Image processing outline for the publication

“Cell size regulation in budding yeast does not depend on linear accumulation of Whi5”

11/19/2019

Guide written by: Felix Barber (felix1635@gmail.com)

Outline

Extracting data on individual cell cycles from time-lapse microscopy images is a multi-step process. The first step in this pipeline involves image segmentation to distinguish individual cells within each image. Cells are then subsequently tracked over successive timepoints, and these individual tracks are combined with fluorescence data to infer individual cell cycles and track lineages. Examples of various points of this analysis pipeline are presented in Figure 1. We performed brightfield image segmentation and cell tracking using the opensource CellSTAR algorithm (1). Cell volume was inferred based on this brightfield segmentation by fitting an ellipse to the 2D mask. Cell volume was then approximated as that of a prolate spheroid with that given 2D elliptical cross section. We designed a custom, semi-automated image processing pipeline to incorporate fluorescence data and compile measurements on individual cell cycles. Cell cycle progression was assessed based on Whi5 nuclear localization, in accordance with previous approaches (2). Whi5 nuclear localization was detected using an SVM machine learning classifier from the *Scikitlearn* python package, trained on manually annotated data for each individual time-lapse. Since we obtained z-stack time-lapse images, fluorescence signal from the constitutively expressed mCherry protein was used to create 3D masks of each cell. This mask was constructed by thresholding, using a threshold of 3 standard deviations above the average background fluorescence level. As such, these masks did not include vacuoles, since fluorescence was excluded from these organelles. These 3D masks were used to infer average cellular fluorescence as the average of pixel intensities within each mask. All relevant code is available at https://github.com/felixbarber/image_processing_cellstar.

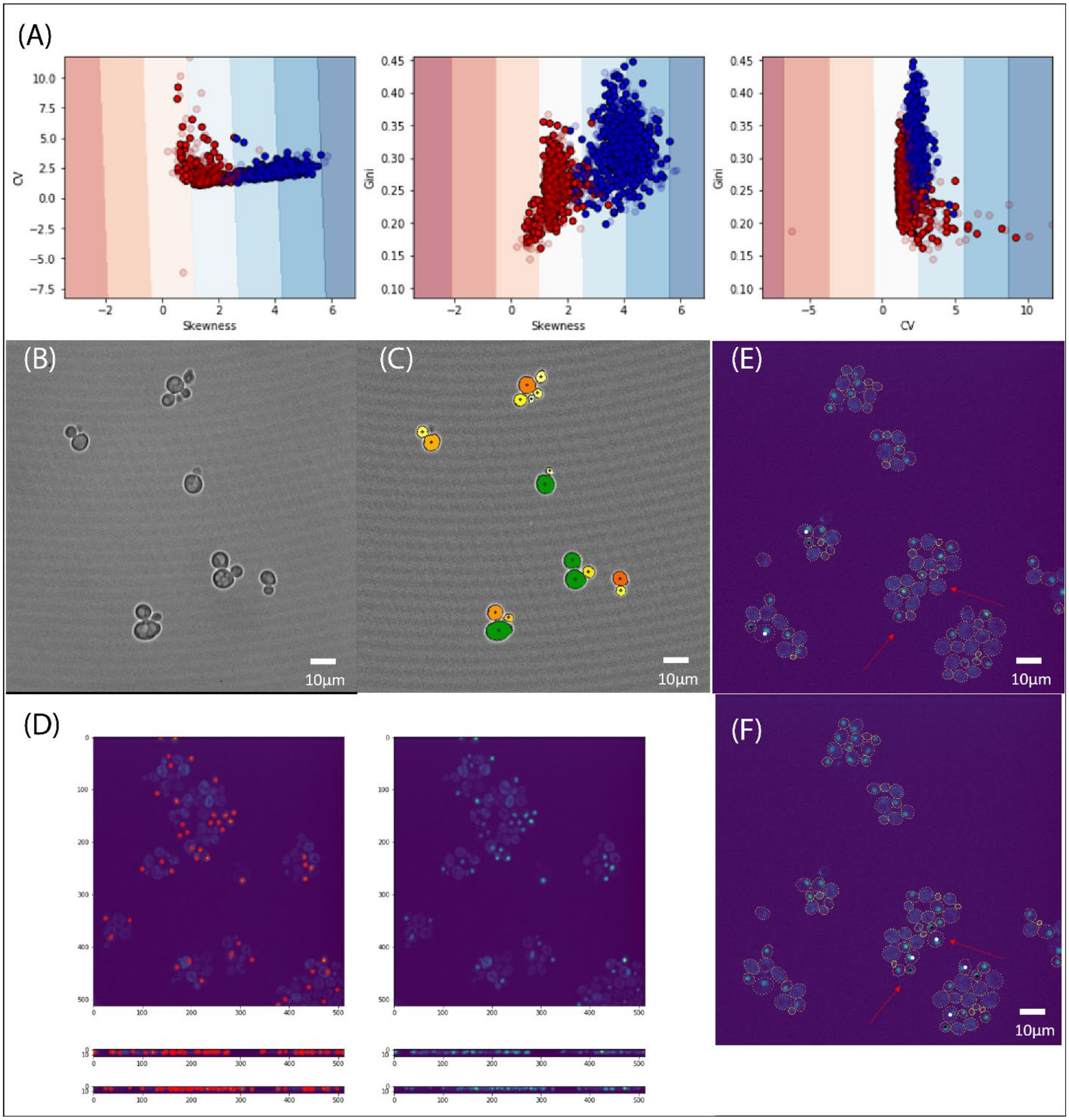

Figure 1: Examples of various stages in our image processing pipeline. (A) Plots of the classification of Whi5 localization based on the CV, the Gini coefficient, and skewness measured for the distribution of pixel fluorescence brightness based on a 3D segmentation of cell vs non-cell. (B) Sample brightfield image for segmentation. (C) Example of segmentation performed by the CellSTAR algorithm on the image in (B). (D) 3D spot detection using the *skimage regionprops* function. (E-F) Cell lineage tracking based on Whi5 localization for two subsequent timesteps. Red arrows highlight cells with Whi5 nuclear localization in the subsequent timestep, causing the assignment of mother-daughter cell pairs for adjacent cells. White dots correspond to mother cells while black dots correspond to daughter cells.

Brightfield segmentation and tracking using the CellSTAR Algorithm

Our image segmentation was done based on brightfield data acquired using a spinning disc confocal microscope. The inference of individual cells based on brightfield data is a challenging computational problem, made particularly difficult in the context of budding yeast due to the abundance of out-of-plane budding events which take place within an expanding colony. To approach this problem we employed the CellSTAR Algorithm via an open-source MATLAB plugin (1). This plugin requires MATLAB Release 2014b. We strongly recommend gaining familiarity with this software by thoroughly reading the user manual available online prior to analyzing microscopy data. However, certain functions are available which are not discussed in this manual, and to address this we recommend using the help function within the software by pressing the “h” key. This provides a pop-up menu with a brief description of useful functions.

In order to use the CellSTAR algorithm on individual fields of view (FOVs, or scenes) in a time-lapse, we have designed a python script “*make_image_dirs.py”* to generate the necessary file structure and populate these files with the corresponding brightfield images. Navigating the current directory to within the folder for each scene will then ensure that the corresponding track images are generated in the correct manner. The relevant file path in this script should be adapted before running.

Key features which we observed to be important for optimal performance of this algorithm using the default settings were acquiring brightfield images with the focal plane adjusted to create a dark and white band around individual cells. Additionally, we found it to be beneficial to use a background image in each FOV by taking the first image in the time-lapse series associated with that FOV and using the inbuilt functionality to apply a blurring algorithm, eliminating cells from each field. If images are of high quality, with a sufficiently high brightness to reduce noise but low enough to avoid thresholding by the camera CCD, cell segmentation should be performed automatically with high accuracy. However, we strongly recommend manually correcting individual segmentation errors, and advise the use of podcasts or audiobooks during this rather tedious task. We note that seeding of additional cells within the CellSTAR algorithm is made quite easy through the action of a single mouse-click, however, the accuracy with which this seeding tracks the associated cell shows some consistent dependence on where in the cell a seed is initiated. We recommend gaining familiarity with this by seeding the same cell repeatedly at first.

Once cell segments are generated, cell tracking can be performed. This again can be performed automatically using the Cellstar algorithm, and is generally quite effective in the early stages of colony growth. Once colonies grow to be sufficiently large, it becomes both computationally intensive and challenging for the software to perform tracking effectively, and we strongly recommend manually inspecting individual tracks as a quality control. This process is tedious, but crucial, since a single break in the trace of a cell’s full cell cycle will eliminate that cell cycle from consideration in later stages of our data analysis pipeline, causing a loss of potentially useful data.

Cell cycle tracking and fluorescence integration with the custom imaging toolkit

Once tracking is completed to a satisfactory level, the python script "*assemble_from_segments.py*" can be run with appropriate parameters for each time-lapse to attach the appropriate fluorescence traces to each time-lapse. Within this study we consistently used two fluorescence traces: a Whi5-mVenNB label to track cell cycles, and a constitutively expressed mCherry fluorophore (known as c2 for channel 2). Use of the mCherry fluor is not strictly necessary but is recommended if the user desires to use this algorithm with the minimum of effort, since not using this may require some re-working of the appropriate subroutines. In brief, our pipeline initially associates each cell trace with a "Cell" object instance, first populating each Cell with data that can be inferred based on brightfield images, before populating fields corresponding to each of the fluorescence channels mentioned above. Our algorithm then proceeds to use an SVM machine learning classifier to detect Whi5 nuclear localization events within each Cell. This requires additional user input by running the python script "*track_localization_final.py"* to randomly select specific frames from the full time-lapse from which to generate ground truth data. Within this script, individual cell clicks are recorded within tracked cells (denoted by yellow boundaries) to detect localization events. The help key "h" provides a guide to the functions available within this custom-designed interface. Once a user is familiar with this program, the generation of ground truth data for a single time-lapse typically takes roughly 15-20 minutes. We note that this program will generate errors if no localization events are recorded in any ground truth frames, and we have accordingly generated a function to re-draw individual frames when no localization events are present within tracked cells.

Once the training data has been generated, one may re-run the "*assemble_from_segments.py*" python script to detect Whi5 localization events and extract data on individual cell cycles. This data is now stored as individual "Cellcycle" object instances, with references to parent cells and daughter cells providing a means of inferring lineages. Each object instance will now have an associated bud during the S/G2/M phases of the cell cycle, which may be combined with data from the main cell to make statements about the full cell. Finally, the "*export_cycles_pandas.py*" python script takes these object instances and generates a pandas dataframe with data corresponding to each cell at birth, Start and division. These dataframes can be combined for different time-lapse measurements, provided experiment IDs are appropriately recorded, to rapidly and easily compare and contrast data between different time-lapse experiments.

Many of these scripts will involve calls to functions contained in the "*custom_image_toolkit.py*" file. Within this file, a number of different versions of many functions are contained. If editing of these functions is required, we recommend first ensuring that the function in question is edited.

One difficulty we encountered in inferring cell lineages based on Whi5 localization was that Whi5 nuclear entry can happen for multiple mother-daughter pairs in close proximity simultaneously. For a single isolated mother-daughter pair our algorithm performs with acceptable accuracy, however, at present it rejects from consideration any situations in which multiple pairs of cells show nuclear entrance simultaneously to avoid the corresponding ambiguity and error in assigning cell lineages. One function which has been in development but has not been fully implemented involves manual correction when such situations arise using the "*assign_troublesome_pair*" python function in the "*custom_image_toolkit.py*" script. This function may improve the extent to which useful data can be extracted from our time-lapse data, but will require further optimization before being fully operational.

References

1. C. Versari, *et al.*, Long-term tracking of budding yeast cells in brightfield microscopy: CellStar and the Evaluation Platform. *Journal of the Royal Society Interface* **14** (2017).

2. S. D. Talia, J. M. Skotheim, J. M. Bean, E. D. Siggia, F. R. Cross, The effects of molecular noise and size control on variability in the budding yeast cell cycle. *Nature* **448**, 947–951 (2007).