



OPEN

## HARLEY mitigates user bias and facilitates efficient quantification and co-localization analyses of foci in yeast fluorescence images

Ilya Shabanov✉ & J. Ross Buchan✉

Quantification of cellular structures in fluorescence microscopy data is a key means of understanding cellular function. Unfortunately, numerous cellular structures present unique challenges in their ability to be unbiasedly and accurately detected and quantified. In our studies on stress granules in yeast, users displayed a striking variation of up to 3.7-fold in foci calls and were only able to replicate their results with 62–78% accuracy, when re-quantifying the same images. To facilitate consistent results we developed HARLEY (Human Augmented Recognition of LLPS Ensembles in Yeast), a customizable software for detection and quantification of stress granules in *S. cerevisiae*. After a brief model training on ~20 cells the detection and quantification of foci is fully automated and based on closed loops in intensity contours, constrained only by the a priori known size of the features of interest. Since no shape is implied, this method is not limited to round features, as is often the case with other algorithms. Candidate features are annotated with a set of geometrical and intensity-based properties to train a kernel Support Vector Machine to recognize features of interest. The trained classifier is then used to create consistent results across datasets. For less ambiguous foci datasets, a parametric selection is available. HARLEY is an intuitive tool aimed at yeast microscopy users without much technical expertise. It allows batch processing of foci detection and quantification, and the ability to run various geometry-based and pixel-based colocalization analyses to uncover trends or correlations in foci-related data. HARLEY is open source and can be downloaded from <https://github.com/lnilya/harley>.

Live and fixed cell fluorescence microscopy is a common approach utilized by biologists to elucidate understanding of cellular function. The ability to accurately identify and quantify objects (i.e., foci) in microscopy data is thus key, and yet routinely papers are published with microscopy data that suffers from a lack of rigorous, unbiased analysis of foci of interest. An accurate measure of foci number, size, intensity, geometry or colocalization with other cellular foci or compartments may often reveal variations in intriguing underlying biological phenomena. However, variation in the aforementioned foci properties, and in cellular background signal or other relevant cellular contexts, can make foci quantification resistant to efficient, automated, analysis. Several approaches to quantifying cellular microscopy data have thus been developed as briefly outlined below.

**Threshold based methods.** Threshold based methods assume that features of interest have a higher or lower intensity than the background surrounding them. Identifying such thresholds leads to a binary segmentation in feature and background areas. Various morphological and denoising operations are usually combined to obtain a good segmentation.

Top-hat or H-maxima transformations are typical choices employed for denoising, detection or accentuation of features by algorithms like FoCo<sup>1</sup>, FociCounter<sup>2</sup> or CellProfiler's<sup>3</sup> "Speckle Counting" pipeline. Statistical thresholding like Otsu's method or many other alternatives are then used to yield a final threshold. These methods are readily extendable to 3D data as demonstrated by FocAn<sup>4</sup>.

Department of Molecular and Cellular Biology, University of Arizona, Tucson, AZ 85721, USA. ✉ email: [ilyashabanov35@gmail.com](mailto:ilyashabanov35@gmail.com); [rbuchan@arizona.edu](mailto:rbuchan@arizona.edu)

Morphological operations often require a kernel (i.e., a small search pattern image) that presupposes a shape of the feature in question. In most cases a disc of a set radius is used, limiting the algorithm to detecting round foci. More importantly, thresholding-based techniques are sensitive to a non-uniformly distributed background signal. For example, in the analysis of stress granules (SGs) in *S.cerevisiae*, vacuoles, which have no cytoplasmic signal, and vary greatly in size, can impair the identification of the actual cytoplasmic background signal.

**Kernel based methods.** In a very general sense kernel-based methods identify local features of an image by scanning through the image and scoring positions according to their similarity to an expected kernel image. A convolution operation between kernel and image yields a similarity map. Portions of the image correlating most closely to the kernel receive high scores and thus can be identified as features of interest by using thresholding or local maxima detection. FociQuant<sup>5</sup> detects kinetochores (and other foci) and BUHO<sup>6</sup> detects SGs using this general method.

The main limitations of kernel-based methods are the kernels themselves, since they offer a somewhat rigid representation of what a feature is. Different shapes, rotations or scales require multiple kernels and may rely on weight or threshold parameters for the final result. Convolutional Neuronal Networks (CNN) solve some of these problems by learning the shapes of the kernels from a dataset and performing convolutions at different scales. CNNs however require large amounts of labeled data to be trained and can be computationally expensive.

**Differential scale-space methods.** For a given image  $I(x, y)$  differential scale-space methods generally first construct a stack of images smoothed at different strengths (or scales)  $t$  called a scale space  $L(x, y, t)$  and then look for extrema of differential features (e.g., any combination of partial derivatives of  $L$ ) therein. Detected maxima thus not only contain spatial coordinates of the feature, but also a scale corresponding to the size of the feature making them scale invariant.

The most common differential operators are the Laplacian of Gaussian (LoG), its approximation Difference of Gaussian and the Determinant of the Hessian Matrix (DetH). A significant number of other operators have been proposed, described in Ref.<sup>7</sup>. The challenge with these methods arises in delimiting a boundary of the feature, since they inherently only detect points. One solution is to define a region that is convex around the detected point and define the boundary along the zero crossing of the Gaussian curvature as proposed by Ref.<sup>8</sup>.

These methods detect many false negatives that usually require a threshold parameter to be removed. Another problem with methods like LoG and DetH is that they yield the highest values for round features; irregular and elongated shapes therefore pose a problem in noisy data. Furthermore, in cell images with strong background signal the curvature of the cell outline itself starts contributing to a set of false negatives as the scale is increased.

**Machine learning methods.** Machine Learning approaches typically work on raw images, and given labeled datasets, learn automatically how to extract and classify the features of interest. As in our approach (described later) they can be easily combined with classical computer vision methods to increase performance or reduce the amount of data needed for training.

Many architectures have been proposed for various tasks, with CNNs being commonly used. DeepFoci<sup>9</sup> utilizes U-Net, a specific CNN architecture to detect and segment foci and nuclei with promising accuracy in a fully unsupervised manner. A combined approach in<sup>10</sup> uses classical techniques to detect regions of interest, extracts an over abundant set of features using filters and feeds these results into a classifier. FindFoci<sup>11</sup> uses a machine learning approach to devise parameters to filter out foci of interest in accordance with a human experimenter.

Deep Learning approaches can also leverage context. Having the whole image as input, information about location inside the cell and from foci to foci can be incorporated. Unsurprisingly, these approaches usually outperform previously mentioned methods. This however comes at a cost of long training times and the need for large, labeled datasets. To our knowledge there are no deep learning-based approaches specific to yeast SGs.

**Other methods.** In theory many of the methods can be combined to create a more nuanced quantification. An example of such work is found with Obj.MPP software<sup>12</sup> that uses a marked point process framework. A set of parametrically defined objects like disks, squares, or ellipses is compared via different quality functions to the image. These quality functions process differential (e.g., gradient) information along the edges of objects and around them. AutoFoci<sup>13</sup> combines a top-hat transformation, an estimate of local curvature (using LoG) and a foci shape derived coefficient to create an Object Evaluation Parameter.

Several issues regarding analysis of microscopy data remain unresolved. First, excluding manual classification (e.g., in Fiji) or generic approaches (e.g., CellProfiler), the identification of SGs, or other cytoplasmic foci with significant cytoplasmic background signal has seen very few tailored automated approaches. Second, as reported in an exhaustive study on bioimaging informatics tools<sup>14</sup>, overall usability is a significant hurdle for the use of automated methods, making CellProfiler one of most popular, yet not necessarily optimal<sup>9</sup> methods. Finally, in biological publications the issue of human variability and error is often not addressed. A previous study<sup>11</sup> showed with 3 users that in manual counting of  $\gamma$ H2AX foci, 20–30% percent of assignments are unmatched between any two experimenters. While there is no evaluation of how consistent the same human is and the use of the “F1 score” (a typical measure to compare two classification results) for unbalanced datasets can be somewhat misleading<sup>15</sup>, this study is very indicative of significant human bias. A similar study on  $\gamma$ H2AX foci with 5 experimenters from 3 labs<sup>16</sup> reported 81.1% of assessments between experts to deviate in a statistically significant manner.

Here, we asked 7 users of varying degrees of self-identified expertise to quantify a yeast SG dataset twice and observed a striking 3.7-fold variation between users in terms of foci calls. Furthermore, users could only replicate 62–78% of their individual quantification performance when re-quantifying a subset of the SG dataset.

This motivated us to develop a new user-friendly framework, cell, and foci detection software termed HARLEY (Human-Augmented Recognition of LLPS Ensembles in Yeast) which enables manual or trained quantification of SG and other foci-like structures or organelles. HARLEY is freely available on Github (<https://github.com/lnilya/harley>). In this paper, we describe its mechanism of action, and how users may utilize it for their yeast microscopy data analysis tasks. The software also allows for various analyses of colocalization, pre-processing, denoising and other common yeast microscopy related tasks.

## Materials and methods

**HARLEY installation and Github repository detail.** A readme document with installation instructions for HARELY, which runs in the Google Chrome browser, is present at <https://github.com/lnilya/harley>. Here, any bugs or troubleshooting issues may also be reported (Issues and Discussions tabs). Note that some of the walkthrough videos were recorded with earlier versions of HARLEY than the current release (1.2.4), thus minor, mostly aesthetic differences may be present; these do not detract from explanations given. To further aid in explanation of the parameters of HARLEY, which already utilizes help pop-up boxes for each parameter, we have provided a more detailed discussion of these parameters in supplementary methods S1.

**Yeast SG induction, microscopy data capture and applicability of HARLEY.** BY4741 yeast transformed with pRB1 (Pab1-GFP, Edc3-mCh) were grown to mid-log phase in -Ura synthetic dropout media, and subject to  $\text{NaN}_3$  stress as previous described<sup>17</sup>. Live yeast cells were imaged on glass slides (Globe Scientific, cat: 1301) with 1.5 coverslips (VWR, cat: 48366-227). Z-stack data (10 slices, 0.4  $\mu\text{m}$  each) was collected using a 100X oil-immersion (1.515; Cargille, cat: 16245) objective (NA 1.40) with 1.6 $\times$  auxiliary magnification on a Deltavision Elite widefield microscope equipped with a 15-bit PCO Edge sCMOS camera. Pab1-GFP was imaged with a 0.2 s exposure at 100% transmittance (Excitation 475/25 nm, Emission 525/36 nm) while Edc3-mCh was imaged with a 0.15 s exposure at 100% transmittance (Excitation 575/25 nm, Emission 632/60 nm). Images were subject to deconvolution using standard parameters (Enhanced ratio, 10 cycles, medium filtering) using Softworx 7.0.0 (Deltavision software), followed by maximum intensity Z-stack projection. While Deltavision Elite images have a bit depth of 0-32768, images of any bit depth equal or greater than 256 (i.e., 8-bit) should work well with HARLEY. A brightfield light image was also collected at the Z-stack center to facilitate cell boundary detection. HARLEY has only been tested on data generated from the Deltavision Elite. However, given its design principles (see later), we see no reason why equivalent data generated from many other microscopy platforms (widefield, confocal, super-resolution  $\pm$  deconvolution) with comparable or superior resolution and sensitivity should not also yield satisfactory results.

**Software versions.** Our software is based on ReactJS 17 for its frontend and Python 3.8 for its backend. Exact versions of employed packages can be found in the respective package files in our Github repository at <https://github.com/lnilya/harley>.

**Recruitment of users and dataset.** 7 independent users with familiarity in quantifying SG data were recruited by the PI to quantify SGs in a collection of 55 individual cell images, using a web-based app we developed (see the labeling step of the model training pipeline in HARLEY; Supplementary video S3). This allowed users to both label and delimit a boundary around the periphery of each putative SG foci.

On a scale of 1 = Layman, 2 = Intermediate and 3 = Expert, only the user “R” self-classified as expert, with all others being intermediates.

Foci candidates were determined using our contour loop method. After labeling a candidate (consisting of shortest and longest contour loop) as a focus, users could choose a contour loop most closely matching their perception of foci outlines by dragging the mouse across the focus in an interactive user interface (UI). The same method is used in the model training step of our software and can be explored further there.

A second round of quantification of the same cells, randomized in order from the previous scoring, was used to assess intra-user quantification variability.

Two users (A and M) failed to delineate boundaries around foci and only clicked on the foci in question; we therefore excluded their results in foci size analysis.

**Model evaluation using Matthews correlation coefficient.** For a set of identified foci candidates, a decision needs to be made whether a candidate is a feature of interest, leading to a binary classification problem with a heavily unbalanced dataset (i.e., several times fewer foci than non-foci). As described previously<sup>15</sup>, the commonly employed measures of Recall, Precision or their combination (the F1 score; also known as Dice coefficient) to compare these classifications are poorly suited when the dataset is unbalanced. To compare the results of models and users we therefore use the Matthews Correlation Coefficient (MCC)<sup>18</sup>, a much more robust measure for unbalanced datasets.

The MCC is defined as a function of the confusion matrix consisting of the four numbers:

- *TP*: True positives—number correctly classified foci
- *TN*: True negatives—number of correctly rejected foci
- *FP*: False positives—number of wrongly classified foci
- *FN*: False negatives—number of wrongly rejected foci

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}$$

Values range from -1 to 1, with 1 being perfect classification, 0 being the correlation of two completely randomized experiments (e.g., coin toss) and -1 being perfect misclassification.

**Cross validation MCC.** When comparing the results of a user's classification with the results of a model trained on this same data (i.e., the user's model) it would be unwise to simply use the MCC, since an overfitted model would yield over optimistic scores.

We therefore use the mean of a sixfold cross validation error over 6 models trained on the respective subsets of data. K-fold cross validation is a technique that splits the dataset into k disjunct sets of equal length  $D = \bigcup_i D_i$  with  $D_i \cap D_j = \emptyset \forall i \neq j$ . We then train the model on the set  $D_{ni} = \bigcup_{k \neq i} D_k$  and evaluate the MCC on the prediction of the remainder of the dataset  $D_i$ .

$$CVMCC = \frac{1}{k} \sum_i MCC(h(D_i), m_{ni}(D_i))$$

This denotes the user prediction and the model prediction trained on dataset  $D_{ni}$  with  $h(D_i)$  and  $m_{ni}(D_i)$  respectively.

The resulting correlation will be lower but a more generalized measure of model performance. We use this score only when comparing the user to their own model. In all other cases the model is trained on the whole dataset and its prediction is then compared using the regular MCC.

**Training of hyperparameters.** Support Vector Machines use a regularization parameter C that governs the size of the margin separating the classes. Since we are using radial basis functions as our kernels, the parameter  $\gamma$  governs the size of this kernel.

A typical method to find these parameters is to optimize them with regards to the Cross Validation error, since it avoids overfitting them to the data. We use a logarithmic grid search with values between  $10^{-3}$  and  $10^3$  for C and  $10^{-8}$  and 10 for  $\gamma$  to find parameters that fit the data better. In practice we found this step to give only very minor improvements over the default settings of 1 and 'scale' for C and  $\gamma$  respectively from the sklearn.svm python implementation.

**Random quantifier.** We use the MCC with a random quantifier (Rnd) as a first measure of human variability in a user's quantification efforts. For each cell, Rnd randomly picks the same number of foci as the user has chosen, resulting in a "classified" dataset which can now be compared to the user's result using the MCC. This comparison is repeated 50 times and the mean is used as the final score, reported in this publication.

**Feature space.** Our feature space is built by first obtaining the closed contours of a given length (as defined by experimenter and scale of images), then predicting the size of the foci with our algorithm and extracting a set of features based on expert knowledge from these values: (1) maximum, minimum, mean intensity of focus; (2) area, eccentricity and solidity; (3) size in relation to cell size; (4) brightness ratio predicted outline and center; (5) brightness ratio longest outline and center; (6) absolute difference in brightness center and outline; (7) normalized distance to cell center; (8) brightness index inside cell (1 brightest in cell, 0 dimmest in cell). These choices were also partially motivated by ease of extraction and speed of execution. The resulting system runs in real time along the user's labelling giving instant feedback on the test and training errors and converges quickly in practical terms. The feature space is whitened and rescaled 0–1 before processing by the SVM. After detection the initial foci intensities can be exported by the software.

**Extraction of cell boundaries from bright field images.** Since cells in fluorescence microscopy may have a varying degree of background signal that could be used to delimit single cells, we use brightfield images to extract cell outlines. The process yields a binary mask that is used with fluorescence images to extract and normalize data for single cells.

**Ridge detection.** Our algorithm first uses the Frangi ridge filter<sup>19</sup> that is then thresholded to give the first draft of cell outlines as a binary mask. A graphical UI guides the experimenter throughout the whole process allowing easy correction of mistakes and cell-identifying parameters.

**Cleaning.** A cleaning step is then employed that removes blobs in that mask that do not match certain criteria. These criteria are size or area constraints, as well as bounds on eccentricity and solidity of blocks available in the skimage.measure v0.19.0 package, yielding a more refined outline of cells. In practice, these parameters once set can be reused with very little modification from image to image.

**Ellipse fits.** The cleaned outlines are then thinned<sup>20</sup> to give a skeleton image of the outlines. We now evaluate a set of pixel positions  $C = \{x_{i,j}\}$  in the image whose distance to the skeleton lies between  $r_{\min}$  and  $r_{\max}$ ; these are parameters on minimum and maximum cell size.

For each of these candidate points we fit an ellipse that approximates skeleton points around  $\bar{x}_{i,j}$  and calculate an approximation error as a sum of squared distances between the skeleton and the resulting ellipse. Additionally, we can filter out points where the skeleton lacks points in a range of angular directions, e.g., the outline is not a closed shape.

The result is a heatmap with local minima representing locations of cell centers. It is important to note that parameters defined here depend on the magnification of the images and are transferrable between images.

**Boundaries.** The final ellipses at these minima of approximation error are then “snapped” to the boundaries using a self-developed, modified version of the active contours model<sup>21</sup> to yield final cell outlines. These are prone to noise and strong internal edges, like vacuoles or nuclei, resulting in only partial recognition of the cell. These errors can be manually rejected by our interactive UI.

Even though this method works well (Supplementary Video S1) and proposes a semi-automated way of detecting cells that in our test images is more than sufficient to gather and review hundreds of cells in minutes, we acknowledge that it probably will not outperform state of the art deep learning techniques. However, the typical number of cells quantified in yeast microscopy research is in the magnitude of hundreds and we therefore did not investigate better methods further.

**Preprocessing of fluorescence images.** Fluorescence images are first stacked using a max-projection and then denoised using Non-Local Means Denoising<sup>22</sup>. Both steps are guided through an interactive UI (Fig. 2b and Supplementary Video S2). Since noise levels and image stacks vary between experiments and microscopes, these settings are selected image by image.

**Ethical approval.** This study is exempt from the need for ethical approval as per advice received from the Institutional Review Board of the University of Arizona.

## Results

### Manual quantification of yeast SG data generates large intra and inter-variability amongst users.

7 users of intermediate to expert familiarity with SG quantification identified SG foci and their boundaries in a dataset of 55 yeast cells exposed to  $\text{NaN}_3$  stress. These users repeated their quantification 1–2 weeks later with a shuffled version of the dataset. Many users exhibited significant differences in foci identification (Fig. 1a). User foci sections varied from 21 to 79% (3.76-fold) of available foci (see “Materials and methods”) on average (Fig. 1b), while average SG foci radii selections ranged from 108 to 152 nm (Fig. 1c). Foci calls from the first and second quantification run by users resulted in a reproducibility range of 62–78% (Fig. 1d; MCC, see “Materials and methods”). These results demonstrate how variable quantification between users can be, and that even the most conservative users (those who score fewer foci on average) have difficulty in reproducing their own quantification results. Finally, for most users we find only small variation in the determination of foci size between quantification runs (Fig. 1e).

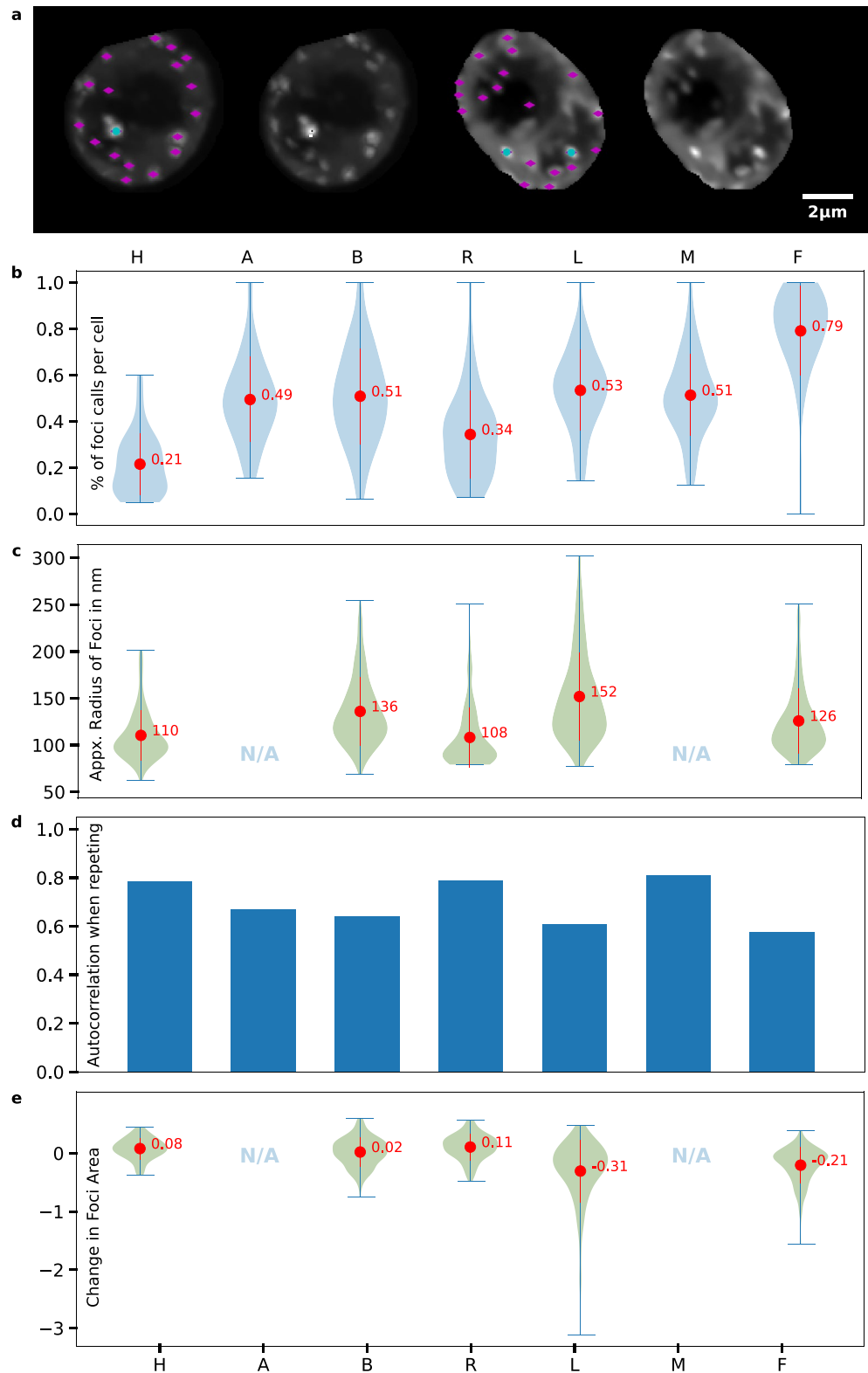
The message from these findings is clear: manual quantification is prone to very large errors especially between different users, particularly when the signal to noise ratio of SG foci to diffuse cytoplasmic background is not especially high (a common issue in SG yeast data analysis). Due to a lack of accessible, automated data analysis tools suited for SG quantification in yeast, we set about developing an automated software solution.

### HARLEY (human augmented recognition of LLPS ensembles in yeast); a novel software for cell segmentation and foci identification via a trained model.

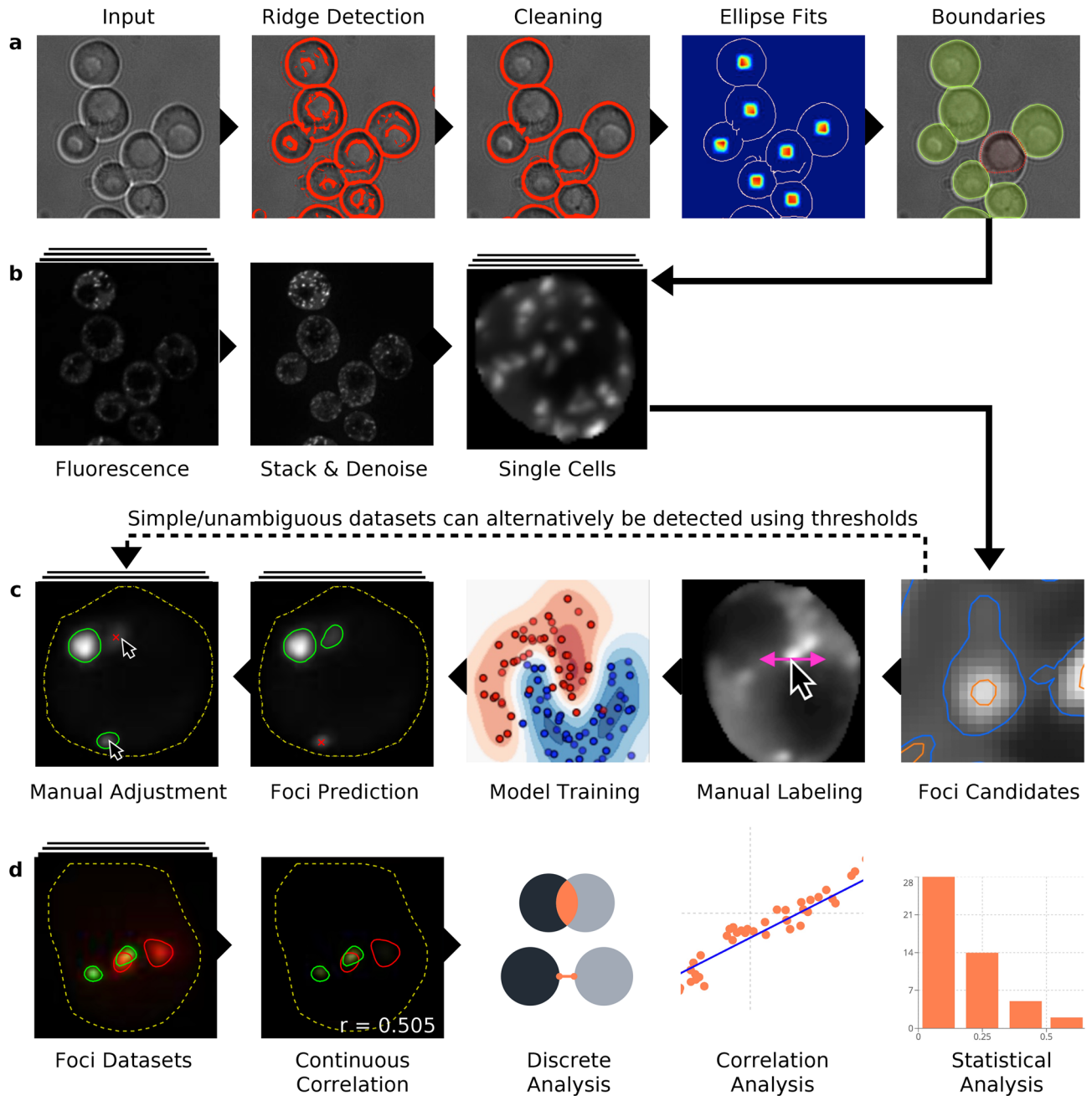
Our software allows researchers to efficiently proceed from image data to final quantification results and consists of a pipeline of steps that are independent and interchangeable with other methods. At important steps of the pipeline a human experimenter may use the UI to adjust results. The full pipeline consists of (1) Extraction of single cell boundaries as masks on bright field images (Supplementary Video S1). (2) Max-intensity stacking and denoising of fluorescence images (preprocessing; Supplementary Video S2). (3) Extraction of contour loops in individual cells as foci candidates (4) Manual labelling of valid foci by a human to train a model (Supplementary Video S3) and (5) Automated classification of foci using the obtained model with the possibility for manual correction (Supplementary Video S4; Fig. 2a–c). Datasets with very clear, bright foci (e.g., P-bodies, Microtubules, Spindle Pole Bodies, Nucleoli) can be inconvenient to train, due to the absence or very low frequency of closed contour loops that are not actually foci (an SVM requires negative examples to learn as well). In this case we recommend use of a different pipeline that uses thresholds on brightness to include/exclude foci without training a model (Supplementary Video S6). All of the above steps, once optimized, can be applied to multiple data files and run in a batch mode format (again with optimal manual correction), which can save users considerable analysis time and effort. Finally, foci data is exportable as an Excel file, which is broken down on a per cell basis to indicate the number of foci in a cell, their overall size and brightness (all exported brightness values are unnormalized and correspond to the original values detected by the microscope). Average foci number per cell, and percentage of cells with foci are also generated.

**Co-localization features of HARLEY.** A second major feature of HARLEY is a colocalization analysis (Supplementary Video S5; Fig. 2d), that allows for classical Pearson Correlation analysis on a per cell and per foci basis, but also for exploring geometrical relations like distances between foci (measured between foci centroids or contour boundaries), and their percentage/area of overlap. Different properties of foci, their overlapping partners or neighbors can be scatter plotted to detect trends and correlation using various built-in regression





**Figure 1.** SG quantification is highly variable amongst users and on repetition. Seven users (H, A, B, R, L, M and F) classified foci. Users that did not specify foci sizes are excluded in panels (c) and (e). (a) Example of two ambiguous cells in dataset, with choices of the most permissive (magenta) and most conservative user (teal). (b) Ratio of selected foci to available foci in individual cells by user. (c) Approximate radius in nm of foci by user. Since foci are not perfectly round, the radius corresponds to a circle of the same area as selected foci. (d) Autocorrelation (i.e., user reproducibility) measured by Mathews Correlation Coefficient (MCC; see “Materials and methods”) between two quantification runs on the same dataset. (e) Ratio of foci area differences in second run relative to the first.

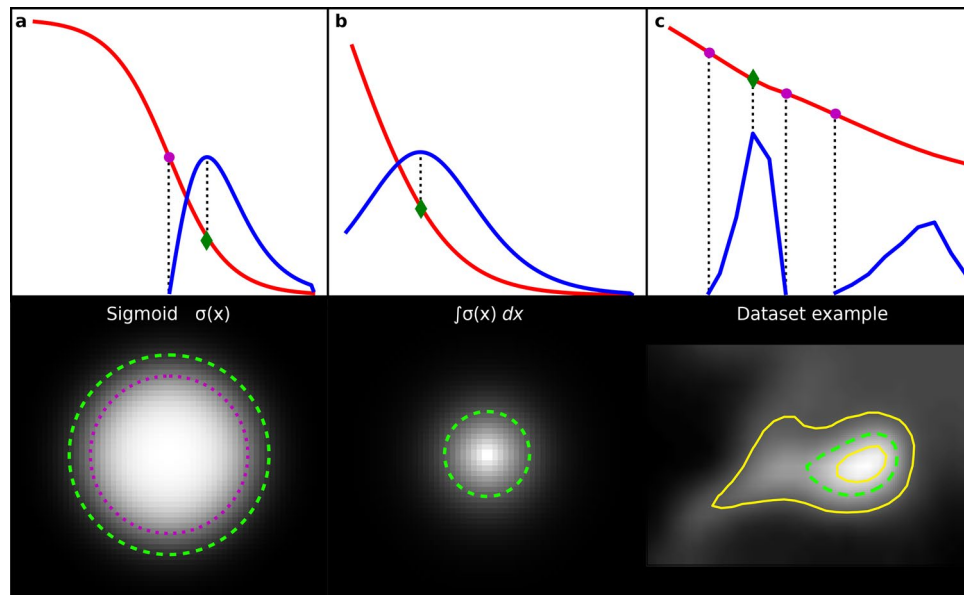


**Figure 2.** Schematic overview of HARLEY. **(a)** Detection of cell outlines using brightfield images. After ridge detection and cleaning to distinguish putative cell edges, cell center calculation (heatmap dots), ellipse fitting and snapping defines cell boundaries (with optional manual rejection of aberrant cell boundaries—e.g., red line). **(b)** Preprocessing and extraction of single cells using the cell outlines and stacked fluorescence images. **(c)** Extraction of foci candidates as contour loops of constrained length, labeling of a training set and automated selection of foci via the trained model. For convenience, a set of thresholds on brightness can be used to detect foci in unambiguous datasets. **(d)** Colocalization and various analyses of foci data.

functions. Data is exportable in various formats including Excel (for tabular data and overview), JSON (for easy import into other scripts) and Python Pickle (raw data of foci geometry and images, for further computational processing).

Below, we describe the underlying mechanistic principles of our algorithm.

**Identifying foci candidates and boundaries.** The key building block of our algorithm are closed contour loops of length within a predefined range, readily found to sub-pixel precision using a marching squares algorithm<sup>23</sup>. If one were to imagine an image in 3 dimensions with the intensity being the height, closed contour lines are lines along which the intensity (or height) does not change and that either circle a peak or a valley, akin to topology lines on maps.



**Figure 3.** Intensities and curvatures (not to scale) of artificial and real foci with resulting images. **(a,b)** Foci modelled by different functions in red and their second derivative in blue (not to scale). The top row represents a slice through the intensity along the radius and has the inflection point (magenta) and the curvature maximum (green) marked. These points correspond to outlines of foci depicted as circles of the same color in bottom row. **(c)** Real life example from our dataset with multiple inflection points. Yellow lines represent inner and outer contours of the foci candidate  $C_k$  and  $C_j$  respectively.

This analogy immediately allows us to identify an important property: contour loops around a point never intersect. They are either non overlapping, equal or the longer one contains the shorter one, rendering it unnecessary to deal with clumped foci explicitly. Our method therefore tends to join clumped features into single larger objects.

We can now define a focus-candidate as a tuple of an outer and a contained inner contour:

$(C_k, C_j)$  with the contour loops  $k \neq j$  having the lengths  $L_{\max}$  and  $L_{\min}$  respectively; the lengths being parameters set by the experimenter (Supplementary Video S3). As we shall see later the algorithm is not very sensitive to these settings, they are simply upper and lower bounds on the features in question.

Each contour line has its respective intensity  $I_k < I_j$  and at this location an intensity  $I_k < i < I_j$  will describe a contour between the outer contour  $C_k$  and the inner contour  $C_j$ .

For nearby foci, an outer contour might contain multiple inner contours, in which case we treat those as multiple separate candidates. Later, when the final outline of foci are calculated, the two candidates will either merge into one or remain two separate foci. The problem of merged or clumped foci is therefore dealt with implicitly and does not require any explicit segmentation, like a watershed approach that is commonly employed in such scenarios<sup>3,10</sup>.

*Identifying the final contour from a focus candidate.* Just as different users would draw contours of a focus differently (Fig. 1c), there cannot be any “correct” method of solving the problem of finding an optimal contour for a candidate. We suggest that more importantly than whether the result looks correct to a given user, contour focus identification should consistently give the same results and not rely on parameters. A good example of a parameterless approach is the Hessian Blob algorithm (Marsh et al., 2018) that uses the zero crossing of the gaussian curvature of an image as a boundary. The Gaussian curvature of the image however requires a scale factor (of the Gaussian), which is a result of the blob detection algorithm but is not present in our approach. Discrete methods of curvature estimates can lack precision given the size of the features that is often only a few pixels across.

We defined our foci candidates as a tuple of two contours  $(C_k, C_j)$  an inner and an outer. The problem of finding final contour  $C_{opt}$  between these two thus can be described as finding an intensity  $I_k < I_{opt} < I_j$  that maximizes some function  $f_{k,j}$  of the two contours:  $I_{opt}(k, j) = \operatorname{argmax}(f_{k,j}(I))$

The concept is visualized in Fig. 3c with inner and outer contours displayed as yellow lines. The choice of the function to optimize is arguably ambiguous. However, in order not to introduce any parameters we examined differential properties. Inflection points and maxima are typical choices in the literature. We suggest using the intensity at which the curvature (i.e., the second derivative  $f = \frac{d^2}{dx^2} I$ ) is at a maximum rather than the inflection point (where it has a zero crossing).

To motivate this choice, we examine different models for an intensity peak by rotating a function of intensity around its y-axis (Fig. 3a,b). We see that it is easy to conceive a focus without any inflection points of intensity



(Fig. 3b). Generally, the curvature maximum outline corresponds more closely to the human perception of foci, while the inflection point boundary appears somewhat too tight.

In the above examples the intensity is a function of radius  $I(r)$ ; what we have though are contour outlines along a fixed intensity that are not circular. In order to approximate a solution, we can imagine a circular focus where the intensity drops linearly with the radius and the area is therefore  $a(I) = \pi r^2$ . We observe further that the area is now in a simple relation to the intensity function  $I(r)$  we are looking for:

$$\sqrt{\frac{a(I)}{\pi}} = I^{-1}(I(r))$$

This means we can obtain the intensity function by inverting the square root of the area function of a contour, which leads to a correct solution in case of a circular focus, where the intensity equals the radius or an approximation otherwise. Evaluation of areas of contours between  $(C_k, C_j)$  at different intensities gives a piecewise linear function that can be inverted (since the area is a monotonous function) and differentiated using numerical methods to derive a final contour for this focus.

Figure 3c shows the result for a focus in our dataset. Note that in this case we have multiple inflection points, but only one absolute maximum, making the resulting contour intensity unambiguous. Furthermore, this example demonstrates the limited influence of parameters  $L_{\max}$  and  $L_{\min}$  for determining the final foci. While they need to be chosen reasonably to include those points of maximum curvature, they have no significant influence on the choice of the final focus outline making the algorithm more stable regarding these parameters. Practically, our software also allows adjustment to the area of the generated foci by a fixed factor, if the automatically determined outline does not correspond to the users understanding.

**Foci classification.** Having found foci as closed contour lines along the maximum curvature we end up with many more candidates, most of which are just perturbations of background signal rather than what human users would classify as *bona fide* foci. The choice of which candidates to count as foci and which to reject strongly depends on the user, as previously shown (Fig. 1b).

Typical criteria a user may intuitively use in identifying foci include a certain % of intensity above the surrounding background signal, foci morphology and co-localization or proximity with other identifiable cellular structures. Some of these criteria are hard to quantify and due to their subjective nature only of limited reproducibility. Defining these rules for all users and cases is therefore not possible. Instead, we extract basic properties of the image and let a classifier algorithm deduce rules based on a labeling provided by a user.

To train an automated classifier we first extract a feature vector  $\underline{x}_i$  from each focus (see “Materials and methods”) yielding a dataset  $D = \{\underline{x}_i | i = 1, \dots, N\}$ . Our software allows users to simply set bounds on these features to generate labels  $l_i \in \{0, 1\}$  where 1 and 0 mean that a closed contour loop is or is not a focus respectively.

More generally such rules are a plane that partitions the feature space:

$$l_i = \begin{cases} 0 & \text{if } \underline{w}^T \underline{x}_i + b > 0 \\ 1 & \text{else} \end{cases}$$

Given a manually labeled dataset we use a Support Vector Machine (SVM) to solve the problem of finding optimal weights  $\underline{w}$  and  $b$  under the side constraint that the plane should be separating the two classes with as large a margin as possible. In simplified terms, a kernel SVM extends this plane of separation to a non-linear surface of separation. In our approach we use radial basis functions as kernels (details on hyper parameter settings in the “Materials and methods”). SVMs are a powerful and flexible class of supervised algorithms that have been successfully applied to image and microscopy data for decades<sup>24–26</sup>. We forego the formal mathematical explanation and implementation details, which are well described elsewhere<sup>27</sup>.

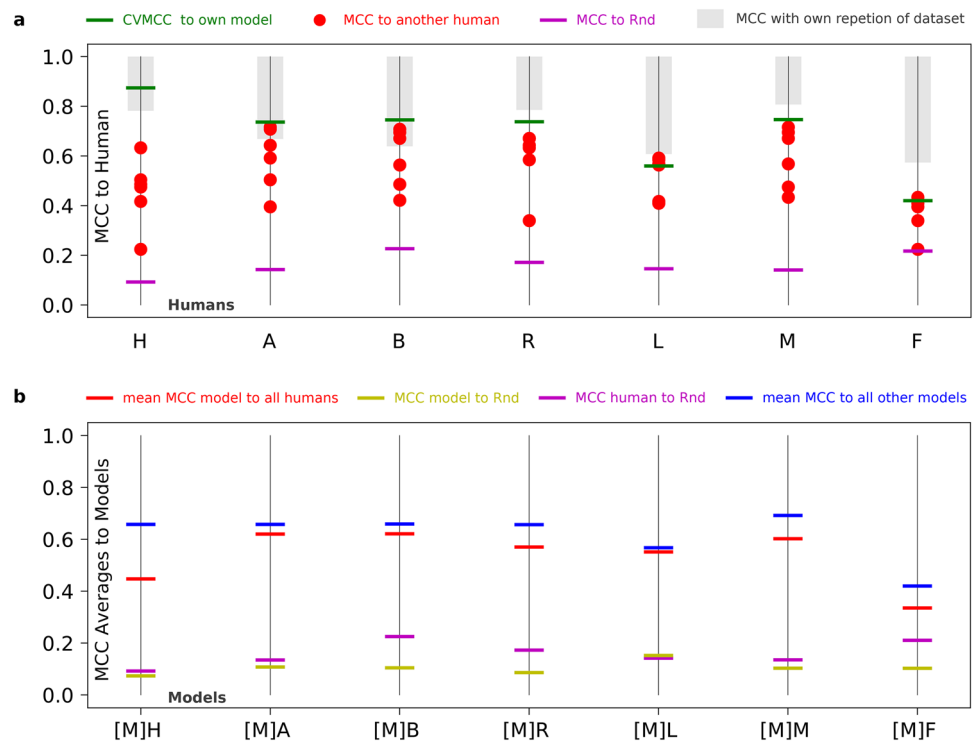
In summary we have (1) devised a method to detect potential foci using two contour loops of minimal and maximal length; (2) proposed a method of finding an optimal contour within these two by approximating the intensity function from the area of contours; and (3) proposed the use of a SVM on various features extracted from these foci to automatically and rapidly label valid foci. Our software deals with all the preparatory steps to this point, like cell outline detection, denoising and stacking of fluorescence microscopy images (Fig. 2, Supplementary videos S1–S4).

### Models match their human behavior closer than other humans while generalizing towards a shared “ground truth”.

While human users usually classify foci from a set of self-chosen rules like “foci are 50% brighter than their surroundings” we have shown that they do not follow these consistently with the best correlations on reclassification of the same dataset being only around 80% (Fig. 1d). The result of user classification could therefore be thought of as a deterministic predictor  $pr$  plus some added random “noise”  $\xi$ . The term “noise” used here refers to an unexplained variability within the labeling, which is a function of human user variability given the dataset.

$$l_i = pr_h(\underline{x}_i) + \xi_h(\underline{x}_i|D)$$

Intuitively a classifier capable of generalizing the data should primarily capture the deterministic term, leaving the noise term indicative of an upper bound for the model performance. Captured in the deterministic term is a user’s familiarity with what a SG should look like and that which a model should capture primarily. While our sample size is too small to undertake a thorough statistical examination of the noise and assess performance boundaries for the model, our results are indicative that our model indeed removes some of this noise.



**Figure 4.** Evaluation of model performance. **(a)** Correlation of users to the trained model (green), other users (red) and the random classifier (magenta). The gray bar goes from 1 to the correlation of the user with themselves (see Fig. 1d). **(b)** Mean correlations of models to all other models (blue), all users (red) along with the correlations of this model with the random classifier (yellow) and for comparison the user's correlation from panel A (magenta).

For most users, the correlation to their model lies above the correlation to other users (Fig. 4a), which means that the model performs a quantification that closer matches that user versus another.

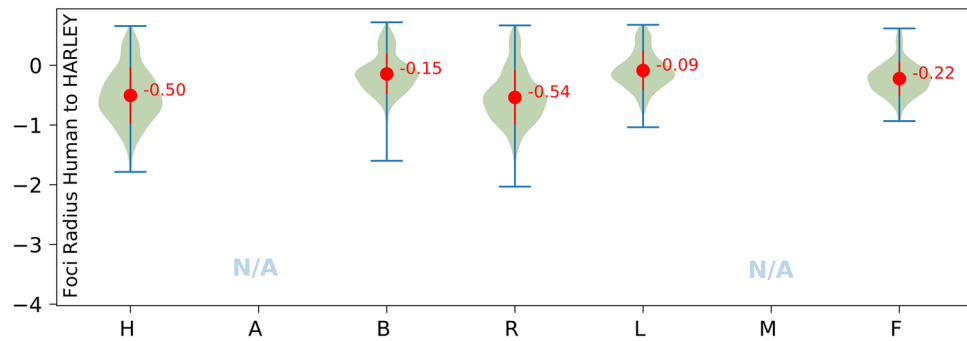
To assess the relation to noise, we compared the users and models to a random classifier (“Rnd”, see “Materials and methods”). Figure 4b compares models trained on their respective user to other users and other models. Notably, models generally correlate closer together than to the other users (blue bars above red), regardless of the user they have been trained on. This suggests that models tend to generalize the deterministic term of the classification rather than learn the noise. In theory if the understanding of a SGs was identically defined across all users (like what a “Dog” or “Cat” is), then users would only differ in the noise term and all models would tend to generalize towards the same result. Furthermore, we observe that the correlation to the random classifier is lower for all models compared to the users (yellow bars below magenta). This demonstrates that model training reduces the noise in the classification.

**Prediction of foci sizes by our algorithm sizes is often close to some users but is of limited use depending on image resolution.** Our goal for determining an outline for foci was to not use any parameters and we devised an approximative differential approach (see earlier, Fig. 3). While some users have a preference close to the results of HARLEY, other deviate by up to 54% (Fig. 5).

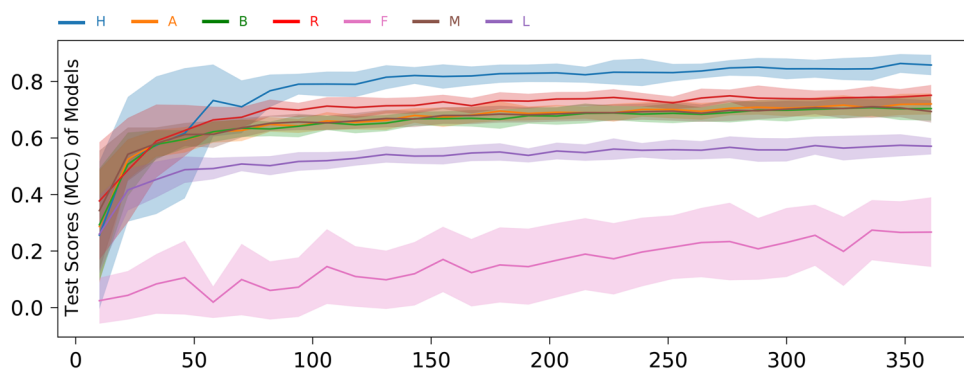
SGs in our dataset are as little as 4-5px in radius. A deviation of 1px (or 40.7 nm) therefore already constitutes a 25% deviation. (For data input the users worked on an upscaled image of the cell allowing for more control, but blurry images).

Under these considerations, the results of the size prediction are fairly close to the users with a tendency to overestimate the area. A further improvement is achieved by adjusting a parameter to increase or decrease all foci sizes by a parameter in the UI of our software (Supplementary Video S4), which in the best case corresponds to shifting the results in Fig. 5 upwards by their mean. Other solutions like a separate model for size prediction are feasible, however size of foci in absolute terms is often not informative by itself. Instead, detecting relative changes in foci size is more often informative, for which the problem of overestimating the size does not really matter.

**A stress granule training set  $\geq 15$  cells achieves effective user SVM modelling.** To assess how quickly HARLEY can learn the SG-identifying behavior of a given user, we randomly took an increasing subset of data, trained the model and evaluated its performance on the remainder of the data. In contrast to cross validation, the subsets are chosen randomly, and we generated 30 samples for each subset size. The subset size is measured in number of foci candidates with the average cell containing 13.2 candidates.



**Figure 5.** Foci radii determined by HARLEY typically exceed user-defined radii. 0 corresponds to equal sizes. Negative values correspond to an overestimation on the side of the model. Since foci are not round, radii correspond to circles of same area as selected foci.



**Figure 6.** Model performance by size of dataset. Model hyper parameters are not optimized. Shaded areas show one standard deviation above and beyond the mean determined from 30 repeats for each setting.

While the performance of the model keeps increasing, a training set of 100–200 foci candidates (on average less than 15 cells in our standardized  $\text{NaN}_3$  dataset) is sufficient for the model to learn most users (Fig. 6). In practice, labelling 15 cells takes not much more than 5–10 min with our software. Training times for the SVM itself for such small datasets are on the magnitude of milliseconds to seconds (for determining optimal hyper parameters), making this approach highly practical for users with relatively consistent scoring of foci.

User F seems to be an outlier with the model learning his behavior only very slowly. We suspect that this user has a very high level of inconsistency in their behavior. The highest correlation with a random classifier (Fig. 4), lowest reproducibility of their own results (Fig. 1) and the large variance in Fig. 6 support this. Furthermore, this user exhibited a high propensity of choosing all available foci candidates.

## Discussion

We have shown that user variability is a significant problem in scoring of somewhat ambiguous cellular foci such as SGs. In a group of just 7 users, variability approached fourfold, with user reproducibility of foci identification only reaching a maximum of about 80%. This casts significant doubt on interpreting marginal differences in SG microscopy data generated by different labs, users and even within studies where only a single user has been responsible for quantification. To counteract such variability, and aid meaningful data comparisons, we developed HARLEY to facilitate rapid and consistent identification of yeast cell boundaries and foci using a novel contour-loop based approach. Our approach is much more invariant to foci shape (i.e., works equally well for round or elongated shapes) and deals with clumped foci in an elegant implicit fashion. We have shown that our data training approach can rapidly generate a model that will allow users to consistently and rapidly quantify foci of interest, including via a batch mode. The ability of users to reproduce their own results is indicative of model performance with the best results coming from users who can reproduce their own results with ~80% correlation, which is not surprising, since human quantification inconsistencies accumulate in the training dataset. This greatly improves upon prior data analysis strategies for yeast SGs and PBs that we have previously described<sup>28</sup>.

Many popular software tools currently used for quantification of foci in microscopy data enable practical parameter-constrained solutions, that are very sensitive to a technical understanding of the interplay and influence of these parameters<sup>14</sup> or produce poor results on noisy data. A model training-based approach solves these issues by asking the user to define the desired outcome rather than parameters that lead to it. Biologists are often familiar with software like ImageJ, that while having a diverse array of uses, may become limited for high-throughput analysis of microscopy data, owing to a somewhat inflexible structure and the need for manual input.

A chief advantage of HARLEY is a clear, UI guided process that allows users with minimal technical expertise to quickly create consistent and reliable quantification results. The ability to share trained models allows reproduction and comparison of results reported by other labs, which we hope will increase the quality of reported data. While we are aware that deep learning approaches may generate better results, no deep learning approach can be trained in milliseconds using only a small number of cells.

HARLEY's design was focused on foci detection, quantification and co-localization analysis in *S.cerevisiae* (yeast). In particular, HARLEY excels at reproducibly and efficiently distinguishing foci like stress granules, whose often relatively low signal to noise ratio, and frequent resistance to classification based on simple thresholding methods, due to cell to cell variability, makes their classification cumbersome using other existing software tools (e.g., Fiji, Cellprofiler). While such tools can quantify such foci (and perform many other functions), in our experience HARLEY is more efficient at this particular task, and more intuitive to non-expert users who are not required to gain detailed understanding of Fiji or Cellprofiler features, and optimal ways to combine them. Thus, we anticipate that HARLEY will benefit the community and encourage more biologists to use automated software for their quantification. In supplementary Table S1, we further highlight advantages and reasons a user may choose to use HARLEY.

To improve our approach for foci detection the feature space likely needs to be extended. A straightforward solution would be to use a feature extraction CNN to generate the feature vectors for the SVM that are most relevant to the user or generate extensive datasets using a variety of filters. While a SVM only needs to learn the equation for a separating plane in the dimension of the feature space (expressed as linear weights on data-points, with most being irrelevant, i.e., not support vectors), CNNs require learning of weights many orders of magnitudes above that (with values of  $10^5$ – $10^6$  being not uncommon), which increases demands on time and computational power significantly. While a gain on accuracy is possible, the quality of training data may be more problematic due to strong variability in human labeling. A reasonable trade off on accuracy and practical usability must be found with the application of these approaches. A potential gain in consistency might be achieved by using quantifications from multiple human labelings combined into one model, to assess a common denominator of what humans deem foci.

More work should be conducted to assess how well a trained model solves the task of foci detection. However, it should be stressed that comparing a model to a user baseline is somewhat limited, due to inconsistencies across human users. As in our case, some humans can be “learned” very well (H), while others are very hard to learn (F). The user “baseline” is therefore a spectrum which is not surprising, since there is no clear definition of what foci are. One solution is the use of standardized datasets and performance measures that the scientific community generally lacks. We are aiming at developing a more extensive dataset for SGs and P-bodies, as well as other cellular structures amenable to contour-based identification, in the future. The presented dataset (and others) is freely available already at <https://github.com/lnilya/harley>.

While our model readily learns some users the performance for others is poor, which is likely due to the noisiness of their classification. If a user can reproduce their own result only to a 60–80% correlation, arguably any generalizing model will not learn to more than this precision. While reproducibility and model performance show a correlation (green bars and gray area in Fig. 4a) more data is needed to estimate the variability distribution for a user and dataset and derive an upper bound for model performance, to thoroughly test this hypothesis. The results so far indicate that users should assess their own performance to determine the quality of their quantification reproducibility, to estimate how well their trained model is expected to perform.

As previously noted, HARLEY can be readily applied to quantification of other cell types as well as other organelles/foci that are approximately convex in nature and do not overlap extensively. Regarding cell types, HARLEY can be used “as is” on other cell type, so long as the cells are elliptical in shape and their brightfield images resemble those in our demo images datasets (link available at <https://github.com/lnilya/harley>); specifically, cell edges should be detectable by the Frangi edge filter<sup>19</sup>. However, it is possible to detect cells with other software (or manually), and exporting the result as a binary mask image, which will allow a user to bypass the cell detection pipeline. As for foci detection, this is readily possible so long as the foci are quite distinct and are not excessively merged. Indeed, besides stress granules and P-bodies, we have readily quantified other types of foci in yeast, including eIF2B bodies, microtubules, nucleoli and spindle pole bodies (example files of all present in demo dataset at link above). In cases of high signal/low noise foci, and objects present in all cells, our threshold pipeline, which bypasses SVM training, should be used (Supplementary Videos S4). Future goals for HARLEY are to develop new types of cell and foci detection (e.g., for “clumped” shapes such as mitochondria or vacuoles) that we can integrate into the existing pipeline framework. In this way, we hope to maximize HARLEY usability amongst users as an intuitive platform for biologists working with yeast microscopy data.

Finally, since HARLEY is based on a problem agnostic pipeline-based framework that we are developing separately, interested researchers are strongly encouraged to reach out and participate in the development of a new platform for delivering simple, intuitive UIs to any algorithms that would be too complex or unwieldy as a plugin to existing software. The nascent platform is named SAMMIE (Scientific Algorithms Modelling and Mixing Interface Engine) and is also accessible via Github (<https://github.com/lnilya/sammie>). Since HARLEY is based on ReactJS, the usability, ease of development and cross platform compatibility are among its greatest strengths. We encourage the community to build or retrofit their algorithms into beautiful easy to understand tools using this framework, thus benefiting the scientific community.

### Data availability

HARLEY is freely available on our Github repository(<https://github.com/lnilya/harley>). Standardized demonstration datasets of stress granules, P-bodies and other types of cellular foci (see discussion) are also available here. The raw unscaled images of stress granules in individual yeast cells, which were used in our human foci calling variability study, along with anonymized data responses of users, and the brightness distribution of this

dataset, is publicly available at [https://drive.google.com/file/d/12okQ6\\_vgtCLkNKki2zUJY3wqqV6psQDG/view?usp=sharing](https://drive.google.com/file/d/12okQ6_vgtCLkNKki2zUJY3wqqV6psQDG/view?usp=sharing) Extensive statistical processing and coding was conducted to generate the figures. This code, while requiring extensive coding expertise, is available upon request, as is the software used to gather the raw human variability in foci calling data.

Received: 27 January 2022; Accepted: 8 July 2022

Published online: 18 July 2022

## References

- Lapytsko, A., Kollarovic, G., Ivanova, L., Studencka, M. & Schaber, J. FoCo: a simple and robust quantification algorithm of nuclear foci. *BMC Bioinformatics* **16**, 392 (2015).
- Jucha, A. *et al.* FociCounter: a freely available PC programme for quantitative and qualitative analysis of gamma-H2AX foci. *Mutat. Res.* **696**, 16–20 (2010).
- McQuin, C. *et al.* CellProfiler 3.0: Next-generation image processing for biology. *PLoS Biol.* **16**, e2005970 (2018).
- Mommel, S. *et al.* FocAn: Automated 3D analysis of DNA repair foci in image stacks acquired by confocal fluorescence microscopy. *BMC Bioinformatics* **21**, 27 (2020).
- Ledesma-Fernández, E. & Thorpe, P. H. Fluorescent foci quantitation for high-throughput analysis. *J. Biol. Methods* **2**, e22 (2015).
- Perez-Pepe, M. *et al.* BUHO: A MATLAB script for the study of stress granules and processing bodies by high-throughput image analysis. *PLoS ONE* **7**, e51495 (2012).
- Lindeberg, T. Scale selection properties of generalized scale-space interest point detectors. *J. Math. Imaging Vis.* **46**, 177–210 (2013).
- Marsh, B. P., Chada, N., Sanganna Gari, R. R., Sigdel, K. P. & King, G. M. The Hessian Blob Algorithm: Precise particle detection in atomic force microscopy imagery. *Sci. Rep.* **8**, 978 (2018).
- Vicar, T. *et al.* DeepFoci: Deep learning-based algorithm for fast automatic analysis of DNA double-strand break ionizing radiation-induced foci. *Comput. Struct. Biotechnol. J.* **19**, 6465–6480 (2021).
- Hohmann, T., Kessler, J., Vordermark, D. & Dehghani, F. Evaluation of machine learning models for automatic detection of DNA double strand breaks after irradiation using a  $\gamma$ H2AX foci assay. *PLoS ONE* **15**, e0229620 (2020).
- Herbert, A. D., Carr, A. M., Hoffmann, E. & Lichten, M. FindFoci: a focus detection algorithm with automated parameter training that closely matches human assignments, reduces human inconsistencies and increases speed of analysis. *PLoS ONE* **9**, e114749 (2014).
- De Graeve, F. *et al.* Detecting and quantifying stress granules in tissues of multicellular organisms with the Obj.MPP analysis tool. *Traffic* **20**, 697–711 (2019).
- Lengert, N. *et al.* AutoFoci, an automated high-throughput foci detection approach for analyzing low-dose DNA double-strand break repair. *Sci. Rep.* **8**, 17282 (2018).
- Schneider, J. *et al.* Open source bioimage informatics tools for the analysis of DNA damage and associated biomarkers. *J. Lab. Precis. Med.* **4**, 21–21 (2019).
- Chicco, D. & Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* **21**, 6 (2020).
- Runge, R. *et al.* Fully automated interpretation of ionizing radiation-induced  $\gamma$ H2AX foci by the novel pattern recognition system AKLIDES\*. *Int. J. Radiat. Biol.* **88**, 439–447 (2012).
- Buchan, J. R., Yoon, J.-H. & Parker, R. Stress-specific composition, assembly and kinetics of stress granules in *Saccharomyces cerevisiae*. *J. Cell Sci.* **124**, 228–239 (2011).
- Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta* **405**, 442–451 (1975).
- Frangi, A. F., Niessen, W. J., Vincken, K. L. & Viergever, M. A. Multiscale vessel enhancement filtering. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **1496**, 130–137 (1998).
- Zhang, T. Y. & Suen, C. Y. A fast parallel algorithm for thinning digital patterns. *Commun. ACM* **27**, 236–239 (1984).
- Kass, M., Witkin, A. & Terzopoulos, D. Snakes: Active contour models. *Int. J. Comput. Vis.* **1**, 321–331 (1988).
- Buades, A., Coll, B. & Morel, J.-M. Non-local means denoising. *Image Process. Line* **1**, 208–212 (2011).
- Lorensen, W. E. & Cline, H. E. Marching Cubes: A High resolution 3D surface construction algorithm. *Proc. 14th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '87* **21**, (1987).
- Miao, Q., Derbas, J., Eid, A., Subramanian, H. & Backman, V. Automated cell selection using support vector machine for application to spectral nanocytology. *Biomed Res. Int.* **2016**, 6090912 (2016).
- Maglogiannis, I. G. & Zafiropoulos, E. P. Characterization of digital medical images utilizing support vector machines. *BMC Med. Inform. Decis. Mak.* **4**, 4 (2004).
- Wang, M. F. Z. & Fernandez-Gonzalez, R. (Machine-)Learning to analyze in vivo microscopy: Support vector machines. *Biochim. Biophys. Acta. Proteins Proteom.* **1865**, 1719–1727 (2017).
- Chang, C. C. & Lin, C. J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**, 27 (2011).
- Buchan, J. R., Nissan, T. & Parker, R. Analyzing P-bodies and stress granules in *Saccharomyces cerevisiae*. *Methods Enzymol.* **470**, 619–640 (2010).

## Acknowledgements

The authors wish to thank the anonymous volunteers who scored stress granule yeast data, and members of the Buchan lab who facilitated software debugging, in particular Mayra Karina Rivera and Lucas Harrell. J.R.B. acknowledges support from National Institute of General Medical Sciences (NIH) Grant R01-GM114564 and kind donations from Keith and Mara Aspinall.

## Author contributions

I.S. performed all coding and data analysis. J.R.B. recruited users and generated demonstration microscopy data. I.S. and J.R.B. designed the project and wrote the paper.

## Funding

This article was funded by National Institute of General Medical Sciences (R01-GM114564).

## Competing interests

The authors declare no competing interests.



### Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-022-16381-2>.

**Correspondence** and requests for materials should be addressed to I.S. or J.R.B.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022