**Supplementary Note 1: Example *Escherichia coli* data and segmentation/tracking results**

One dataset used to demonstrate our algorithm is collected by using an *E. coli* strain, MC4100z1 carrying pCFPT7 plasmids (Fig. S1). Specifically, we induced cells with 1000μM isopropyl-b-D-thiogalactopyranoside (IPTG) and observed them by fluorescent microscopy. 10ml of 1.5% M9 melted agar (supplemented with 0.4% glucose, 0.1% Casamino acids, and 50μg/ml ampicilin) was dropped onto a custom agar plate. Immediately after the agar solidified, 1μl cell culture was pipetted onto the agar and covered with a glass cover slip. Images were taken using a Leica DMI6000B fluorescent microscope (Leica, Bannockburn, IL) with a 455nm mercury excitation lamp and a 480±40 emission filter. Phase and fluorescent images were collected by using a 100X oil-immersion lens. Microscope chamber was maintained at 37°C.

Three movies are provided here to demonstrate the collected data as well as segmentation and tracking results:

- Movie 1: The initial time-lapse images
- Movie 2: The segmentation results overlaid on top of the original movie. Here the red channel is the original phase images and the green channel is the fluorescent intensities and the blue channel give the segmentation results. Note that, only the cell outlines are shown in the movie.
- Movie 3: This movie shows one particular track of a cell in the first image.

![Figure S1: Plasmid map of pCFPT7 plasmids. T7 RNA polymerase binds to a P_{T7Lac} promoter carrying a lac operator site, which is repressed by LacI. The gene circuit can thus be activated by a chemical inducer, isopropyl-b-D-thiogalactopyranoside (IPTG). The circuit activity is reported by a cyan fluorescent protein (CFP), co-expressed with T7 RNAP.](image)

Figure S1: Plasmid map of pCFPT7 plasmids. T7 RNA polymerase binds to a P_{T7Lac} promoter carrying a lac operator site, which is repressed by LacI. The gene circuit can thus be activated by a chemical inducer, isopropyl-b-D-thiogalactopyranoside (IPTG). The circuit activity is reported by a cyan fluorescent protein (CFP), co-expressed with T7 RNAP.
A selected subset of images from these movies is also shown in Figure S2. The top panel gives the initial images at time 1, 21 and 41 respectively. The middle panel gives the corresponding segmentation results and the bottom panel gives the tracking results of one particular cell.

Figure S2. Example image frames and analysis results.
Supplementary Note 2: Hybrid images and hybrid image based operators

1. Background

Typical segmentation algorithms take a one-way three-step approach in recognizing objects from a grey scale image: image enhancement, thresholding and object modeling. The first step in such approach is to enhance the signal strength for pixels that represent the objects in the image and reduce/de-noise the signal strength for pixels that represent the background. Many algorithms have been developed for this purpose. Some examples are mean/median/range filters, contrast enhancement and iterative erosion/dilation. All these operators take grey scale images as input and output modified/enhanced grey scale images. The second step in these approaches is to convert the modified grey scale images into black-white masks and to label the disconnected blobs as potential candidate objects. This step is typically done using (adaptive) thresholding, edge detection or (masked/hierarchical) watershed methods. A third step is usually followed to further address the over or under segmentation issues arising from previous step. Some algorithms have been developed for this purpose as well, including methods that break down blobs based on some shape models and merge certain blobs based on feature classifications and/or predictions. These approaches are one-way due to the fact that data is transformed to binary from grey scale in the process and no interactions between image enhancement and object modeling can be easily made.

We take an iterative approach to the segmentation problem to make image enhancement and object modeling interactive. At each iteration, the regions of interest that contain objects are first enhanced; then the object modeling decides how to refine the regions of interest. These coupled steps are then repeated to gradually reach a satisfactory segmentation result. The approach requires any operator used within each iteration 1) to have the same type of input and output data and 2) to operate only on the regions of interest. Here we introduce hybrid images and hybrid image based operators to address these challenges. These new approaches are used extensively in our image preprocessing and segmentation algorithms.

2. Hybrid images

A hybrid image is a combination of a grey-scale image and a corresponding binary black-white image of same size. While the grey scale image usually represents the observed data, the associated binary image represents the regions of interest in the observed data set. Let I be any grey scale image, we can always convert it to a hybrid image H by adding a trivial all 1 mask M and define H as H = (I,M). Figure S3 gives such an example of hybrid image. Figure 1a is an observed phase image from certain microscopy experiment; Figure 1b is a binary mask that outlines the regions of interest from Figure 1a. Figure 1c is the hybrid image derived from Figure 1a and Figure 1b with green area representing pixels that are masked out. Hybrid images have been used as both input and output in our segmentation algorithm for most of image operators defined in the paper, including modified filters and morphological operators.
3. Hybrid image based operators

Let \( H = (I,M) \) be a hybrid image with \( I \) denotes a grey scale image and \( M \) a binary image of same size. We define a hybrid image based operator \( F \) on \( H \) as any function or procedure that operates on values only from \( I \) that have corresponding nonzero values on \( M \) and returns a hybrid image \( H'=(I',M') \) of same size as \( H \). By definition, all existing linear or nonlinear image filters such as mean and median filters and morphological operators such as erosion and dilation are special cases of hybrid image based operators with corresponding masks having all nonzero values.

Many existing filters/operators defined on grey scale images can be extended to be hybrid image based in a straightforward way by ignoring the pixels that have zero mask values.
and adjusting the normalization factor accordingly. For example, for a typical mean filter on grey scale image, the output image is based on the local averaging of the input image in the neighborhood defined by a structure element such as a rectangle or circle. In the case of hybrid image $H = (I, M)$ and hybrid filter $F$ with output $H' = F(H)$, the output image $H'$ is still based on the local averaging of the input image, but with three modifications: 1) the structure element is locally modified by superimposing the corresponding mask; 2) local averaging is only performed on pixels that have non-zero mask values; and 3), for pixels that have zero mask values, the corresponding output pixel values can either be arbitrary or undefined, which really does not make a difference since they won’t be used by any other hybrid operators. The same approach can be extended to quantile, median, rank and range filters, which are all used in the image preprocessing and iterative segmentation steps. In the same way, we can extend the standard morphological operations on grey scale images such as dilation/erosion to hybrid images. We take the dilation operator as an example here. For a dilation operator, the output image is based on the local maximum of the input image. Just like a mean filter, the dilation operator takes two pieces of data as inputs: the image to be dilated and a structure element that defines what neighborhood pixels will be used to determine the output value. By restricting operations only on the corresponding masks, we can readily extend it to hybrid images.

The above-mentioned hybrid operators mainly focus on changing values on the corresponding masks. It is also possible to define operators that change the masks directly as well. In this case, we could apply the existing binary operators such as thickening/thinning directly to the masks as they are binary already by definition. In the case of shrinking masks, as in the cases of morphological thinning or binary erosion, we don’t have to make any changes to the corresponding image; in the case of enlarging masks, as in the cases of morphological thickening or binary dilation, we have to impute values for pixels that just changed mask values from 0 to 1 in some way. One strategy we used in our segmentation algorithm is to have a ‘reference image’ that we simply take a corresponding value from there whenever it is needed. Such an example reference image is just the original input image. Another example is to use the most recently defined value available in the context of an iteration procedure.

4. Some example results using hybrid operators

Here we provide some examples using the hybrid operators described in previous section. Figure S4a shows an image from the data described in Supplementary Note 1. The image is first converted to hybrid image by adding a trivial mask with all values setting to 1, indicating all observed pixel values are in the regions of interest. Figure S4b is the result of applying a hybrid range filter to Figure S4a with the range threshold value 30 and disk size 10 for the structure element. Figure S4c is the result of applying a hybrid morphological erosion operation to Figure S4b, using the same disk structure element from Figure S4b. The operation is done by first applying the usual morphological erosion to the corresponding mask in Figure S4b and then modifying the corresponding grey-scale image by setting pixel values to 0 for newly masked ones. Figure S4d is the result of applying a hybrid high-pass range filter to Figure S4c. A range threshold value 70 is
used so that any pixel with a range value above 70 is filtered out from the grey-scale image and is added to the corresponding mask. The same disk structure element is used again here. Figure S4e is obtained from Figure S4d by applying a **hybrid quantile filter**, using the same structure element and an upper quantile threshold of 75%. Figure S4f is further derived from Figure S4e by hybrid **morphological thickening** using Figure S4d as reference image.

Figure S4: Filter examples
Supplementary Note 3: Object modeling in iterative segmentation

Object Modeling
Given a sequence of images, we usually have some kind of assumptions or prior information about the objects (cells in our context) for which we are looking. Most assumptions on objects are made about either morphological features or intensity value distributions. For example, we assume that the cells described in Supplementary Note 1 have a) certain morphological features such as dimension, shape and volume; and b) generally darker interior with brighter surrounding background. These assumptions are then enforced explicitly or implicitly in segmentation algorithms to separate objects in foreground from the background and to further address the over and under segmentation issues.

Making assumptions and developing algorithms based on that is the subject of object modeling, which plays important roles in our proposed iterative segmentation approach in analyzing time-lapse images from Supplementary Note 1. First, steps taken to manipulate input hybrid images during each iteration are tailored to enforce assumptions made about cells. Second, scores employed to decide if blobs need further attention are calculated based on the same assumptions. More details are discussed in the following sections.

Assumptions and Prior information about cells
The following assumptions are being made about the E. coli cells to be extracted from image sequences in our analysis. Most of them are implicit to many other segmentation and tracking algorithms but explicitly listing them here helps to understand the portability when used in other cell types. In the case that all cells are straight (as in the case in Supplementary Note 1), a more general cell model described in Supplementary Note 6 can also be applied.

   We assume that all cells have straight or curved cigar shapes, and can be modeled as straight or curved cylinders with two half spheres on both ends respectively. Furthermore, the center line/curve of such cell can be represented by a polynomial of degree n, with n lies between 1 and 3.
2. Cell dimensions.
   We assume that all cells have minimum and maximum widths in terms of pixels. These widths represent the minimum and maximum cylinder diameters when cell shapes are modeled as in 1).
3. Cell volumes.
   We assume that all cells have minimum volumes in terms of pixels.
   We assume that all cells have relatively smooth shapes.
5. Image background intensity value distribution.
   We assume that large background regions in images have relatively small variations in intensity values as compared to nearby cells.
We assume that the pixels representing cells have darker interior regions and relatively brighter border regions.

Prior information about these assumptions is required to utilize the proposed iterative segmentation algorithm. As an example, the following values are used in the image analysis for the E. coli dataset from Supplementary Note 1.

1. We assume that all cells have cigar-like shapes and cell centers can be modeled as straight lines.
2. We assume all cells have a minimum width of 15 pixels and a maximum width of 40 pixels.
3. We assume all cells have a minimum volume of 200 pixels, which approximately equals to the area of a circular disk with radius 8.
4. We assume all cells have smooth boundaries so that a morphological opening operation using a 5 by 5 disk structure element won’t change the cell shape.
5. We assume that the intensity values in the neighborhood (defined by a disk with radius = maximum cell width) of any background pixel have a maximum range of 15.
6. We assume that all intensity values in any cell have a maximum range of 70.

The above prior values are usually obtained manually from similar datasets and do not vary much for repeating experiments.

**Using prior information in the iterative segmentation algorithm**

Prior information is used in both image preprocessing step and iteration steps. Most often, they are used as passing parameters for the hybrid filters and morphological operators when appropriate. For example, the maximum cell width in 2) and maximum background intensity value range in 5) are used as input parameters for the hybrid range filter in preprocessing step to identify initial background regions. We also use prior information and model assumptions to create an object scoring procedure to decide if a blob needs further segmentation during each iteration. More details are giving as follows.

At the end of each iteration in our segmentation algorithm, we score and evaluate all labeled blobs to decide if they are subject to further segmentation or they are already well-identified cells. First, each blob is thinned using morphological thinning operation to obtain estimates of the central pixels. Under the cell shape assumption, these central pixels would lie closely to a straight line segment if the blob is indeed a cell. We estimate the straight line by fitting a simple linear regression model using the central pixel coordinates. Although the goodness of fit measure from the regression model is a natural candidate to evaluate how well those points fitting within a straight line, we use the error standard deviation as score instead, due to the fact that we can intuitively interpret it as the average departure in pixels of a central pixel from the straight line. Second, all blobs are thresholded into two subclasses using a user chosen threshold value, which is 5 pixels for the images in Supplementary Note 1. As a result, all blobs in the subclass with score less than 5 are considered as identified cells and those blobs in the subclass with higher scores are subject to further segmentation.
When cell shapes are modeled differently, we will need to modify the above score procedure accordingly. For example, if we assume cells have curved cigar-like shapes, we might want to fit a cubic regression model instead of a linear regression model. We illustrate this through Figure S5 and Figure S6, using a different dataset. Figure S5 shows a typical image from this dataset. Evidently, it is not appropriate to assume all cells are straight here. Figure S6a shows a blob identified by the segmentation algorithm from the image in Figure S5. Figure S6b shows the central pixels for Figure S6a, obtained by morphological thinning operation. Figure S6c shows the result after fitting central pixels in Figure S6b to a cubic regression model using Matlab statistical tool box. The blue crosses are the central pixels from Figure S6b; the green line is the fitted cubic curve. With a score of 16.95, well above our threshold 5, we conclude this blob is not a cell and is subject to further segmentation.

Figure S5: A typical image with curved shape cells

a.                                                                 b.                                                                 c.
Figure S6: An example using cubic regression model
Supplementary Note 4: Flowchart of iterative segmentation algorithm

Figure S7 gives a general flowchart of our proposed iterative segmentation algorithm.

Figure S7: Flowchart of the iterative segmentation algorithm
Supplementary Note 5: Flowchart of neighborhood based cell tracking algorithm

Figure S8 gives the flowchart of our neighborhood based cell tracking algorithm.

Figure S8: flowchart of the neighborhood based cell tracking algorithm
Supplementary Note 6: Cell segmentation/tracking of yeast and human cells

Sigal et al. (Nature Methods 3, July 2006) and Gordon et al. (Nature Methods 4, January 2007) demonstrated segmentation/tracking results on yeast and human cells using the currently best available software packages Cell-ID and CellProfiler (Cell 130, September 2007). The Sigal et al. study provided access to the human cell imaging data in terms of movies (in their supplementary note) as well as summaries of cell tracking results in the primary article. This human data example provides a nice context for a side-by-side comparison with our methodology in a second context in addition to our bacterial examples. A core motivation for our work has been our ongoing interests in mammalian cell studies so this is an important comparative evaluation. We further applied our algorithms to a budding yeast data set (DiTalia et al. Nature 488, 2007) to test the portability of our algorithms to different cell types. All results are reported in the supplementary note.

Human cell imaging dataset

The histone H2AFV fluorescence single-cell imaging movie was downloaded from http://www.nature.com/nmeth/journal/v3/n7/extref/nmeth892-S10.mpeg and all 249 distinctive time course images extracted using the Matlab image processing tool box. These images were then cropped to closely match the image dimensions shown in the Sigal et al paper. It is possible that the image frames obtained this way have lower resolution than those from the original paper, thus degrading our analysis. With this in mind, we nevertheless assume they are of the same resolution and base our comparison on this assumption. More details about the biological aspect of the data can be found from the original paper.

Budding Yeast imaging dataset

The budding yeast image data (120 images) was provided by Dr. Stefano Di Talia and Dr. Eric D. Siggia. We are grateful to them for discussion and provision of this data set.

Cell segmentation

Our iterative segmentation algorithm, as described in Supplementary Note 3 and 4 for E. coli example in Supplementary Note 1, was applied to these data sets following changes to prior parameter values and some cell model assumptions to reflect the context of different cell types.

A) Cell shape: Rather than assuming cells have straight or curved cigar-shapes, we assume that cells have convex or near convex borders and change the cell score procedure accordingly. More specifically, for each blob identified by the iterative segmentation algorithm, the automated analysis calculated: 1) the blob area as the number of pixels in the blob; 2) the blob convex area as the number of pixels in the corresponding convex hull for that blob; and 3) the blob perimeter as the number of pixels from blob boundary. The blob score is then the difference of blob convex area
and blob area divided by blob perimeter. A convex hull for a blob is the smallest convex polygon region that can contain the blob.

**B) Cell intensity value distribution:** Using the same basic assumption that pixels representing cells have darker interior regions and relatively brighter border regions, we also assume that a small percentage of pixels in the cell interior region could have brighter values than those of border pixels. Consequently, we use a hybrid border-rank filter instead of a hybrid rank filter as we did in the *E. coli* example to further break down high-scored blobs. A hybrid border-rank filter is similar to a hybrid rank filter but it only filters out the pixels in the border region of a blob that have relatively brighter pixel values.

It is noted that in case while *E. coli* cells can be modeled as straight cigar-shaped (thus convex), as the case in our examples, the modified model can also be applied to them directly.

**Cell Tracking**

Our automated cell tracking algorithm described in Supplementary Note 5 for *E. coli* examples was applied directly.

**Results**

Five movies are provided here to demonstrate our segmentation and tracking results for the above human and budding yeast cell data. These are to be compared with corresponding results from the original Sigal et al and DiTalia et al papers. The substantial improvements are very clear.

- **Movie 4:** The segmentation results overlaid on top of the original images for human cell data. Here the red channel is the original histone H2AFV fluorescence images and the green channel gives the segmentation results. Note that only the cell outlines are shown in the movie.

- **Movie 5:** This movie shows one particular track of a cell in the first image for human cell data. Full results can be provided upon request.

- **Movie 6:** The original budding yeast data from DeTaila and Siggia, cropped to remove some background regions.

- **Movie 7:** The segmentation results overlaid on top of the original budding yeast data, with similar settings as in movie 4.

- **Movie 8:** This movie is similar to movie 5, but shows the corresponding result budding yeast.