 4, 5, and 6 clusters; we repeated this 20 times and show the results in Figure 8b. The results using 4 or 5 clusters are stable (the clusters are in the same place in all 20 iterations), but are only able to approximately locate one of the three low density regions, while missing two of them (labeled by the circles). The reason k-means does not locate the low density regions well is that the cluster centers it finds are the locations which minimize the variance in each cluster, and the variance can be minimized by moving the cluster centers closer to the high density regions. (The same/similar reasoning also applies for fuzzy c-means or mixture of Gaussians.) This demonstrates an important feature of our method: our method can locate the center of low density regions (also called rare classes) even when the low density regions do not have local maximum.

Looking at the computation time, the k-means with 5 clusters, which was the fastest, takes 15 ± 5 sec per repetition; our method takes 1.15 ± 0.08 sec per iteration with ~ 5.5 sec overhead for binning, smoothing, and computing the confining potential gradient. (Note each repetition of k-means and each iteration of our method can run in parallel. Computer specifications are shown below.) Thus, our method can lead to an important speed increase for some data sets.

5 Discussion

It is interesting to consider how, and under what conditions, SALR clustering leads to these improvements. The answer comes by analyzing the force on the particles in the simulation: the confining potential exerts a force on the particles with a magnitude of $|\nabla V|$ (which is the slope of the confining potential in Figure 1f), and the particles repel each other with a force of about r_a^{-2}. If the force from the confining potential is much larger than the repulsive force, $|\nabla V| \ggg r_a^{-2}$, then the particles will only find the local minimum of the confining potential—this would lead to the results labeled Dist. Trans. in Figure 5, and is the same thing that the previous methods do in finding the local extrema of the voting landscape or fining the local maxima of the point density (mean-shift). Alternatively, if the repulsive force is much larger than the confining force, $r_a^{-2} \ggg |\nabla V|$, then the particles will ignore the confining potential minima and spread out—this would result in particle clusters separated by at least a distance r_a that uniformly cover the region. In between these two regimes, when the two forces are approximately the same order of magnitude, $|\nabla V| \sim r_a^{-2}$, there is an interaction between the confining potential and the particle repulsion, and this is the regime we operate in and that leads to SALR clustering's improved performance. (When using the distance transform to create the confining potential in our images, our confining force is $\propto r^{-2}$, where r is the distance to the nearest region boundary, and the repulsive force between the particles is also $\propto r^{-2}$, where r is the distance between particles.) This need for the forces to be approximately equal is why it is necessary to scale the magnitude of the potential gradient when the inverse point density is used as the confining potential.

Our method can be applied to even higher dimensional data sets, but it likely will not be possible to bin the data as we did above. Binning the data requires a large amount of space (the 5D data set, 37x30x28x34x36, takes ~ 150 MB as single precision), and is not necessary when the density is used as the confining potential. The benefit of binning the data is computational speed, as it allows us to pre-

compute the density and its gradient everywhere. If the data cannot be binned, then the gradient of the density will need to be calculated (using a nearest neighbors range search) while solving the differential equations; this could be quite slow. Additionally, the magnitude of the gradient should be scaled; thus, before modeling the particle dynamics, a random sampling of the gradient magnitude will need to be performed so that the scale factor can be determined.

There are still many aspects of SALR clustering that are missing or can be improved and expanded upon, such as using an asymmetric particle interaction, developing a performance metric, implementing a clustering method (where each scatter point is placed into a specific cluster), and using multiple confining potentials and types of particles. These are each briefly discussed in Supplemental Note 10.

6 Conclusion

SALR clustering can represent a significant improvement in locating the centers of overlapping convex objects: it locates the correct number of nuclei more often and the nuclei centers more accurately than standard and leading methods; it can significantly improve the performance of previous methods; and it is able to determine, not only the number of clusters, but the correct position of the cluster centers in data clustering while not requiring a cluster to have a local density maximum.

Acknowledgments

The authors would like to acknowledge Guillermina Garcia from Sanford-Burnham Medical Research Institute for providing the whole slide fluorescence imaging. J.K. acknowledges Prof. Sylwia Ptasinska for the opportunity and resources to work on this project. J.K. and X.H. acknowledge the support from the US Department of Energy Office of Science, Office of Basic Energy Sciences, under Award Number DE-FC02-04ER15533. This is contribution number NDRL 5175 from the Notre Dame Radiation Laboratory.

Computer specifications

Computation times related to locating the nuclei centers are based on using a 64-bit i7-4790 CPU 3.60 GHz with 32.0 GB ram. Computations times reported for the 5D data set are based on using a 64-bit i7-3770 CPU 3.40 GHz with 12.0 GB ram.

Code & data availability

The SALR clustering code (written in Matlab) and documentation, as well as all images, truth data, and scatter point data used in this work, are in Supplementary Software and Data and available on GitHub (https://github.com/jkpld/SALR_Clustering). In particular, we include example scripts that directly reproduce the results of this work.

Author contributions

J.K. conceived the project, implemented the method, created validation truth data, analyzed the results, and wrote the manuscript. X.H. performed all tasks relating to the cell work (cell culture, cell experiments, staining the cells, having them imaged) and provided helpful discussion throughout the course of the project. D.M. provided valuable support and feedback in elevating the project and improving the manuscript.

References

[1] F. Xing and L. Yang, "Robust Nucleus/Cell Detection and Segmentation in Digital Pathology and Microscopy Images: A Comprehensive Review," *IEEE Reviews in Biomedical Engineering*, vol. 3333, no. c, pp. 1–1, 2016.

[2] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu, "Methods for Nuclei Detection, Segmentation and Classification in Digital Histopathology: A Review. Current Status and Future Potential," *IEEE Reviews in Biomedical Engineering*, vol. PP, no. 99, pp. 1–1, 2013.

[3] T. J. Fuchs and J. M. Buhmann, "Computational pathology: Challenges and promises for tissue analysis," *Computerized Medical Imaging and Graphics*, vol. 35, no. 7-8, pp. 515–530, 2011.

[4] S. Kothari, J. H. Phan, T. H. Stokes, and M. D. Wang, "Pathology imaging informatics for quantitative analysis of whole-slide images." *Journal of the American Medical Informatics Association : JAMIA*, vol. 20, no. 6, pp. 1099–108, 2013.

[5] C. Sommer and D. W. Gerlich, "Machine learning in cell biology teaching computers to recognize phenotypes," *Journal of Cell Science*, vol. 126, no. Pt 24, pp. 5529–5539, 2013.

[6] P. J. Navarro, F. Pérez, J. Weiss, and M. Egea-Cortines, "Machine learning and computer vision system for phenotype data acquisition and analysis in plants," *Sensors (Switzerland)*, vol. 16, no. 5, 2016.

[7] L. D. Cohen, "Contour Models," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 2, pp. 211–218, 1991.

[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[9] C. Zimmer, E. Labruyère, V. Meas-Yedid, N. Guillén, and J. C. Olivo-Marin, "Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing," *IEEE Transactions on Medical Imaging*, vol. 21, no. 10, pp. 1212–1221, 2002.

[10] X. Qi, F. Xing, D. J. Foran, and L. Yang, "Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 3, pp. 754–765, 2012.

[11] B. T. Grys, D. S. Lo, N. Sahin, O. Z. Kraus, Q. Morris, C. Boone, and B. J. Andrews, "Machine learning and computer vision approaches for phenotypic profiling," *Journal of Cell Biology*, vol. 216, no. 1, pp. 65–71, 2017.

[12] T. Blasi, H. Hennig, H. D. Summers, F. J. Theis, J. Cerveira, J. O. Patterson, D. Davies, A. Filby, A. E. Carpenter, and P. Rees, "Label-free cell cycle analysis for high-throughput imaging flow cytometry," *Nature Communications*, vol. 7, p. 10256, 2016.

[13] D. Chen, S. Sarkar, J. Candia, S. J. Florczyk, S. Bodhak, M. K. Driscoll, C. G. Simon, J. P. Dunkers, and W. Losert, "Machine learning based methodology to identify cell shape phenotypes associated with microenvironmental cues," *Biomaterials*, vol. 104, pp. 104–118, 2016.

[14] Q. Zhong, A. G. Busetto, J. P. Fededa, J. M. Buhmann, and D. W. Gerlich, "Unsupervised modeling of cell morphology dynamics for time-lapse microscopy," *Nature Methods*, vol. 9, no. 7, pp. 711–713, 2012.

[15] J. Wang, X. Zhou, P. L. Bradley, S.-F. Chang, N. Perrimon, and S. T. Wong, "Cellular Phenotype Recognition for High-Content RNA Interference Genome-Wide Screening," *Journal of Biomolecular Screening*, vol. 13, no. 1, pp. 29–39, 2007.

[16] B. Parvin, Q. Yang, J. Han, H. Chang, B. Rydberg, and M. H. Barcellos-Hoff, "Iterative voting for inference of structural saliency and characterization of subcellular events," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 615–623, 2007.

[17] X. Zhang, H. Su, L. Yang, and S. Zhang, "Fine-grained histopathological image analysis via robust segmentation and large-scale retrieval," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 5361–5368, 2015.

[18] H. Xu, C. Lu, and M. Mandal, "An Efficient Technique for Nuclei Segmentation Based on Ellipse Descriptor Analysis and Improved Seed Detection Algorithm," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 5, pp. 1729–1741, 2014.

[19] P. Quelhas, M. Marcuzzo, A. M. Mendonça, and A. Campilho, "Cell nuclei and cytoplasm joint segmentation using the sliding band filter," *IEEE Transactions on Medical Imaging*, vol. 29, no. 8, pp. 1463–1473, 2010.

[20] T. Esteves, P. Quelhas, A. M. Mendonça, and A. Campilho, "Gradient convergence filters and a phase congruency approach for in vivo cell nuclei detection," *Machine Vision and Applications*, vol. 23, no. 4, pp. 623–638, 2012.

[21] T. Lindeberg, "Feature Detection with Automatic Scale Selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79 – 116, 1998.

[22] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 841–852, 2010.

[23] H. Kong, H. C. Akakin, and S. E. Sarma, "A generalized laplacian of gaussian filter for blob detection and its applications," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1719–1733, 2013.

[24] H. Xu, C. Lu, R. Berendt, N. Jha, M. Mandal, and S. Member, "Automatic Nuclei Detection based on Generalized Laplacian of Gaussian Filters," *Journal of Biomedical and Health Informatics*, vol. XX, no. XX, pp. 1–12, 2016.

[25] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 233, pp. 281–297, 1967.

[26] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[27] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, 1981.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, 1977, vol. 39, no. 1.

[29] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[30] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[31] F. Bolton and U. Rössler, "Classical model of a Wigner crystal in a quantum dot," *Superlattices and Microstructures*, vol. 13, no. 2, p. 139, 1993.

[32] S. Mossa, F. Sciortino, P. Tartaglia, and E. Zaccarelli, "Ground-state clusters for short-range attractive and long-range repulsive potentials," *Langmuir*, vol. 20, no. 24, pp. 10 756–10 763, 2004.

[33] F. Sciortino, S. Mossa, E. Zaccarelli, and P. Tartaglia, "Equilibrium cluster phases and low-density arrested disordered states: The role of short-range attraction and long-range repulsion," *Physical Review Letters*, vol. 93, no. 5, pp. 5–8, 2004.

[34] P. Bogacki and L. Shampine, "A 3(2) Pair of Runge-Kutta Formulas," *Appl Math Lett*, vol. 2, no. 4, pp. 321–325, 1989.

[35] S. A. Blundell and S. Chacko, "Excited states of incipient Wigner molecules," *Physical Review B*, vol. 83, no. 19, p. 195444, 2011.

[36] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

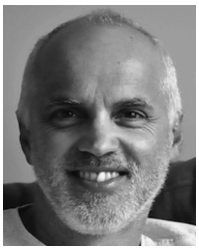
[37] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal," *In British Machine Vision Conference*, pp. 384–393, 2002.

[38] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

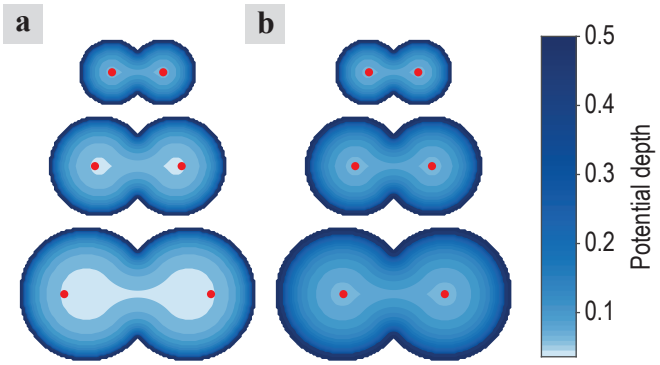
[39] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations 2015*, pp. 1–15, 2015.

James Kapaldo James Kapaldo is a Ph.D candidate in Physics at the University of Notre Dame. His current research interests include machine learning and its use for analyzing scientific experiments, experimental design and automation, the study of cold plasma jets and their effect on oral cancer cells. Kapaldo received a M.Sc. degree in physics from the University of Notre Dame studying Wigner Molecules and Ga-In intermixing in InP/GaInP quantum dots. During the summers '09 (on a DAAD fellowship) and '10 he was a research intern at Max Planck for Quantum Optics near Garching, Germany where he worked on designing and building a ultra-high vacuum beamline for attosecond laser experiments. Contact him at jkapaldo@nd.edu.

Xu Han Xu Han received a Ph.D ('17) in Physics at the University of Notre Dame. Her research interests include studying the effect of cold plasma jets on oral cancer cells, using dosimetry methods for studying the chemical changes induced by cold plasma jets, and cold plasma jet diagnostics. Contact her at xhan1@nd.edu.



Domingo Mery Domingo Mery (M01) received the M.Sc. degree in electrical engineering from the Technical University of Karlsruhe in 1992, and the Ph.D. degree (Hons.) from the Technical University of Berlin in 2000. He was a Research Scientist with the Institute for Measurement and Automation Technology, Technical University of Berlin, in collaboration with YXLON X-Ray International. In 2001, he served as an Associate Researcher with the Department of Computer Engineering, Universidad de Santiago, Chile. In 2014, he was a Visiting Professor with the University of Notre Dame. He is currently a Full Professor with the Department of Computer Science, Pontificia Universidad Catlica de Chile, where he served as the Chair from 2005 to 2009. He has authored or coauthored 60 technical SCI publications and over 70 conference papers. His research interests include image processing for fault detection in aluminum castings, X-ray imaging, real-time programming, and computer vision. He has received scholarships from the Konrad Adenauer Foundation and the German Academic Exchange Service. He received the Ron Halmshaw Award in 2005 and 2012, the John Green Award from the British Institute of Non-destructive Testing in 2013, which was established to recognize the best papers published in the Insight Journal on Industrial Radiography, and the Best Paper Award at the International Workshop on Biometrics in conjunction with the European Conference on Computer Vision in 2014. He is a Local Co-Chair of ICCV2015 (to be held in Santiago de Chile). He served as the General Program Chair of PSIVT2007, the Program Chair of PSIVT2009, and the General Co-Chair of PSIVT2011 (Pacific-Rim Symposium on Image and Video Technology) and the 2007 Ibero-American Congress on Pattern Recognition.



Supplemental Figure 1. Flat-bottomed potentials and scale invariance. **a**, without scale invariance, as the object becomes bigger the potential becomes flat-bottomed, causing the particles to move to the sides. **b**, with scale invariance, the particles have the correct position regardless of scale.

Supplemental Note 1 Connections to other optimization techniques

Our method of simulation can be thought of as being similar to simulated annealing. Simulated annealing is an optimization method to determine the global ground state of a system. It works by having a prescription of going from one state (set of particle locations) to another state, even if that state has higher in energy. In our case, we get a new state by modeling the particle dynamics and we can reach higher energy (potential energy) states because the particles have momentum. The next important step in simulated annealing is that the temperature of the system is decreased; decreasing the temperature makes it less likely for the system to go from a low energy state to a higher energy state. This is accomplished in our case by damping the particles to remove their momentum.

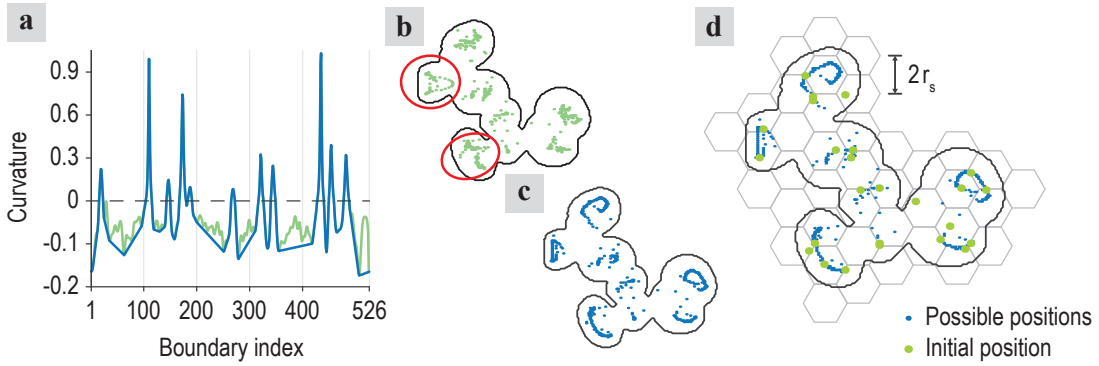
Our method can also be connected to the gradient descent algorithm with momentum. However, instead of the gradient only depending on some surface ∇V, our gradient also depends on the particle interaction. Additionally, the learning-rate in our method is adaptively set by the Runge-Kutta (2,3) method so that the relative error never increases above some threshold. With this connection to gradient descent, it could be possible to improve the speed of convergence of our method by trying to apply more advanced gradient descent algorithms to our problem, such as Adagrad [38] or Adam [39].

Supplemental Note 2 Computation of boundary curvature

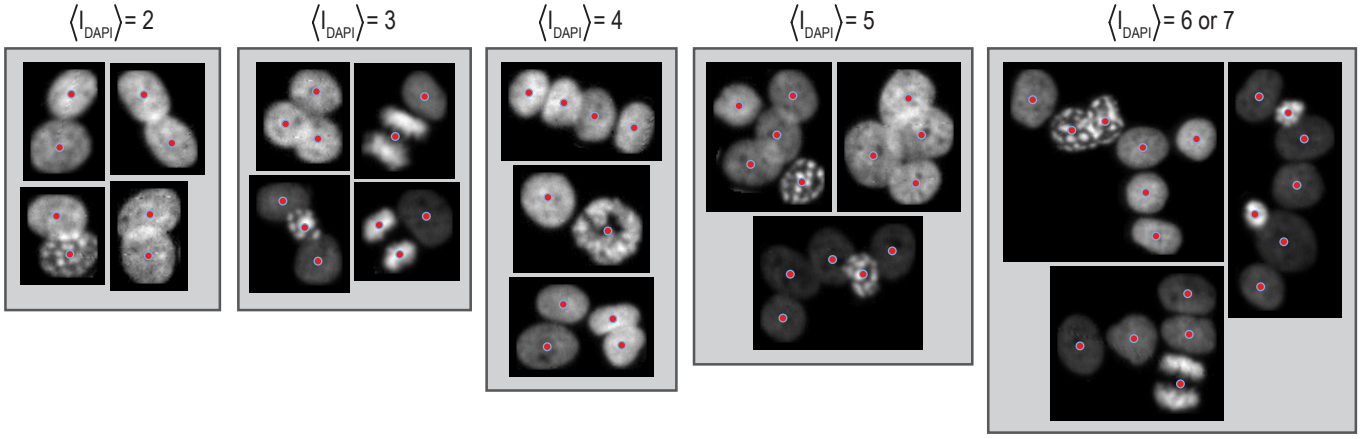
The curvature, $\kappa = (x'y'' - y'x'')/(x'^2 + y'^2)$, is computed by smoothing the boundary by convolution with a Gaussian and then taking the derivatives by convolution with derivatives of a Gaussians. The standard deviation of the Gaussians depends on the object scale, λ from (7), $\sigma = \max(1, \mathrm{round}(\lambda/9))$. This value is empirical.

Supplemental Note 3 Validation Data and Images

The nuclei images used for our experiments were acquired as follows. We cultured squamous cell carcinoma 25 (SCC25) oral cancer cells (epithelial cells from the bottom of the tough) on a coverslip and then fixed them and stained their DNA with 4′,6-diamidino-2-phenylindole (DAPI). The coverlslip was imaged in a fluorescence slide reader with 20x magnification (1 pixel = 0.47 μm, average nuclei radius 19 pixels). The entire image was segmented using adaptive log-weighted Otsu thresholding [36], which resulted in more than 100,000 objects (either single nuclei or clumps of nuclei). These objects were filtered into five groups according the normalized integrated DAPI intensity (denoted $\langle I_{\mathrm{DAPI}} \rangle$) and the size of each object; examples of the objects from each group can be seen in Supplemental Figure 3. From each group, 484 objects were randomly selected, and the center of each nuclei in all 2,420 objects were manually labeled, resulting in a count of 7,789 nuclei. The red dots in Supplemental Figure 3 are the labeled nuclei centers. As can be seen in Supplemental Figure 3, the nuclei in our data set have a variety of shapes, intensity levels, textures, and noise levels. The nuclei in our data set contain nuclei from each cell phase (G_1, S, G_2, and M) as well as apoptotic nuclei (dying).



Supplemental Figure 2. Convex hull transformed center of curvature. **a**, curvature (green) and convex hull transformed curvature (blue) versus boundary index. **b**, the set of possible initial particle positions using the curvature data. Note that along the long side of an ellipse the initial points can be far from the geometric center (red circles). **c**, the set of possible initial particle positions using convex hull transformed curvature. **d**, uniform random distribution of the possible initial positions: a hexagonal lattice is overlaid and one possible position from each hexagon is selected.



Supplemental Figure 3. Example objects. Each box shows several objects (nuclei clumps) from each of the five groups; the contrast for each object has been enhanced. Labels above the boxes give the normalized integrated DAPI intensity. The red dots represent the labeled nuclei centers, and the edge of the markers represent our expected confidence in their location (~ 3 pixel radius).

SUPPLEMENTAL NOTE 4 LOCATING NUCLEI CENTERS: PREVIOUS WORK AND METHOD OPTIMIZATION

A general strategy for locating nuclei centers is searching for circular or elliptical regions in an image. [16] introduced an iterative radial voting approach (also named multi-pass voting, MPV) where cone shaped kernels vote for likely object centers; the cones are initially directed along the image gradient and then in following iterations are directed toward the most likely object centers while decreasing the angular width of the cone. [10] modified this approach to be non-iterative, single-pass voting (SPV) by casting a vote at each image pixel with a large gradient using a cone shaped kernel with an offset Gaussian directed along the image gradient; all of the individual votes are then combined by using a mean-shift clustering algorithm. [18] further modified the SPV approach by first thresholding the image and then only using the boundaries of each object for casting votes along the image gradient.

[19] takes a different approach in using a sliding band filter (SBF) with the image gradient. The SBF works by creating an annular region of fixed width around a given pixel, and then by deforming (sliding) each angular region of the annulus along the radial direction until the dot product between a radial vector field and the image gradient is maximized. This is repeated for every pixel in the image and the resulting voting landscape created from the magnitude of the dot product is used to locate the centers of the nuclei (local maxima). [20] modified this approach to use the image phase congruency instead of the image gradient to address problems when the image contrast is low.

Another popular approach in medical image analysis [1] is Laplacian of Gaussian (LoG) filtering, where the basic principle is to convolute an LoG filter with the image and look for local maxima. [21] introduced a scale normalized LoG filter for selecting the proper LoG scale to use. [22] furthered the method by using a multiscale LoG filter with the maximum scale adaptively limited by the distance transform of the thresholded image. Generalizing these LoG filters for asymmetrical objects, [23] introduced a generalized LoG (gLog) filter where the Gaussian's covariance matrix is not the identity matrix; and, recently, [24] extended this method by creating a set of possible seed-points from each orientation of the gLoG filters and then using mean-shift clustering to combine near by seed-points.

Below we discusses the use and optimization of the seven methods we compared to. The code for these models were received directly from the respective authors or from the code freely available online. The MPV method was used as an ImageJ plugin: https://imagej.nih.gov/ij/plugins/radial-voting.html, and the SBF code was obtained from https://web.fe.up.pt/~quelhas/. The MSER implementation we used was apart of the VLFeat toolbox.

For the MPV, SPV_{Qi}, SBF, and gLoG methods, each of the nuclei images were first contrast enhanced. This contrast enhancement could be called for as our method implicitly corrected for the contrast with the adaptive Otsu thresholding [36]; note that the methods did not perform well, or as well, without the contrast enhancement. The SPV_{Xu} method, as well as the distance transform, only require the binary mask of each object; so, we directly used the mask from our adaptive log-weighted Otsu thresholding [36] for these methods.

gLoG optimization: We optimized over both the small, $\sigma_{\min} \in \{3, 5, 7, 9, 11\}$, and large, $\sigma_{\max} \in \{13, 15, 17, 19, 21, 23\}$, sigma values as well as the mean-shift bandwidth, $b \in r \cdot \{0.3, 0.4, 0.5, 0.6\}$, where $r = 19$ is the average nuclei radius in our images. The parameters that optimized the sum of the FD_0 and $F1_{\delta r=3}$ were $\sigma_{\min} = 9$, $\sigma_{\max} = 15$, and $b = 0.6 \cdot r$. In all the above trials we used 9 orientations of the asymmetric gLoG kernels.

SBF optimization: We optimized over the number of angular sections, $N \in \{8, 16, 32\}$, the width of the band, $d \in \{3, 5, 7, 9\}$, and the threshold used to find maxima, $\Theta \in \{0.4, 0.5, 0.6, 0.7, 0.8\}$. We used $r_{\min} = 8$ and $r_{\max} = 30$. The parameters that optimized the sum of the FD_0 and $F1_{\delta r=3}$ were $N = 32$, $d = 5$, and $\Theta = 0.5$.

SPV_{Qi} Optimization: We optimized over the gradient threshold, $\Theta \in \{1, 2.5, 5\}$, the maximum radius, $r_{\max} \in \{34,$

37, 40, 43}, and the bandwidth, $b \in r\cdot\{0.5, 0.6, 0.7, 0.8\}$, where $r = 19$. The minimum radius was set to $r_{\min} = r/3$. We used a radius step size of 3, a cone angle of 60° with angle step size of 3°, and a Gaussian size of $\sigma = 2$. The parameters that optimized the sum of the FD_0 and $F1_{\delta r=3}$ were $\Theta = 2.5$, $r_{\max} = 37$, and $b = 0.7 \cdot r$.

$\mathrm{SPV}_{\mathrm{Xu}}$ Optimization: We optimized over the maximum radius, $r_{\max} \in \{21, 24, 27, 30, 33, 36\}$, the ellipticity threshold parameters $e_1 \in \{0.85, 0.875, 0.9, 0.925\}$ and $e_2 \in \{0.025, 0.05, 0.075, 0.1\}$, the Gaussian size $\sigma \in \{3, 6\}$, and the bandwidth, $b \in r\cdot\{0.5, 0.6, 0.7, 0.8\}$, where $r = 19$. The minimum radius was set to $r_{\min} = r/3$. We used a cone angle of 60°. The parameters that optimized the sum of the FD_0 and $F1_{\delta r=3}$ were $r_{\max} = 24$, $e_1 = 0.875$, $e_2 = 0.075$, $\sigma = 6$, and $b = 0.7 \cdot r$.

MPV Optimization: As the MPV algorithm was implemented as an imageJ plugin (and not Matlab code), we did not systematically optimize over the parameter values as in the above cases. We used the default threshold on the point selection (1000), and set the average radius to the average radius of our nuclei, 19. The results often contained two seed-points located within ~ 3 pixels of each other. Before calculating the results, we first merged each of these double seed-points into a single seed-point.

MSER: We tried several parameters for MSER; however, none did very well; therefore, we did not perform systematic optimization.

Supplemental Note 5 Model parameters

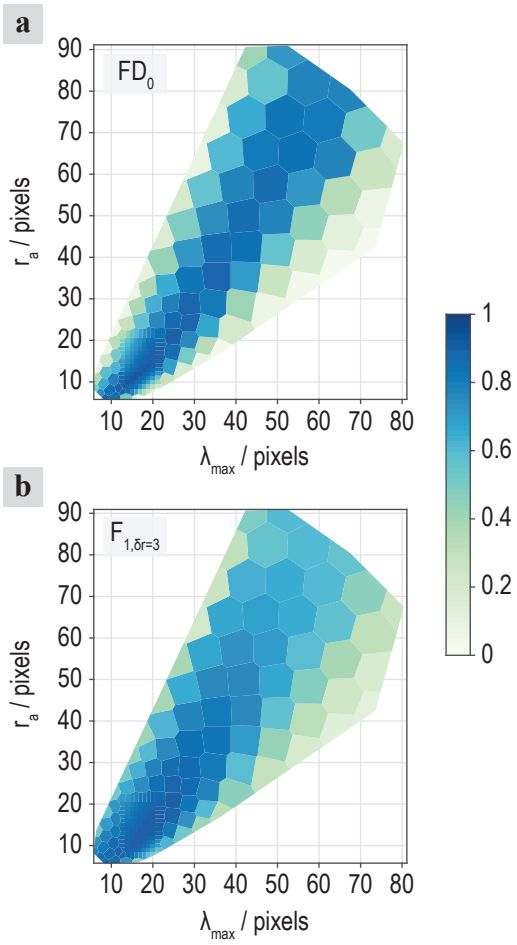
We selected the value of $\lambda_{\max}$ and r_a by evaluating the performance as these parameters are varied, and then selecting the values that maximized the sum of the $F_{1,\,\delta r=3}$ and FD_0 values. Supplemental Figure 4 shows the results of varying these parameters. Note in particular, that as $\lambda_{\max}$ increases to larger values, $F_{1,\,\delta r=3}$ decreases; this is due to the potential becoming flat bottomed as discussed before. See Supplemental Note 6 about scale-invariance for additional discussion about the attractive extent r_a.

It is not possible to solve for A, μ and σ for all possible values of d_0, r_0, and r_a in (2). r_a is the most important parameter of these three, and so the primary criterion used in setting d_0 and r_0 is that the actual location of the potential minimum and attractive extent (when A, μ, and σ are used to create the potential) are only a few percent away from the values we set. In general, we found $d_0 = -1$ to work until r_a becomes larger than 100, and we found $r_0 \approx 0.15 \cdot r_a$ works well.

The value of r_s was selected after considering its dependence on both the results and the computation time, see Supplemental Note 7 for details.

Supplemental Note 6 Scale-invariance

Here we explore the issue of scale invariance and flat bottomed potentials. To do this, we scaled the images by a factor and determined our method's performance as the attractive extent r_a is varied. Supplemental Figure 5a,b show the FD_0 value and $F_{1,\,\delta r=3}$ score when scale invariance is not implemented. You can see that as the scale increases, the



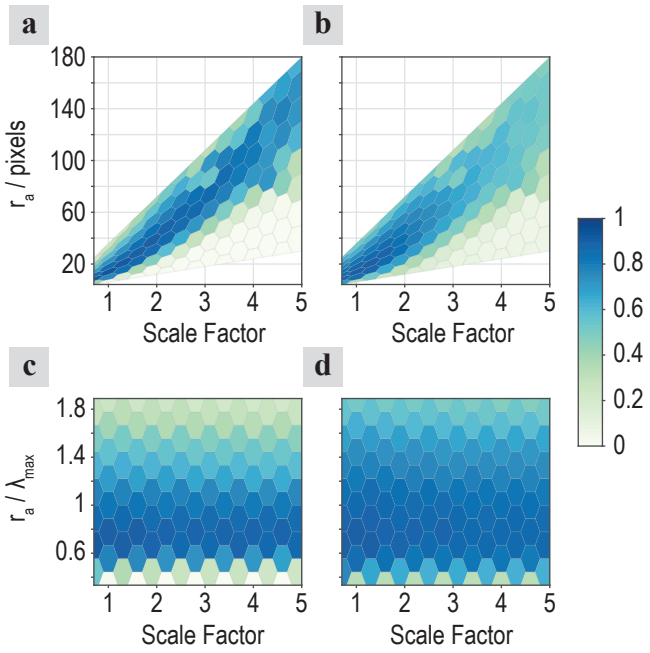
Supplemental Figure 4. **a** and **b** show the FD_0 value and $F_{1,\,\delta r=3}$ score as as $\lambda_{\max}$ and r_a are varied.

attractive extent increases almost linearly. Additionally, as the scale factor increases past ~ 2, the F_1 score decreases; this is caused by the potential becoming flat bottomed, as in Supplemental Figure 1. Supplemental Figure 5c,d show the results when scale invariance was implemented; you can see that the results are stable at all scale factors.

Supplemental Figure 5 also shows that when the attractive extent is too small or too large that FD_0 and F_1 decrease. This is because when r_a is too small, many particle clusters are formed, and when r_a is too large, only a few particle clusters are formed; in both these cases the number of seed points will be wrong which will cause their locations to also be wrong. Remembering that the average nuclei radius in our images is 19 pixels (at scale 1), Supplemental Figure 5a,b show that good results are obtained when the attractive extent is slightly smaller than the average nuclei radius. The same can be seen in Supplemental Figure 5c,d: with the interpretation that $\lambda_{\max}$ is the radius of a nuclei, which is true for an isolated circular nucleus, then the best results are obtained when the attractive extent is $\sim 0.8\lambda_{\max}$.

Supplemental Note 7 Dependence on particle initialization

We determined the dependence of our method's performance on the initial distribution of particles and the num-



Supplemental Figure 5. **a** shows the FD_0 value and **b** shows and the F_1 score at $\delta r = 3$ as the object scale and attractive extent are varied, when scale invariance is not used. **c** and **d** show the same thing when scale invariance is used. In all cases we used $r_0 = \mathrm{round}(0.12\, r_a)$ and $d_0 = \max(0, \log_{10}(r_a/100)) - 1$.

ber of particles. We used four methods of distributing the particles (listed in order of increasing estimated accuracy): random, uniform random, center of curvature (CoC), and convex hull transformed center of curvature (CvxHll CoC). The dependence on the number of particles was determined by using CvxHll CoC point selection method with a Wigner-Seitz radius varying from 2.5 to 20. The results from these trials are shown in Supplemental Figure 6. Supplemental Figure 6a shows that as the density of initial points increases F_1 increases, meaning the calculated seed-points get closer to the true points. Supplemental Figure 6c,d show that as the initial particle locations get better, both the FD_0 and F_1 values increase. Additionally, Supplemental Figure 6d shows that it is important to have enough initial particles: when $r_s = 20$, FD_0 is low and FD_{-1} is large because there are not enough particles to detect all of the nuclei centers. Increasing the number of particles with $r_s = 15$ results in a large FD_0 performance boost, and further increase leads to even better performance.

With the results improving with the number of particles, we could use as many particles as possible; however, there is a cost in computation time as the number of particles increases. (This is the reason $r_s = 5$ is our default value even though better results can be obtained with smaller values.) In Supplemental Figure 6b we show the amount of time needed to solve the differential equations per nuclei clump verses the inverse Wigner-Seitz radius. The solver time increases almost linearly with $1/r_s$, with a value of ~ 80 ms at $r_s = 5$, until it finally flattens out. We suspect the time levels off after $r_s = 3$ because all available particles are already selected. Note that the seed-point calculation of each nuclei clump is independent and can be computed in parallel.

Supplemental Note 8 Requirements of using the distance transform

The distance transform will only work to create a good confining potential if the objects are approximately the same size in each dimension. The reason is as follows: consider a long, thin 2D rectangle oriented so that the long dimension is parallel with the y-axis. The distance transform will only vary along the x-dimension, and therefore the gradient of the distance transform will be zero along the y-axis. This means that there will be no force on the particles along the y-axis pushing the particles to the center. Thus, in general, to be able to use the distance transform, either the objects must naturally be about the same size in each dimension (like 2D nuclei that are not very elliptical), or we must be able to scale the objects to have similar sizes along any direction, which will not be possible in general.

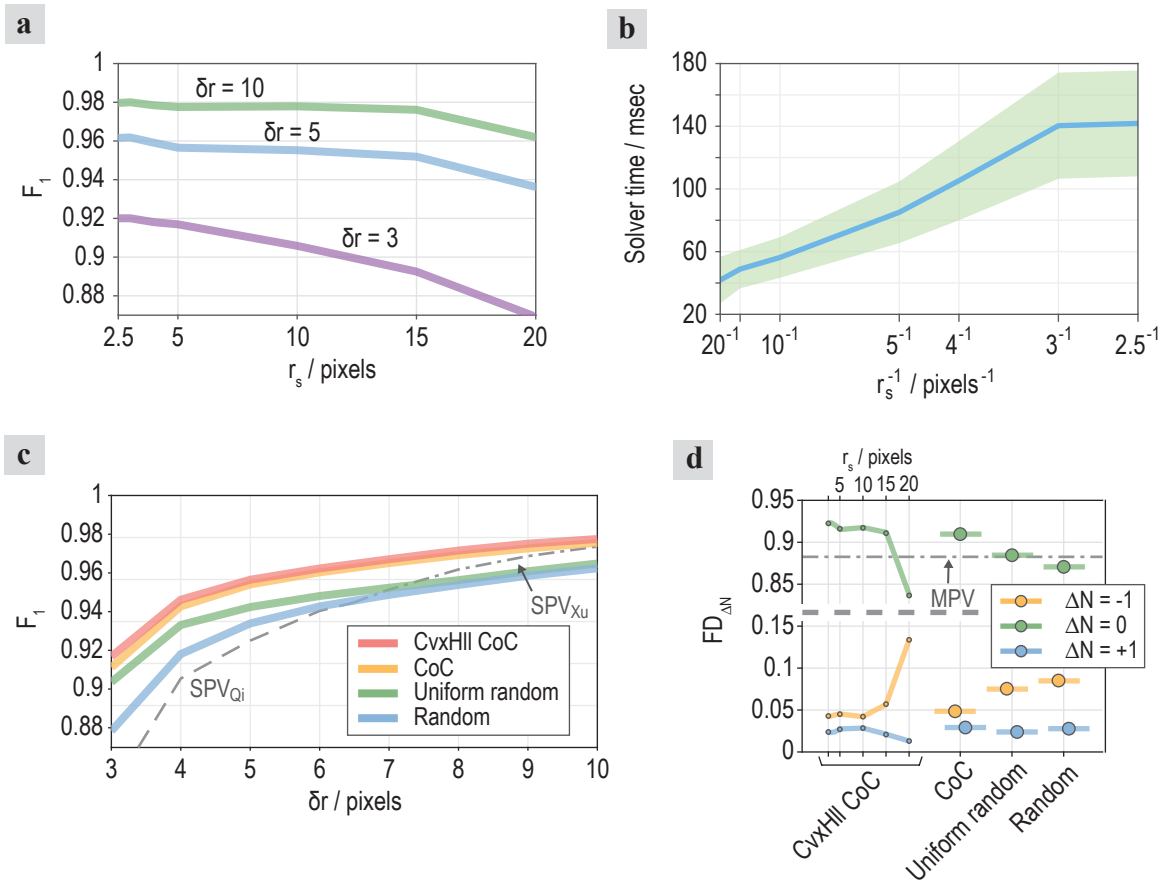
Supplemental Note 9 Dependence on particle damping rate

In this section we model the particle dynamics in the confining potential of Figure 7c using different particle damping rates, α. We modeled the particles 20 times and plot the final cluster positions of each iteration; these results are shown in Supplemental Figure 7. When the damping rate is large, $\alpha = 5 \cdot 10^{-3} t$, the particles are slowed down so quickly that they do not even form clusters, and so there are black dots everywhere. When the damping rate is a bit smaller, $\alpha = 5 \cdot 10^{-4} t$ (the same value used throughout the paper), the particle clusters are more well localized at the correct positions, but there is still some spread and a few points in-between the major clumps. If the damping rate is further decreased to $\alpha = 5 \cdot 10^{-5} t$, then there are no longer any particle clusters in-between the true positions and there is a much smaller spread in the results.

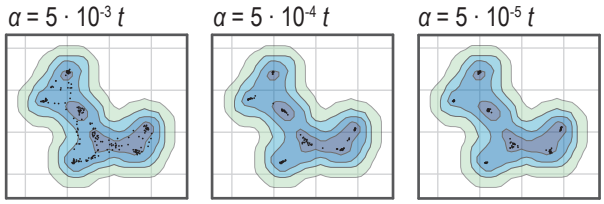
There is a cost to getting the tighter results in computation time. The average solver time for each of the three damping rates is, 91 ± 30, 136 ± 15, and 421 ± 44 ms. Thus, it is faster to simply model the particles M times with $\alpha = 5 \cdot 10^{-4} t$ and then cluster the results of the M simulations (using the same clustering algorithm we use for clustering the particles in the simulation) and keep all clusters that have more than, say, $M/3$ points in them.

Supplemental Note 10 Available changes and improvements

Asymmetric particle interaction: The first possible improvement is related to the particle interaction potential being circularly symmetric. This symmetry can lead to the foci of an ellipse being located instead of the ellipse's geometric center, when several highly elliptical objects are mixed with circular objects with the same area. This problem can be reduced using appropriate initial particle density and initial particle locations; however, a more fundamental fix to the problem has proven difficult. Two potential methods for addressing this issue are as follows: 1) Change the confining potential. Currently, the confining potential will be quite flat



Supplemental Figure 6. **a**, F_1 dependence for $\delta r = \{3, 5, 10\}$ of CvxHll CoC particle distribution method versus the density of initial particles. **b**, solver time per nuclei clump (using CvxHll CoC particle distribution method) versus the inverse particle density. Blue line show the average solver time and green shaded region gives $\pm$ one standard deviation. **c**, F_1 score versus δr for four different initial particle distribution methods. All methods used $r_s = 5$. The dash and dot-dashed lines represent the best values of the methods we compared against in Figure 5. **d**, dependence of $\mathrm{FD}_{\Delta \mathrm{N}}$ on the initial particle distribution method; additionally, the dependence of $\mathrm{FD}_{\Delta \mathrm{N}}$ on the initial particle density using the CvxHll CoC method is shown. The dot-dashed line shows the best FD_0 result of the methods we compared against.



Supplemental Figure 7. The final cluster positions from 20 iterations using three different particle damping rates.

along the major axis of a highly elliptical object because the distance transform only depends on the distance to the nearest boundary pixel; if the potential is modified so that along any line across the object the potential is changing, then the circular symmetry will not be as problematic. 2) Scale the distance between particles using information about the gradient of the confining potential. In the Hamiltonian (1), the argument of the interaction potential could be modified to

$$V_{\mathrm{int}}(|\mathbf{r}_i - \mathbf{r}_j|) \;\rightarrow\; V_{\mathrm{int}}\left(\left|(\mathbf{r}_i - \mathbf{r}_j)\cdot \hat{\mathbf{f}}(\boldsymbol{\nabla} V|_{\mathbf{r}_i,\mathbf{r}_j})\right|\right) \qquad (12)$$

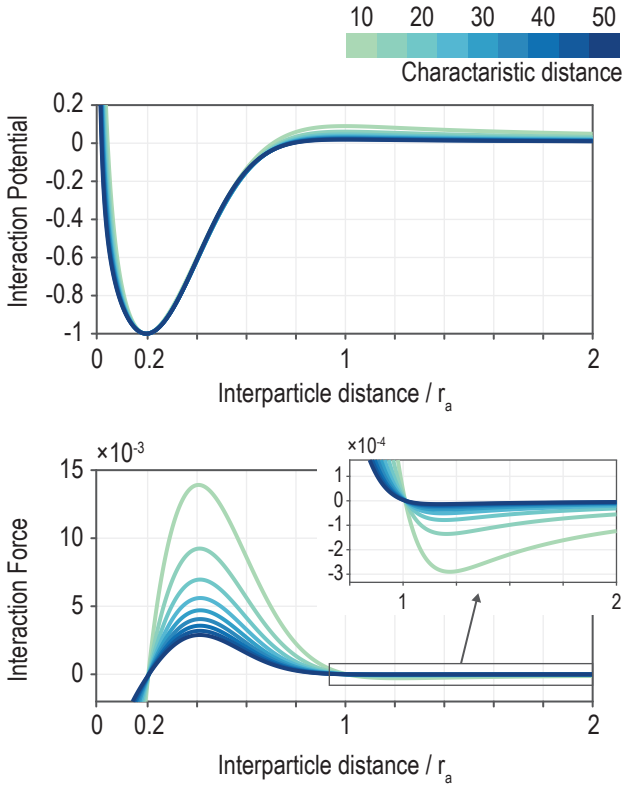
where $\hat{\mathbf{f}}(\cdot)$ is a function that takes in the gradient of the confining potential near the particle locations (and potentially along the line connecting the particles) and returns a unit vector that will scale the distance between the particles. In this way, the particle interaction can be made effectively asymmetric and dependent on the local properties of the confining potential. For example, if the confining potential is flat along the line connecting two particles, then with an appropriate choice of $\hat{\mathbf{f}}(\cdot)$ the distance between the particles can be scaled down so that the two particles are attracted to each other.

Performance metric: When locating the clusters of scatter point data, previous methods have a metric that can be used to describe how well the data was clustered; for example, k-means can use the average distance to a cluster center and mixture of Gaussians can use the log-likelihood. We currently do not have such a metric for our method, though, we suspect that one possible measure could be related to the final total energy of the particles.

Clustering: Our method is able to locate the seed-point of a cluster, but it does not actually divide the scatter point data points into different clusters. One possible method of clustering would be to assign each data point to the nearest

calculated seed-point; potentially better results could be obtained if information from the confining potential is also used.

Multiple confining potentials: In regards to biological images, when more than one florescence channel is available, a different confining potential could be created from each. One could then create several different types of particles: particle type 1 would interact with with confining potential 1, particle type 2 would interact with confining potential 2, and so on. Additionally, the particles within each type can have a different interaction potential, and (importantly) particles of different types could interact with each other. This could allow for complex data clustering and pattern recognition to be performed.



Supplemental Figure 8. The interaction potential and interaction force in data space as the characteristic distance of the solver space is changed. Note, negative force is repulsive and positive force is attractive.