High-Throughput Computational Analysis of Biofilm Formation from Time-Lapse Microscopy

Ola E. Salama[1,3] and Aleeza C. Gerstein[1,2]

[1]Department of Microbiology, University of Manitoba, Winnipeg, Manitoba, Canada
[2]Department of Statistics, University of Manitoba, Winnipeg, Manitoba, Canada
[3]Corresponding author: *ola6997@gmail.com*

Candida albicans biofilm formation in the presence of drugs can be examined through time-lapse microscopy. In many cases, the images are used qualitatively, which limits their utility for hypothesis testing. We employed a machine-learning algorithm implemented in the Orbit Image Analysis program to detect the percent area covered by cells from each image. This is combined with custom R scripts to determine the growth rate, growth asymptote, and time to reach the asymptote as quantitative proxies for biofilm formation. We describe step-by-step protocols that go from sample preparation for time-lapse microscopy through image analysis parameterization and visualization of the model fit. © 2021 Wiley Periodicals LLC.

Basic Protocol 1: Sample preparation
Basic Protocol 2: Time-lapse microscopy: Evos protocol
Basic Protocol 3: Batch file renaming
Basic Protocol 4: Machine learning analysis of Evos images with Orbit
Basic Protocol 5: Parametrization of Orbit output in R
Basic Protocol 6: Visualization of logistic fits in R

Keywords: Candida • drug response • image analysis • orbit • R

How to cite this article:
Salama, O. E., & Gerstein, A. C. (2021). High-throughput computational analysis of biofilm formation from time-lapse microscopy. *Current Protocols*, *1*, e194.
doi: 10.1002/cpz1.194

INTRODUCTION

Biofilm formation is an important virulence factor in *Candida albicans* (Cavalheiro & Teixeira, 2018; Chandra et al., 2001; Douglas, 2003; Fanning & Mitchell, 2012; Mayer, Wilson, & Hube, 2013). There are three distinct developmental stages of biofilm formation: (1) adherence of the yeast cells, (2) an intermediate phase of dimorphic switching from yeast to hyphae and formation of the matrix, and (3) the formation of the three-dimensional mature biofilm architecture (Cavalheiro & Teixeira, 2018; Kojic & Darouiche, 2004). *C. albicans* biofilms are the leading cause of medical device−acquired nosocomial infections because they have the ability to adhere to and grow on different types of medical devices, such as prostheses, cardioverter defibrillators, catheters, and cardiac devices (Douglas, 2003; Elving, van der Mei, Busscher, van Weissenbruch, & Albers, 2002; Kojic & Darouiche, 2004; Potera, 1999). *C. albicans* biofilms have been reported to be highly resistant to a variety of antifungal drugs including fluconazole, itraconazole, amphotericin B, nystatin, and chlorhexidine (Chandra et al., 2001; Fanning

& Mitchell, 2012; Hawser & Douglas, 1995; Kuhn, Chandra, Mukherjee, & Ghannoum, 2002; Sherry et al., 2017). There are well-developed *in vitro* biofilm protocols to study antifungal resistance by examining drug penetration in mature biofilms (Gulati et al., 2018; Pierce, Uppuluri, Tummala, & Lopez-Ribot, 2010). These assays measure the antifungal drug response by evaluating the biofilm biomass by optical density (Fox et al., 2015; Lohse et al., 2017), metabolic activity (Kuhn et al., 2002; Melo, Bizerra, Freymüller, Arthington-Skaggs, & Colombo, 2011; Nett, Cain, Crawford, & Andes, 2011), or direct observation through confocal scanning microscopy (Chandra et al., 2001; Fox et al., 2015). However, fewer protocols exist to examine the earlier developmental stages, i.e., to determine whether drugs inhibit adherence and/or the intermediate phases of biofilm formation (Abdelmegeed & Shaaban, 2013; McCall, Pathirana, Prabhakar, Cullen, & Edgerton, 2019; Nagy et al., 2014).

Time-lapse microscopy is the technique to capture a sequence of images at regular time intervals (Collins, van Knippenberg, Ding, & Kofman, 2019). Time-lapse microscopy has previously been used to examine the growth rate of biofilms qualitatively, by visually evaluating the effect of the drug (Kawai, Yamagishi, & Mikamo, 2017), or quantitatively, by measuring the thickness of biofilms grown in the presence of drug (Kaneko et al., 2013). However, time-lapse images are often only examined qualitatively. When the large number of images created by a typical time-lapse are quantified manually for parameters that correlate with biofilm formation ability, it can be very time consuming and subjective. Although numerous analysis tools are available for image analysis, such as CellProfiler (Carpenter et al., 2006), ImageJ (Lind, 2012), and NeuronStudio (Rodriguez, Ehlenberger, Dickstein, Hof, & Wearne, 2008), these tools are designed for specific and typically singular purposes, such as cell counting, detection of specific cell types (e.g., human cells), and/or work with a specific type of microscopic image (e.g., fluorescent images).

The authors of this article have developed a method that quantifies biofilm formation from the earliest phases, i.e., before adherence, in a high-throughput, quantitative, and unbiased fashion. To accomplish this, we combined 96-well time-lapse microscopy of bright-field images with Orbit image analysis. Orbit is an open-source software application (available at *https://www.orbit.bio/*) which incorporates a machine-learning algorithm that can be trained to detect and quantify cells, and provides a built-in mechanism to analyze a large number of images (Stritt, Stalder, & Vezzali, 2020). Orbit does not require the user to have extensive prior knowledge of image analysis or scripting, and does not depend on having a huge training set of images. We use Orbit to quantify the area of each well that is covered by cells at each time point. We then developed custom scripts in the R programming language (R Core Team, 2020) to fit a logistic equation to % area over time, to determine the growth rate (slope during exponential phase), % area at the growth asymptote, and the time it takes to reach the asymptote. Finally, we provide a script to visualize the model fit. We describe how we have used this protocol to examine biofilm formation of *C. albicans* in the presence of the broad-spectrum antimicrobial boric acid; however, note that biofilm formation by any species in any desired environment is possible.

Basic Protocol 1 describes the growth conditions required to go from freezer stock to sample preparation for the time-lapse microscopy; Basic Protocol 2 describes Evos FL Auto 2 setup and how images are taken; Basic Protocol 3 describes how to batch-rename images using the Bulk Rename Utility and R; Basic Protocol 4 describes Orbit setup and how a model is trained to detect the cells; Basic Protocol 5 describes how important parameters (% area at the asymptote, slope, and time to reach the asymptote) are extracted from Orbit output using the R programming language; and Basic Protocol 6 describes how to visualize the logistic fit of the % area and the other important

parameters extracted from Basic Protocol 5. Overall, this approach allows us to extract quantitative parameters from qualitative data, which will facilitate a novel method to interpret time-lapse microscopic data and expand our understanding of biofilm formation in the presence of antifungal drugs. The R scripts detailed here are included as Supporting Information (.R files) alongside an example of their use (.PDF files). The scripts and example image files for practice are also located in a github repository, accessible at *https://github.com/acgerstein/biofilm-formation*.

SAMPLE PREPARATION

Basic Protocol 1 describes sample preparation for use in time-lapse microscopy for one biological replicate from eight strains of interest, each grown in five different concentrations of two different drugs. Any desired 96-well configuration is possible (e.g., 10 concentrations of a single drug, or five concentrations of a single drug with two different types of base media). Basic Protocols 2-5 describe the results from one clinical isolate of *C. albicans* (P75016) grown in five different concentrations of boric acid as an example.

The initial growth from freezer stocks is done in yeast extract peptone dextrose (YPD) in this example, as this medium is the standard for general laboratory growth. 96-well plate preparation for the time-lapse assay is done in RPMI-1640, a standard environment for biofilm growth assays. These environments can be adjusted as desired for the particular experimental goals. Depending on the specific isolates examined and the type of medium used, the growth period may need to be optimized to ensure robust growth prior to the initiation of the time-lapse protocol. This part of the protocol is based on the biofilm-formation assay described in Pierce et al. (2010).

Materials

YPD liquid medium (see recipe)
Candida albicans strain(s) of interest (the protocol has been tested on *C. albicans* strains SC5314, FH1, P87, GC75, P78048, P75016, P76055, T101)
0.01 M phosphate-buffered saline (PBS; see recipe)
RPMI-1640 medium (see recipe)
Stock solution of drug or stress of interest (at least 2× desired maximum concentration)

15-ml conical polypropylene centrifuge tubes (Fisherbrand, cat. no. 14955238)
10-ml glass serological pipette (Fisherbrand, cat. no. 14955234)
30°C shaking incubator (Eppendorf, cat. no. M1299-0094)
Centrifuge (ST16; Thermo Scientific, cat. no. 75004240)
Vortex (Thermo Scientific, cat. no. S28673)
96-well microtiter plate PS F-bottom (Greiner Bio-One, cat. no. 655161)
Sterile pipette tips (Rainin, cat. no. 30389299)
Aliquotting pipet controller (VistaLab, cat. no. 2000-0005)
Spectrophotometer (Multiskan Sky Microplate Spectrophotometer, Life Technologies Inc., cat. no. 51119600) and plate reader
200-μl multichannel pipettor (Pipet-Lite Multi Pipette L12-200XLS+; Rainin, cat. no. 17013810)
Clear seal (Thermo Scientific, cat. no. 14-245-18)
Polystyrene microplate lids (Greiner Bio-One, cat. no. 656176)
Parafilm M wrapping film (Fisher Scientific, cat. no. S37441)
1.5-ml microcentrifuge tubes (Fisherbrand, cat. no. 02682002)

1. Inoculate 10 μl of desired *C. albicans* strain from glycerol stock into 10 ml of YPD in a 15-ml conical polypropylene tube.

Inoculation can also be done from freshly grown colonies streaked from freezer stock onto YPD plates.

2. Place the tube in a shaker (200 rpm) or rotator drum at 30°C and incubate for 24 hr.

   *Some strains may take longer than 24 hr to grow.*

3. Centrifuge 5 min at 3000 × g, room temperature, and remove the supernatant.

    An easy way of removing the supernatant is by inverting the centrifuge tube over a liquid waste container. Do that quickly and cover the tube right away.

4. Add 10 ml of 0.01 M PBS to the tube and vortex for at least 10 s on the highest speed to resuspend the colony.

5. Centrifuge 5 min at 3000 × g, room temperature, and remove the supernatant.

6. Repeat steps 4 and 5 three times to wash the cells. After the third PBS wash, resuspend again in 10 ml of 0.01 M PBS.

7. Transfer 100 µl of each of the washed cultures to a single well of a flat-bottom 96-well microtiter plate and measure the optical density (OD600) using a plate reader.

8. Standardize the cultures in 1.5-ml microcentrifuge tubes to OD_{600} of 0.005 in 1 ml of RPMI-1640.

    Use $C_1V_1 = C_2V_2$ for your calculation. For example, if the OD_{600} of your culture is 0.6, and you want to standardize to 0.005 in 1 ml of RPMI-1640, then $(0.6)(V_1) = (0.005)(1000)$. Therefore, $V_1 = 8.33$ µl ≃ 8 µl, and you would transfer 8 µl of the overnight culture into 992 µl RPMI-1640.

9. Add 100 µl of RPMI-1640 medium to all wells of columns 1-9 of a new flat-bottom 96-well microtiter plate (Fig. 1).
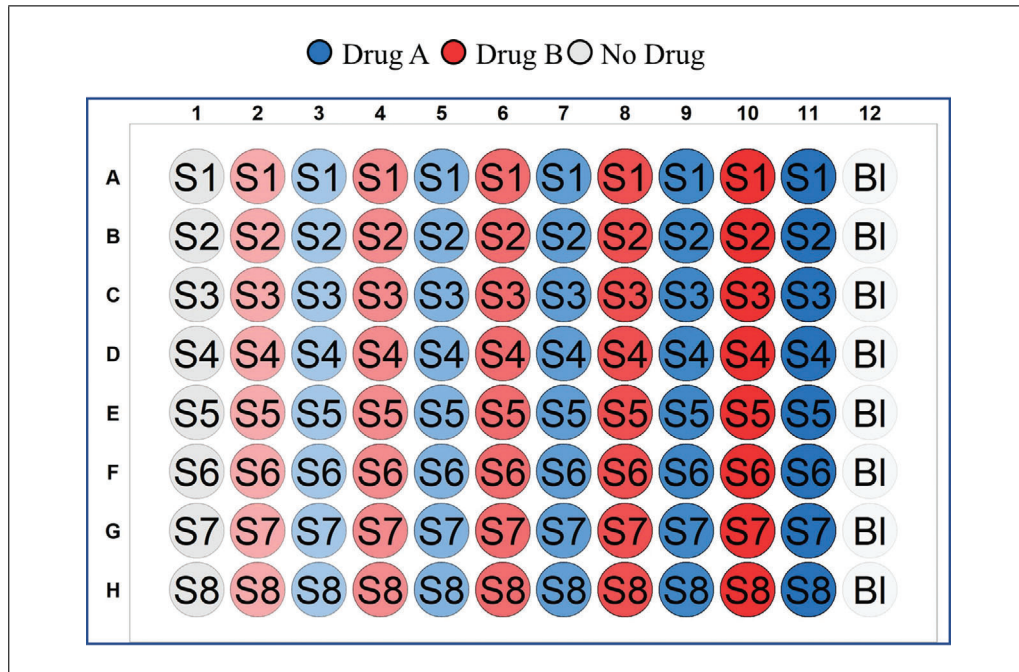
Figure 1 Example plate setup for time-lapse microscopy with 8 different strains (S1-S8) and 2 different drugs (Drug **A** and **B**). To ensure the strains are growing as expected, include wells that contain no drug (column 1). As a contamination check, include blank wells with medium but without cells (column 12; "Bl"). The gradient of the colors from high to low saturation represent the decreasing concentration of drug with each dilution (column 3 has the lowest concentration of drug **A** and column 2 has the lowest concentration of drug **B**).

10. Add 200 µl of RPMI-1640 to column 12. Add 200 µl of RPMI-1640 plus twice the desired final concentration of drug A to all wells of column 11, and add 200 µl of RPMI-1640 plus twice the desired final concentration of drug B to all wells of column 10.

11. Using a multichannel pipettor, do a serial dilution by transferring 100 µl from column 11 to column 9. Mix by pipetting up and down three to four times. Transfer 100 µl from column 9 to column 7 and mix again. Keep doing that for all of the odd-numbered columns until column 3 (REMEMBER to stop at column 3). After mixing column 3, discard the extra 100 µl.

12. Repeat step 11 for column 10, and transfer and mix with even-numbered columns. REMEMBER to stop at column 2. After mixing column 2, discard the extra 100 µl.

    All of the wells of columns 1-11 should now contain 100 µl of medium. Columns 1 and 12 should have no drug. Odd-numbered columns (3-11) contain drug A, with the highest concentration in column 11 and the lowest concentration in column 3. Even-numbered columns (2-10) contain drug B, with the highest concentration in column 10 and the lowest concentration in column 2 (Fig. 1).

13. Transfer 100 µl of standardized culture (from step 8) to all wells of columns 1-11 of the flat-bottom 96-well microtiter plate (step 12). Column 12 should not contain culture (Fig. 1).

    Keep in mind that this will dilute the culture and the drug concentration by 2-fold. For example, if the drug concentration in column 11 is 6.4 mg/ml, the final drug concentration after transferring 100 µl of culture will be 3.2 mg/ml. Similarly, the final well OD_{600} should be 0.0025.

14. Cover with a sterile lid and seal with parafilm.

    To ensure that you are not covering the bottom of the plate with parafilm, use scissors to cut parafilm to the necessary quantity to only cover the sides of the plate.

TIME-LAPSE MICROSCOPY: Evos PROTOCOL

The Evos FL Auto 2 (Thermo Scientific) is used to take images every 1 hr for 24 hr. Between images, the sample plate from Basic Protocol 1 is kept in a 37°C incubator. The Evos can be set up to incubate and automatically scan a 96-well plate for 24 hr without manual intervention if the on-stage incubation is available.

Any microscopy system capable of taking high-resolution images can be used in lieu of the Evos FL Auto 2 Imaging System.

Materials

96-well plate with samples (Basic Protocol 1)
Evos FL Auto 2 (Thermo Scientific, cat. no. AMAFD2000)

Computer with at least 4 GB RAM
Evos FL Auto 2 Imaging System

1. Turn on the machine.

2. Place the 96-well plate containing your samples on the stage.

    Make sure you are using the 96-well plate vessel on the microscope.

3. Open Evos FL Auto 2 Imaging System and click on the **Vessel** (top right). Click on **Well Plates** and select **Well Plate** for the **holder** and **96 Well** for the **Plate**. Click **Done** to finish your vessel set up.

4. Click on a well that contains your sample, select **10 ×** for your **Objective** and **Trans** for your **Light source**.

5. Bring the cursor of the **Bright** to the middle and click on **AutoFocus**. When your sample comes into focus, click on **Capture**.

   After focusing, you can adjust the brightness to suit your needs.

6. To set up a scanning protocol, click on the **Automate tab**. Go to **Hardware → Edit** and select **20 ×** for **Objective** and **Trans** for **Channels**. Click on **Done** when you finish picking your parameters.

7. To adjust the scan area, click on **Select scan areas → Assign scan area**, select the desired area of the plate to be scanned, and click **Done**. Click on **Pattern** and enter **1** in the **W** and **H fields of view**. Check the **Stitching** box and click on **Create**.

8. Open **AutoFocus** and **Z Stacks** and click on **review**. Check the **Use Autofocus** box and select **Single Channels**. Make sure that **Trans** is selected in the **Channel** option and the **Z-Offsets** box is checked. Select the **Every field** option for the **AutoFocus Frequency**. Click **Done** to save your settings.

9. Click on **Image Save Settings → Edit**. Specify the location where you wish to save your images and click on **Done**. You can also save the scan protocol you created for future use by clicking on **Save** in the **Automate tab**.

   *If you wish to use the same protocol in future, in the Automate tab, click on **Load** and navigate to the folder containing your saved protocol. Select the desired protocol and click **Open**.*

10. Click on **Run** to scan your plate.

    This is the 0 hr (t0) reading. Every new scan will be exported into its own unique folder. Images will be named according to their position (i.e., images of well G11 will be named G11.TIF*). The default naming convention for Evos is* scan.date-time
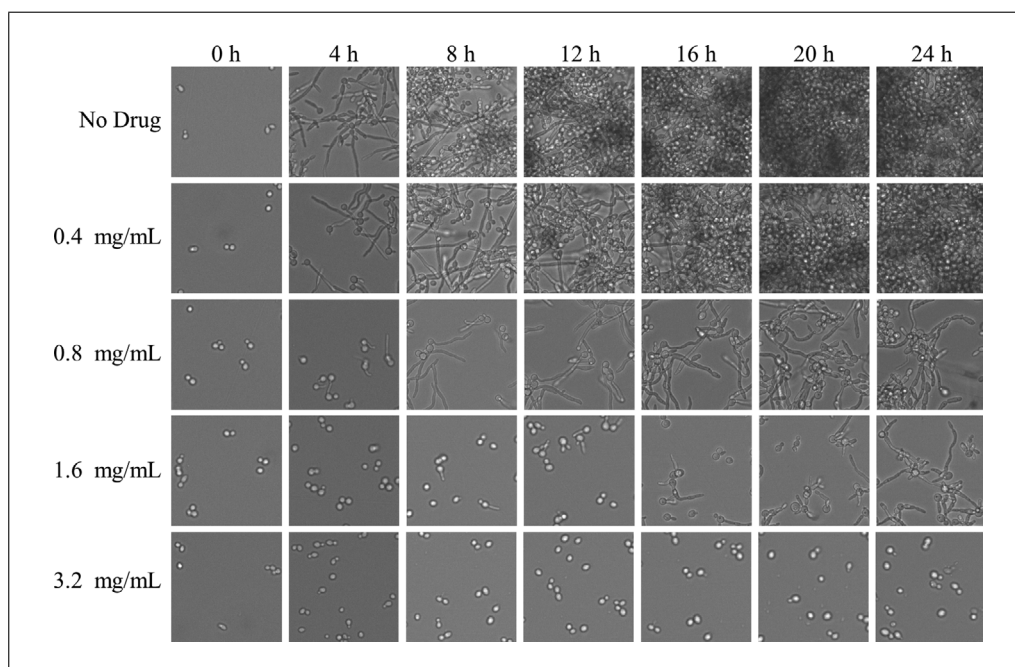
Figure 2   Images were captured every hour for 24 hr using the Evos FL 2 Imaging System. Shown are sample images of a clinical *Candida albicans* strain grown in RPMI-1640 plus boric acid at 37°C.

(e.g., `scan.2021-02-16-16-22-15`). It is recommended that you change the name of the folder that contains your images to a name that indicates the time point.

11. Incubate the plate statically at 37°C for 24 hr and take images hourly using the Evos FL Auto 2. As long as the machine settings have not been changed, at each desired time point click **Run** to scan your plate. If the settings have been changed between images, just reopen the saved protocol (step 9).

     If on-stage incubation is set up then the plate will stay in the machine and readings can be automatically taken every hour.

12. If you are doing a 24 hr time-lapse, you will have 25 folders by the end of your experiment, which each contain 96 images (example images in Fig. 2).

BATCH FILE RENAMING

Time-lapse of a 96-well plate with images taken every hour for 24 hr produces 2304 images; this number will be even larger if the experiment is repeated to get additional biological replicates. The Evos automatically names images by their position and splits different time points into different folders (t0-t24); all images will need to be put into one folder for Basic Protocol 4. Renaming each image manually is a time-consuming and tedious task that is prone to typos. Described below is an efficient way to automate file renaming using a combination of the free Windows filename program "Bulk Rename Utility" and the R Programming Language. As described below, files can be renamed on a Mac without downloading an additional program.

Materials

Computer with at least 4 GB RAM
Bulk Rename Utility (*https://www.bulkrenameutility.co.uk/*).
R (*https://www.r-project.org/*).
RStudio (*https://www.rstudio.com/products/rstudio/download/*).
tidyverse package (Wickham et al., 2019).
TIFF images from Evos output (see Basic Protocol 2; other image formats should also be fine)
`.csv` file that contains plate layout (see Table 1 for an example)
`BP3-BatchRenaming.R` (Supporting Information)

1. Download Bulk Rename Utility from *https://www.bulkrenameutility.co.uk/*. Follow the installation instructions.

    The installer has both the 32-bit and the 64-bit version. Bulk Rename Utility is only available for Windows 7, 8, and 10. Files can be bulk-renamed in Mac through Finder.

2. Open Bulk Rename Utility and go to the folder that contains your images.

    For example, go to the folder that has the images from time point 1. On a Mac, open Finder and go to the folder that contains your images.

Table 1   Plate Layout Example

Well	Strain	Drug	Concentration
A01	SC5314	ND	0
A02	SC5314	Drug-A	2
A03	SC5314	Drug-B	0.4
A04	SC5314	Drug-A	4
A05	SC5314	Drug-B	0.8
A06	SC5314	Drug-A	8

Salama and Gerstein

3. In the **Add (7)** panel, type underscore plus the time point in the **Suffix**. Select all of the images in the folder and click **Rename**.

   *Before clicking Rename, check the **New Name** column and make sure this is what the name should look like. The names of the files should be well position plus time point plus image file format (for example, G11_t01.TIF).*

   *On a Mac, click on **Action** → **Rename Items** → **Add text** to add the appropriate time-point as a suffix as above.*

4. Repeat step 3 for all of the folders that contain the different time points.

5. Put all of the images into one folder.

   Now, all of the images have unique names and will not overwrite each other when put together in one folder.

   Before proceeding to the next step, you will need to install R and RStudio if you have not already done so. You will also need to create a .csv file that contains your plate layout (Table 1). You will be using that .csv file to rename your files. You can add more or less information to the file names by adding or removing columns. For example, you can add a 'Replicate' column or remove the 'Drug' column. At the end of Basic Protocol 4, your files will look something like this: Calb_DrugA_2_t01.TIF (Strain_Drug_Concentration_timepoint.TIF). It is strongly recommended that you keep a backup of your images in case you misname them!

6. Open RStudio and load the tidyverse library.

   ```
   library(tidyverse)
   ```

   If this is the first time you are using tidyverse, you will need to install the package using the install.packages("tidyverse"). You only need to install the package once; however, you will need to load the tidyverse library every time you exit your R session.

7. Load the .csv file that contains your plate layout. Click on **Import Dataset** → **From Text (readr)** → **Browse**. Find the plate layout file and click **Open**. In the **Import Options**, change the name to a simple name such as plate_layout. Click on **Import** when you are done.

8. Go to **Session** → **Set Working Directory** → **Choose Directory**. Open the folder that contains the images you want to rename and click **Open**.

9. Run list.files(pattern = "*.TIF") and make sure that the files returned in the R console are the ones you want to rename.

   If your images are not TIFF files, change the pattern to match your image format (for example, if you have .jpg files, your code would change to: list.files(pattern = ".jpg")*

10. Create a dataframe (Evos_tbl) that contains a column ("EvosNames") with the image names (e.g., A01_t01.TIF):

    ```
    Evos_tbl <- tibble(EvosNames =
            unlist(list.files(pattern = "*.TIF")))
    ```

11. Create a dataframe (Evos_separate) with a separate column for each piece of information (i.e., one column for well, one column for time point, and one column with the image type). This information is then attached to the dataframe Evos_tbl created in step 10.

    ```
    Evos_separate <- Evos_tbl %>%
        separate(EvosNames,
    ```

```
            into = c("well", "timepoint", "image")) %>%
        cbind(Evos_tbl)
```

12. Join `plate_layout` dataframe with `Evos_separate`. Create a new column ("NewFullName") that combines the strain, drug, concentration, and time point columns. The final dataframe, `Full`, is then selected to keep only 2 columns; column 1, "EvosNames," will contain the existing file names and column 2, "NewFullName," will contain the corresponding new name (strain_drug_concentration_time point).

```
    Full <- Evos_separate %>%
        merge(plate_layout, by = "well") %>%
        unite(col = "NewFullName", c(strain, drug,
                        concentration, timepoint),
                sep = "_", na.rm = TRUE) %>%
        select(EvosNames, NewFullName)
```

13. Rename the files using the `Full` dataframe.

```
    file.rename(from = Full$EvosNames,
            to = paste0(Full$NewFullName, ".TIF"))
```

After running this line, R will look for the file names matching the EvosNames column and will replace them with the corresponding new names. If there is an image that has a name not listed in EvosNames, it will not be renamed. If your images are not TIF files, change ".TIF" in the paste0 argument to the image format you are using (for example, if your images are .jpg, then change it to paste0(Full$NewFullName, ".jpg"). By the end of this step, all of your images will have a new name.

MACHINE LEARNING ANALYSIS OF Evos IMAGES WITH ORBIT

Time-lapse experiments of 96-well plates produce large numbers of images. Analyzing those images by eye is time consuming and potentially subjective. We developed a novel quantitative pipeline to measure parameters that correlate with adherence and biofilm formation. Orbit Image Analysis is a user-friendly tool that uses machine learning to detect cells of interest. In this protocol, Orbit is trained to detect fungal cells and calculate the % area covered by cells from time-lapse microscopy images.

Materials

Hardware

Computer with at least 8 GB RAM

Software

Orbit Image Analysis version 3.64 or higher

Files

TIFF images from Evos output (other image formats should also be fine)

Download and install Orbit

1. Go to *https://www.orbit.bio/download/*, look for your OS, and press download (if you are using a 32-bit OS, download the JAR version).

2. Launch `orbit_xxx.exe` and follow the installation instructions.

Importing images and defining classes

3. Open Orbit by double-clicking the icon. Orbit will ask whether you want to configure an Omero image server; press No if you do not have one (Fig. 3A). When Orbit asks whether a local filesystem fallback is used, press Continue (Fig. 3B).

Figure 3 Orbit Image Analysis interface running on PC Windows 10. (**A**) Open Orbit and click on No when the "Configure Omero Server?" message pops up. (**B**) Click on Continue when "Image provider not available, using a local filesystem fallback" message pops up. (**C**) Now you are in the main Orbit interface with the drop-down class menu on the top left and the menu bar just below. (**C** and **D**) To open an image, select Image from the menu bar and click on Open Image from File System. Select the image/images and click Open.

4. Go to **Image → Open Image from File System**. Select a few images or an image that contains all of the cell types that need to be detected (Fig. 3C and 3D). Click **Open** when you are done.

   Typically, three images are sufficient for training: one from t0, one that has an intermediate concentration of cells (often t08), and one that is highly saturated with cells (often t24).

5. In the menu bar, go to **Model → Classes** (or press **F2**). Define the different classes in your images (e.g., Background and Fungi). The default is three classes: **Background**, **Celltype1**, and **Celltype2**. To rename a class, click on the class to be renamed, write the new name, and press **rename class**. You can add a class by pressing on **add class** or remove one by pressing on **remove class**. Press **OK** when you are done with defining classes. In this protocol, the images have two classes: Background and Fungi (Fig. 4A).

Training and saving a model

6. Select class Background from the drop-down class menu (top-left). From the menu bar, go to **Classification** and choose **Polygon**. You have to mark representative regions of Background (Fig. 4A and 4B). When you are defining the background, try to come close to the cells.

7. Repeat step 6 but change the class to Fungi. When you are defining class Fungi, try as much as possible to not include background in the selected regions.
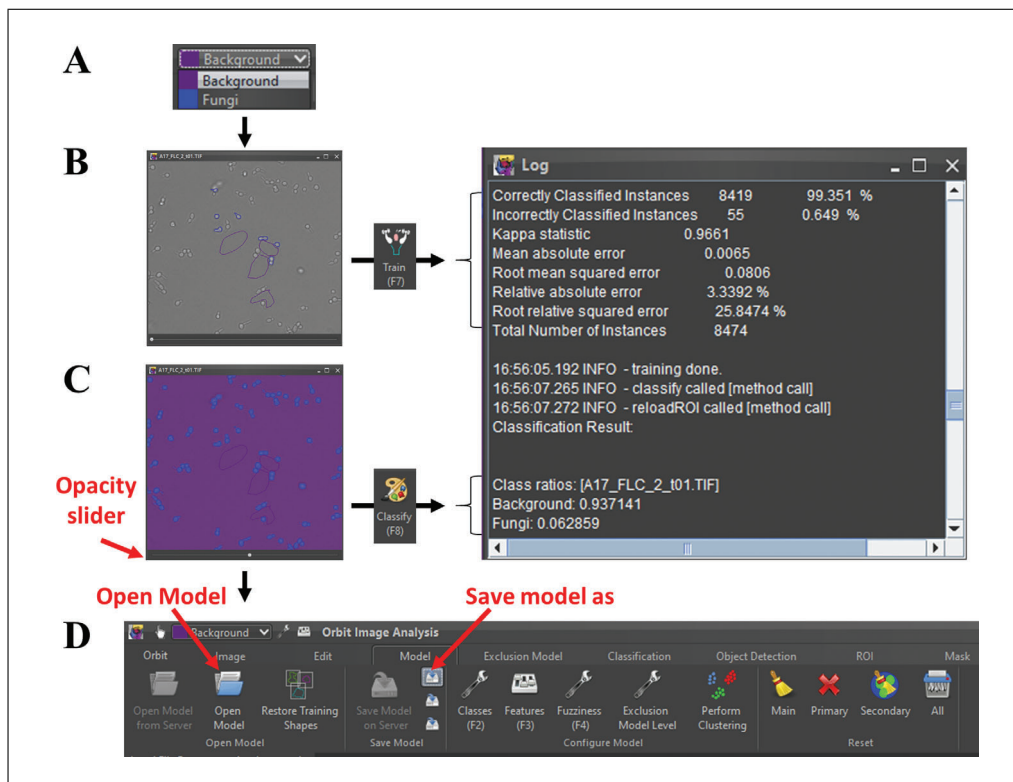
Figure 4 Training the machine learning algorithm based on the selected classes. (**A** and **B**) Select representative regions for background (purple) and fungal cells (blue) using the polygon tool. Train a classification model by clicking on Train. The log output indicates model fit; in this case, 99.3% of instances were correctly classified. (**C**) Test the model by clicking on Classify; use the opacity slider to check if the model is capable of differentiating between the background and cells. The log output indicates the % area covered by fungal cells and the percent area of the background. (**D**) Save the model by going to Model in the menu bar and clicking on 'Save model as'.

8. Repeat steps 6 and 7 for all opened images.

9. Press on **Train (F7)** to train a model to detect the defined classes.

   Training will be done only on opened images. If you close an image that had the different classes/regions selected, it will not be included in the training.

10. Go to **Help**, click on **Show Log**, and check the quality of your model.

11. Go back to the **Classification** panel and click on **Classify (F9)**. Drag the Opacity slider to check whether Orbit detected the right objects (Fig. 4C).

    If you are not happy with the detection model, repeat steps 6-11. Select more regions representative of your cells and background. Double check that you are using the right class when using the polygon and marking your cells and background. A common mistake is marking your cells with the polygon tool while selecting the wrong class from the drop-down class menu.

12. Open an image that was not included in the training, click on **Classify**, and check whether the model that has been created can still accurately identify cells. Use the opacity slider to check the quality of your detection model.

    *If you are not satisfied with the model, repeat steps 6-12. It is recommended that you include the image in which Orbit had trouble detecting cells in your opened images, in order to include it in the training model. Open a different image when the training is done, and click on **Classify** to check if the newly created model is better. Keep in mind*

Figure 5 Batch Analysis and data export. (**A**) After creating a successful model, run a batch analysis by going to Batch and clicking on Local Execution. (**B**) Select all of the images to be analyzed. (**C**) Click on Yes when a "Starting Batch Mode" message pops up. (**D**) When the analysis is done, a "Classification" output window will pop up. (**D**) Select the classification output using Ctrl + A and (**E**) copy to Excel. Save the excel file as a comma-delimited (`.csv`) file.

> *that every time you click **Train**, you are overwriting the previous model, and any image you close will not be included in the new model.*

13. Go to Model and click on Save Model as (Fig. 4D).

    *You can load your saved model anytime using **Open Model**.*

Batch analysis

14. Go to **Batch** and click on **Local Execution** (Fig. 5A). Select all of the images to be analyzed and click on **Open** (Fig. 5B). Orbit will ask you whether the model you are applying is a ratio quantification model—click on **Yes** (Fig. 5C). This step may take up a few hours, and you can check that it is running by going to **Help → Show Log**. Orbit will eventually pop up a window with the classification results (Fig. 5D). Copy the results to an Excel file and save it as a comma-delimited (`.csv`) file (Fig. 5E). The quantification model calculates the % area covered by cells for each time-lapse image.

PARAMETRIZATION OF ORBIT OUTPUT IN R

In this protocol, the growthcurver R package (Sprouffske, 2020) is used to fit a logistic equation to the % area data output from Basic Protocol 4. From the logistic fit determined for each well, we extract relevant parameters that correlate with biofilm

formation: growth rate (measured as the slope of cellular growth, measured as % area covered); growth plateau (the asymptote of % area covered); and time to reach the asymptote. Prior to running the logistic fit function, the data are cleaned using functions from the tidyverse R metapackage (Wickham et al., 2019). Here we show how the Orbit data are cleaned, how to extract the asymptote and slope, and how to calculate the time to reach the asymptote.

Materials

Hardware

Computer with at least 4 GB RAM

Software

R (*https://www.r-project.org/*)
RStudio (*https://www.rstudio.com/products/rstudio/download/*)
The tidyverse package (Wickham et al., 2019)
The growthcurver package (Sprouffske, 2020)

Files

`.csv` file that contains Orbit output (see `OrbitOutput_SampleData.csv` found in Supporting Information for mock Orbit sample)
`BP5-OrbitParameterization.R` (Supporting Information)

1. Open RStudio. Go to **View → Panes → Show All Panes**.

   Depending on your setup, this may be an unnecessary step, as all panes may already be showing.

2. Go to **Session → Set Working Directory → Choose Directory** and find the folder that contains the Orbit output `.csv` file. Select the folder and click **Open**.

   Any files and images that you will save will be saved to that working directory.

3. Load the tidyverse and growthcurver libraries.

   ```
   library(tidyverse)
   library(growthcurver)
   ```

   If this is the first time using these libraries, you will have to install them first: `install.packages("tidyverse")`.

   `install.packages("growthcurver")`.

4. Under the **Environment** panel, click on **Import Dataset → From Text(readr)**. Click on **Browse** and look for the folder containing your Orbit output `.csv` file. Select the file and click on **Open**. At the bottom left, find **Import Options** and change the **Name** to **df** (which stands for dataframe). Click on **Import** when you are done. Under the Environment panel, you should be able to see df data that were just imported.

   A pop-up view window of the dataset will appear after you click Import. Double check that the right data are being imported and being read correctly.

   You can also import datasets programmatically using a `read_csv()` *function call.*

5. The file name format described in Basic Protocol 3 contains information about: strain, drug, drug concentration, and the time point at which the image was captured. Separate this information into different columns using tidyverse functions. The end result is a dataframe (`df_cleaned`) that contains columns for each component of your file name and % area covered by cells.

```
df_clean <- df %>%
    separate(Filename, c("strain", "drug",
      "concentration",
                          "timepoint"), sep = "_") %>%
    separate(timepoint, c("timepoint",
      "image_format")) %>%
    separate(timepoint, c("t", "time"), sep = "t")
      %>%
    select(strain, drug, concentration, time, Fungi)
      %>%
    rename("area" = "Fungi") %>%
    mutate_each(funs(as.numeric), time)
```

If the file name format is different from what is described in Basic Protocol 3, the code needs to be tweaked. For example, if your file name does not contain drug *and* concentration, *change the code to:*

```
df_clean <- df %>%
    separate(Filename, c("strain",
                          "timepoint"), sep = "_") %>%
    separate(timepoint, c("timepoint",
      "image_format")) %>%
    separate(timepoint, c("t", "time"), sep = "t")
      %>%
    select(strain, time, Fungi) %>%
    rename("area" = "Fungi") %>%
    mutate_each(funs(as.numeric), time)
```

6. Create a new dataframe (df_0) where the % area at the first time point (t0) has been subtracted from all time points.

    To fit a logistic equation, the % area at the first time point has to be equal to zero. The subtracted area will be added back after model fitting.

    ```
    t0 <- df_clean %>%
        filter(time == 0)

    df_0 <- merge(df_clean, t0, by = c("strain","drug",
                                       "concentration")) %>%
        select(!time.y) %>%
        mutate(area = area.x - area.y) %>%
        select(c(strain, drug, concentration, time.x,
          area)) %>%
        rename("time" = "time.x")
    ```

7. Convert df_0 from a long dataframe to a wide dataframe (df_0_wide). This way each column contains the data for all time points from each well, which is required for the next step.

    ```
    df_0_wide <- df_0 %>%
        unite(sample, strain, drug, concentration) %>%
        pivot_wider(names_from = sample, values_from =
          area)
    ```

8. Obtain the summary metrics of the logistic fit to each well by running the SummarizeGrowthByPlate() function (from the growthcurver R package), saved in the logfit_params0 dataframe. View logfit_params0 to see the resulting dataframe: "sample" corresponds to the file name, "r" is the slope of the logistic fit

(rate of growth), "k" is the asymptote of the logistic fit, and "t_gen" is the generation time (time it takes for a population to double in size).

```
logfit_params0 <- SummarizeGrowthByPlate(df_0_wide)
view(logfit_params0)
```

9. Create a new dataframe `logfit_params` that combines the `t0` with `logfit_params0` dataframes, separates the sample column into strain, drug, and drug concentration, and contains and adds back the subtracted area from time point zero (t0) to the asymptote (k).

```
logfit_params <- unite(t0, sample, strain, drug,
concentration) %>%
  merge(logfit_params0) %>%
  group_by(sample) %>%
  mutate(k = area + k) %>%
  select(!time & !area) %>%
  separate(sample, into = c("strain", "drug",
                            "concentration"), sep = "_")
```

10. Check the notes column of logfit_params. If the `SummarizeGrowthByPlate()` function was unable to fit a logistic equation to a given well, it will say "cannot fit data"; rerun step 7 and exclude those wells.
    For example, if you want to remove sample Calb_FLC_16:

```
df_0_wide <- df_0 %>%
  unite(sample, strain, drug, concentration) %>%
  filter(!sample == "Calb_FLC_16") %>%
  pivot_wider(names_from = sample, values_from =
    area)
```

If you want to remove multiple samples, adjust the filter function to include the samples you want to remove:

```
filter(!sample %in% c("S2_FLC_1", "S7_FLC_2",
  "S11_FLC_1"))
```

11. Create a new dataframe, `df_predicted_plate`, that contains predicted y values (predicted % area based on the best fitting logistic equation) for each model fit (obtained by applying the `SummarizeGrowth()` function to all wells). This is used for plotting.

```
models_all <- lapply(df_0_wide[2:ncol(df_0_wide)],
function(x) SummarizeGrowth(df_0_wide$time, x)
)
df_predicted_plate <- data.frame(time =
  df_0_wide$time)
for (i in names(df_0_wide[2:ncol(df_0_wide)])){
    df_predicted_plate[[i]] <-
        stats:::predict(models_all[[i]]$model)
    }
```

12. Create a dataframe, `df_0_logfit`, that contains the % area and predicted % area from the logistic fit.

```
df_0_logfit <- cbind(
  pivot_longer(df_0_wide, !time,
                  names_to = "sample",
                  values_to = "area"),
  pred_area = pivot_longer(df_predicted_plate, !time,
```

```
                    names_to = "sample",
                    values_to = "pred_area")[,3]) %>%
        separate(sample,
          c("strain", "drug", "concentration"), "_")
```

13. Create another dataframe df_final that adds the subtracted t0 % area back to the area and predicted % area columns.

```
df_final <- merge(df_0_logfit, t0,
                          by = c("strain", "drug",
                                 "concentration")) %>%
    select(!time.y) %>%
    rename("time" = "time.x") %>%
    group_by(strain, drug, concentration, time) %>%
    summarise(area = area.x + area.y,
              pred_area = pred_area + area.y)
```

14. Save df_final as a .csv file.

```
write_csv(df_final,
        file = "example_data_predicted.csv")
```

   The file will be saved to the folder you set as your working directory in step 2. Note that the file you save can be named whatever you like.

15. Create a new dataframe, logfit_params_corrected, that contains the previously calculated logistic fit parameters and uses the predicted areas calculated above to determine the time it took to reach the asymptote. If you have wells with no or low growth, two additional columns, k_corrected and r_corrected, are used to account for poor or incorrect model fit. Respectively, they set the % area at asymptote (k) to be equal to the % area at the end time point and the rate of growth (r) to be equal to 0. t_asym (the time it takes to reach k) is also adjusted to be equal to the final time point + 4 (this is used for visualization, below). If you have robust growth in all wells, data in the k_corrected and r_corrected columns should be the same as in the k and r columns. Save logfit_params_corrected as a .csv file.

```
logfit_corrected <- logfit_params %>%
  select(strain, drug, concentration, k, r) %>%
  merge(df_final, by = c("strain", "drug",
                    "concentration")) %>%
  mutate(k_corrected = replace(k, round(k, digits =
    2) > 1,
                    pred_area[which(time == "24")]),
      r_corrected = replace(r, k < 0.15, 0),
      k = round(k, digits = 2),
      pred_area = round(pred_area, digits = 2)) %>%
  group_by(strain, concentration, drug) %>%
  summarise(t_asym = time[which.min(abs (pred_area -
    k))],
          k_corrected = mean(k_corrected),
          r_corrected = mean(r_corrected)) %>%
  mutate(t_asym = replace(t_asym, r_corrected == 0,
    28))
logfit_params_corrected <- merge(logfit_corrected,
                            logfit_params,
                          by = c("strain", "drug",
                                "concentration"))
```

```
write_csv(logfit_params_corrected,
          file = "example_data_logfit.csv")
```

If your experiment is not 24 hr, change the number in `pred_area[which(time == "24")])` *to the experiment length (e.g.,* `pred_area[which(time == "48")])`.

If you have wells with low or no growth and your experiment is not 24 hr, change "28" in `mutate(t_asym = replace(t_asym, r_corrected == 0, 28))` *to the length of your experiment + 4 (e.g., for a 48-hr experiment,* `mutate(t_asym = replace(t_asym, r_corrected == 0, 52))`*).*

The file will be saved to the folder you set as your working directory in step 2.

Note the file you save can be named whatever you like.

VISUALIZATION OF LOGISTIC FITS IN R

Visualization of the data is an important step in data analysis; it makes it easy to identify patterns, provides valuable insights about the data, and is required to communicate the results to your intended audience. In this protocol, `ggplot()` (part of the tidyverse package) will be used to visualize the output data from Basic Protocol 5.

Materials

Hardware

Computer with at least 4 GB RAM

Software

R and RStudio
The tidyverse package (Wickham et al., 2019).

Files

The two `.csv` files output from Basic Protocol 5
`BP6-LogFitParamsVis.R` (Supporting Information)

1. Open RStudio and go to **Session → Set Working Directory → Choose Directory** and find the folder that contains the Basic Protocol 5 output `.csv` files. Select the folder and click **Open**.

   You can skip this step if you are on the same R session used in Basic Protocol 5.

2. Load the tidyverse library.

   ```
   library(tidyverse)
   ```
   You can skip this step if you are on the same R session used in Basic Protocol 5. You only need to load the tidyverse library.

3. Read in the `.csv` files you saved in steps 14 and 15 in Basic Protocol 5. You will be using those files for visualizing your data.

   *Import the data using **Import Dataset → From Text(readr)**; remember to change the name in the **Import Options** to* `df_final` *and* `logfit_params_corrected`. *You can skip this step if you are on the same R session used in Basic Protocol 5.*

4. Plot the % area covered by cells for each time point (one panel for each strain) and a line of the logistic fit using `ggplot()`.

   ```
   df_final %>%
     ggplot(aes(x = time, y = area, color = concentration,
     group = concentration))+
     geom_point() +
   ```

```
geom_line(aes(y = pred_area)) +
scale_x_continuous(breaks = seq(0, 24, 4)) +
facet_wrap(~strain)+
xlab("Time (hours)") +
ylab("Area % Covered by Cells") +
scale_color_gradient(low = "blue", high = "red") +
theme_bw()
```

If you have multiple conditions (e.g., drugs) and you want a separate panel for each, you can change the `facet_wrap` function to, e.g., `facet_grid(drug ~ strain)`. If your incubation time is not 24 hr, change 24 in `scale_x_continuous()` to your incubation time. For example, if your incubation time is 48 hr, change `scale_x_continuous()` to `scale_x_continuous(breaks = seq(0, 48, 4))`.

5. Plot the parameters that were extracted from the logistic fit (asymptote, slope, and time to reach the asymptote):

```
# plot the asymtotes
logfit_params_corrected %>%
    ggplot(aes(x = concentration, y = k_corrected,
            group = strain)) +
    geom_point(size = 2) +
    geom_line() +
    facet_wrap(~strain) +
    xlab("BA Concentration (mg/ml)") +
    ylab("% Area at Asymptote") +
    theme_bw()
# plot the slopes
logfit_params_corrected %>%
    ggplot(aes(x = concentration, y = r_corrected,
      group = strain)) +
    geom_point(size = 2) +
    geom_line() +
    facet_wrap(~strain) +
    xlab("BA Concentration (mg/ml)") +
    ylab("Slope") +
    theme_bw()
# plot the time to reach the asymptote
logfit_params_corrected %>%
    ggplot(aes(x = concentration, y = t_asym,
      group = strain)) +
    geom_point(aes(shape = t_asym > 24), size = 2) +
    geom_line() +
    scale_y_continuous(breaks = seq(0, 28, 4),
                labels = c(seq(0, 24, 4), "> 24"),
                limits = c(0, 28)) +
    scale_shape_manual(values = c(16, 1)) +
    facet_wrap(~strain) +
    xlab("BA Concentration (mg/ml)") +
    ylab("Time to Asymptote (h)") +
    theme_bw() +
    theme(legend.position = "none")
```

If you have multiple conditions (e.g., drugs) and you want a separate panel for each, you can change the `facet_wrap` function to, e.g., `facet_grid(drug ~ strain)`.

When plotting time to reach the asymptote, if your experiment is not 24 hr, change "24" in the `geom_point()` *and* `scale_y_continuous()` *functions to the appropriate numbers (as in Basic Protocol 5). For example, if your experiment is 48 hr, change to* `geom_point(aes(shape = t_asym > 48), size = 2)` *and* `scale_y_continuous()` *to* `scale_y_continuous(breaks = seq(0, 52, 4), labels = c(seq(0, 48, 4), "> 48"), limits = c(0, 52))`*. The* `geom_point()` *code changes the point type for wells with poor or low growth to be a different point type than other wells (point types are specified in the* `scale_shape_manual()` *function). If all wells have robust growth, the "limits" flag in the* `scale_y_continuous()` *function can be changed to, e.g.,* `c(0, 24)` *(for a 24-hr experiment).*

6. Save the plots you have made by clicking on **Export** in the Plots panel (typically on the bottom right) and click on **Save as Image**. You can change the file name and image format, width, and height of the plot you wish to save. Click on **Directory** to select the folder in which you want to save your image, and click **Open** after selecting the folder. Click **Save** to save the image.

   Use the arrows in the Plots panel to navigate through your plots. You have to repeat this step for each plot.

REAGENTS AND SOLUTIONS

Ampicillin stock, 1000×

750 µg of ampicillin sodium salt (Fisher BioReagents, cat. no. BP176025)
7.5 ml of dH$_2$O
Divide into 750-µl aliquots
Store up to 1 year at 4°C

Chloramphenicol stock, 1000×

187.5 mg of chloramphenicol (Fisher BioReagents, cat. no. BP904-100)
7.5 ml of 95% ethanol (Greenfield Global, cat. no. P025EA95)
Divide into 750-µl aliquots
Store up to 1 year at 4°C

Glucose, 40%

20 ml of dH$_2$O
Dissolve 15 g dextrose (Fisher Chemical, cat. no. D16500) using stir bar
Bring volume to 37.5 ml using dH$_2$O
Autoclave
Divide into 750-µl aliquots
Store up to 1 year at room temperature

Phosphate-buffered saline (PBS), 0.01 M

Dissolve two PBS tablets (BioReagents, cat. no. BP308100) in 400 ml of dH$_2$O. Autoclave. Store up to 1 year at room temperature.

RPMI-1640 medium

10.4 g of RPMI-1640 powder (Sigma-Aldrich, cat. no. R6504)
15.0 g dextrose (Fisher Chemical, cat. no. D16500)
17.3 g of MOPS (Fisher BioReagents, cat. no. BP308-100)
pH to 7 using NaOH (Fisher Chemical, cat. no. S318-500)
Filter sterilize
Store up to 1 year at 4°C

YPD liquid medium

15 g Bacto Peptone (Gibco, cat. no. DF0118-17-0)
7.5 g Yeast Extract (Gibco, cat. no. 288610)
120 mg adenine hemisulfate (Thermo Fisher Scientific, cat. no. A1696409)
60 mg uridine (Sigma-Aldrich, cat. no. U3750)
712.5 ml dH$_2$O
Autoclave
Add 750 µl of 1000× ampicillin stock (see recipe)
Add 750 µl of 1000× chloramphenicol stock (see recipe)
Add 37.5 ml of 40% glucose stock (see recipe) when ready to use
Store up to 1 year at room temperature

COMMENTARY

Background Information

Sample preparation

Common growth conditions that work for a large number of *C. albicans* strains have been used to grow the culture from frozen stock. YPD is an inexpensive, easy-to-make rich medium that can be stored for a long time and is commonly used to grow strains from frozen stock. RPMI-1640 is a standard medium used in susceptibility testing. Compared to other media, RPMI-1640 enhances hyphal formation and induces biofilm growth in *C. albicans* (Kucharíková, Tournu, Lagrou, Van Dijck, & Bujdáková, 2011; Weerasekera et al., 2016). The described growth conditions result in consistent biofilms between different biological replicates. The sample preparation protocol for biofilm formation was adapted from Pierce et al. (2010) with minor modifications. Since the goal was to examine the drug effect on biofilm formation (not just drug penetration of preformed biofilms), the cells were treated with drug from the beginning to examine how biofilms form under biofilm-stimulating conditions.

Time-lapse microscopy and Orbit Image Analysis

C. albicans forms highly structured biofilms under biofilm-stimulating conditions, and the drug response at the early steps of biofilm formation can be visualized using time-lapse microscopy. Time-lapse is a powerful tool that provides spatiotemporal data, and if done using a microscope capable of scanning a 96-well plate, it is a high-throughput tool for studying cells. When combined with image analysis, time-lapse can provide both qualitative and quantitative data. Orbit Image Analysis is a new open-source analysis tool that can extract quantitative data from time-lapse images. It includes a machine-learning algorithm that can be easily trained to detect cells (Stritt et al., 2020). It can be downloaded on any computer with at least 8 GB RAM. This facilitates image analysis on a personal computer.

R for renaming, parametrization, and visualization

The R programming language is widely used in many disciplines, including biology and statistics. It is a powerful tool to clean, analyze, and visualize data. It is free and can run on all major operating systems. R packages are extensions of the core R functionality that are available for free. We use the tidyverse and growthcurver packages (Sprouffske, 2020; Wickham et al., 2019). The tidyverse is a metapackage (collection of R packages), which includes dpylr, the main tidyverse package used for cleaning the data in Basic Protocol 5, and ggplot2, which is used for visualization in Basic Protocol 6. The growthcurver package is used to fit a logistic equation to the % area data and extract important parameters from the Orbit output, including the asymptote and the slope in Basic Protocol 5. Custom scripts in Basic Protocols 5 and 6 are provided for additional parameterization (growth rate, % area at asymptote, time to reach the asymptote), and to give an example of the visualization that is possible.

Critical Parameters and Troubleshooting

Sample preparation

Sterile technique is critical for Basic Protocol 1. It is important to include blank wells to ensure that the medium has not been contaminated. It is recommended to always start by growing up culture from freezer stocks, as mutations (including chromosomal aneuploidy) can arise rapidly in *C. albicans* cultures. In most cases, culture for biofilm assays should typically be grown for less than 18 hr

from stationary phase or frozen culture; however, if culture is not visible in liquid YPD, a longer growth period may be necessary. To ensure accurate spectrophotometer readings, it is often required to dilute the overnight culture. Actual OD_{600} values vary depending on the strain, species, and growth conditions. In the biofilm assay, cells in the control column (no drug) should exhibit robust and rapid growth, and the blank column should remain cell-free for the duration.

Time-lapse microscopy

It is important to use the same machine and scanning protocol for biological replicates. If the scanning protocol is done manually, as described here (i.e., the plate is manually taken out of an incubator every 1 hr for a scan), ensure that the incubation temperature is consistent. The interval between readings should also be kept as constant as possible. If the microscope has an autofocus option, for consistency it should be used for every scan, not just the first reading. *C. albicans* clinical strains in some environments (e.g., no drug) form hyphae within the first hour; thus, it is important to minimize the time between setting up the 96-well plate and the first microscope scan of the plate. It is thus recommended that you set the scanning protocol before beginning, using a mock plate that contains some cells.

Batch file renaming

Renaming files through Bulk Rename Utility and R is irreversible; thus, it is important to keep a backup of the images that are being renamed, in case of error. Consider doing a mock test on a subset of files before renaming all of your raw images. Some of the common mistakes in renaming through R are not setting the working directory to the folder containing the images to be renamed, not loading the tidyverse library, not installing the tidyverse library for the first time, and not changing the image format (e.g., from `.TIF` to `.jpg`). See the file `BP3-BatchRenaming.pdf` (Supporting Information) to see how the script should work. It is recommended that you avoid spaces or special characters in image file names, to ensure code compatibility.

Orbit Model training

Image analysis in Orbit and the quality of the detection model is highly dependent on the image quality. Focus, illumination, magnification, and scanning time intervals should be kept consistent between biological replicates.

The polygon object selection step is critical for the accuracy of the model. Use the slider to check the detection model that you created, and ensure that the model properly differentiates between the cells and the background. If you are not satisfied with the detection model, tweak your selections (i.e., select more cells and more background regions) and train the model again. Model quality will also depend on cell saturation. Sometimes, when the whole field of view is covered by cells, Orbit will detect the cells as background (which will influence the logistic fit parameterization). To help address this, highly saturated images should be included in the model training. It is important to test the model on images that are not included in the training set before proceeding to the batch analysis step. If you are using a version of Mac OS, you might experience a very slow response of Orbit if you are using `.TIF` images. To address this issue, convert the `.TIF` files to `.jpg` files.

Parametrization and visualization using R

Parameters such as slope and % area at the asymptote are extracted using the growthcurver package, and parameter fits are visualized using the tidyverse package. If this is your first time using these packages, ensure that they are installed and loaded. Remember that the packages need to be loaded in every R session (this is a common error). If the predicted logistic does not fit the data properly, check whether you initially forgot to subtract the t0 % area and whether you added the subtracted t0 % area back to both the predicted % area and the measured % area. Similarly, if the asymptote plotted by eye does match the value in the `logfit_params_corrected` dataset, check whether you added the t0 % area to the asymptote (k) column. Sometimes `SummarizeGrowthByPlate()` will give an unrealistically high asymptote reading (k > 1); this indicates that you have not collected enough data for the model to fit properly. To address this, either run a longer experiment, or use the data in the k_corrected column from the `logfit_params_corrected` dataframe. In the k_corrected column, k values > 1 are replaced with the k values from at 24 hr. Another potential issue with the algorithm used to calculate the growth rate (`SummarizeGrowthByPlate()`) is that occasionally high growth rates (r) appear for wells that visually appear to have minimal or no growth. The r_corrected column in the

`logfit_params_corrected` dataframe records r = 0 when k < 0.15 at 24 hr.

Statistical Analysis

The goal of these protocols is to calculate parameters from time-lapse images that correlate with the ability to form biofilms. With those parameters (growth rate, % area at asymptote, time to reach asymptote), standard statistical analyses can be conducted. The specific tests will depend on the experimental framework. For example, if you are testing whether strain background influences biofilm formation in the presence of the same drug (and have measured biological replicates), you could use an ANOVA with strain as the predictor variable and any of the calculated parameters as the response.

Anticipated Results

Culturing and sample preparation

With some exceptions and depending on the strain, cultures should visually appear saturated (visible clumps of cells at the bottom of the tube) in liquid YPD after overnight growth. After 24 hr incubation with drug in the biofilm assay, cells in the control column (no drug; column 1 in Fig. 1) should have a lot of growth visible, while the blank column (column 12 in Fig. 1) should have no growth. Depending on specifics of the strain and the drug environment, you may or may not see growth in experimental wells.

Time-lapse microscopy

C. albicans biofilms are highly structured; they contain yeast and hyphal cells. Generally, at 0 hr, all of the cells should be in the yeast form (Fig. 2). After 1 hr, hyphae should be visible in the control column (column 1). Depending on the drug, concentration, and medium, hyphae might be seen in the drug-treated columns in as little as 1 hr or not at all after 24 hr. If the Evos FL Auto 2 is used with the autofocusing feature, all of the images are expected to be in focus even when cells are highly saturated. For each time point, the images taken will be saved in a separate folder and will be named after the position (i.e., images from well A01 will be named `A01.TIF`).

Orbit Image Analysis

The results of this step should reflect what you see by eye (i.e., if you see few cells in the field of view, Orbit should give a low % area, whereas if it visually appears that cells are covering the whole field of view, Orbit should give you > 90% area). If the results do not reflect what you see, check Critical Parameters for some suggestions on how to troubleshoot this step.

R for renaming, parametrization and visualization

The expected results of each protocol step alongside the code can be seen in the Supporting Information files `BP3-BatchRenaming.pdf`, `BP5-OrbitParameterization.pdf`, and `BP6-LogFitParamsVis.pdf`.

Time Considerations

Culturing

The time required for starting a culture from frozen stock will vary depending on the number of strains and growth conditions. Starting culture straight from frozen stock as described for Basic Protocol 1 for 8 strains will take ∼5-10 min.

Sample preparation for time-lapse microscopy

The time required to set up a 96-well plate will vary depending on the number of strains, the growth environment, and the number of 96-well plates. If the conditions are similar to what is described in Basic Protocol 1, the washing step takes ∼30 min, standardization takes ∼15 min, putting the medium into the 96-well plate and performing the drug dilutions takes ∼10 min, and transferring cultures into the right wells and sealing the plate takes ∼10 min.

Time-lapse microscopy

The time required for setting up time-lapse microscopy will vary depending on the machine. If you use the scanning protocol described in Basic Protocol 2, it takes ∼15 min to set up. Note that the protocol can be set up prior to the actual experimental day. Loading in the protocol on the experiment day takes less than 5 min. The 96-well plate scans can be done in under 5 min. The time that will be spent at the microscope will depend on whether the machine's on-stage incubator is used. If the incubator is used, then after setting up the protocol and starting the experiment, you can just come back the next day to take the plate out. If you manually take the plate out of the incubator, put it on the microscope, scan it, and place it back into the incubator, this requires ∼5 min every 1 hr for 24 hr (or whatever your incubation time is).

Batch file renaming

If you are dealing with 2304 images (as described here), adding a time point suffix to the files using Bulk Rename Utility and combining the images in one folder should take ~20 min. Tweaking the R script and renaming the images through R will take ~30 min the first time, but will take less time (5-10 min) once you are familiar with the script and how to use R.

Orbit training

Training Orbit takes ~30 min the first time you are using the software. This will decrease with program familiarity. The time spent on batch analysis by the program will vary depending on the number of images and the computer. Note that Orbit recommends having a 32 GB RAM (and 64 GB if possible) computer. However, a 5-year-old 8 GB RAM PC was capable of running the analysis and took ~3 hr to analyze 2304 images. Make sure that your computer is plugged in.

R parametrization and visualization

The time spent on this step will depend on your familiarity with R. It is recommended that you read the Supporting Information documents `BP5-OrbitParameterization.pdf` and `BP6-LogFitParamsVis.pdf` for a description of what the different code chunks do and to see the expected output. If this is your first time using R, it could take several hours to understand the functions and steps taken in Basic Protocol 5 and Basic Protocol 6. Once you are familiar with R and have adapted the script (if necessary) to the specifics of your experiment, it should take ~15-20 min the next time you do a similar analysis.

Author Contributions

Ola Salama: conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, software, validation, visualization, writing-original draft, writing-review and editing; **Aleeza Gerstein:** conceptualization, formal analysis, funding acquisition, methodology, project administration, resources, software, supervision, writing review and editing.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request. The scripts and example image files for practice are located in a github repository, accessible at *https://github.com/acgerstein/biofilm-formation*.

Literature Cited

Abdelmegeed, E., & Shaaban, M. I. (2013). Cyclooxygenase inhibitors reduce biofilm formation and yeast-hypha conversion of fluconazole resistant *Candida albicans*. *Journal of Microbiology*, *51*, 598–604. doi: 10.1007/s12275-013-3052-6.

Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O. … Sabatini D. M. (2006). CellProfiler: Image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, *7*, R100. doi: 10.1186/gb-2006-7-10-r100.

Cavalheiro, M., & Teixeira, M. C. (2018). Candida biofilms: Threats, challenges, and promising strategies. *Frontiers of Medicine*, *5*, 28. doi: 10.3389/fmed.2018.00028.

Chandra, J., Kuhn, D. M., Mukherjee, P. K., Hoyer, L. L., McCormick, T., & Ghannoum, M. A. (2001). Biofilm formation by the fungal pathogen *Candida albicans*: Development, architecture, and drug resistance. *Journal of Bacteriology*, *183*, 5385–5394. doi: 10.1128/JB.183.18.5385-5394.2001.

Collins, J. L., van Knippenberg, B., Ding, K., & Kofman, A. V. (2019). Time-lapse microscopy. In R. A. Mehanna (Ed.), *Cell Culture*. Rijeka, Croatia: IntechOpen.

Douglas, L. J. (2003). *Candida* biofilms and their role in infection. *Trends in Microbiology*, *11*, 30–36. doi: 10.1016/S0966-842X(02)00002-1.

Elving, G. J., van der Mei, H. C., Busscher, H. J., van Weissenbruch, R., & Albers, F. W. J. (2002). Comparison of the microbial composition of voice prosthesis biofilms from patients requiring frequent versus infrequent replacement. *The Annals of Otology, Rhinology, and Laryngology*, *111*, 200–203. doi: 10.1177/000348940211100302.

Fanning, S., & Mitchell, A. P. (2012). Fungal biofilms. *PLoS Pathogens*, *8*, e1002585. doi: 10.1371/journal.ppat.1002585.

Fox, E. P., Bui, C. K., Nett, J. E., Hartooni, N., Mui, M. C., Andes, D. R. … Johnson, A. D. (2015). An expanded regulatory network temporally controls *Candida albicans* biofilm formation. *Molecular Microbiology*, *96*, 1226–1239. doi: 10.1111/mmi.13002.

Gulati, M., Lohse, M. B., Ennis, C. L., Gonzalez, R. E., Perry, A. M., Bapat, P., … Nobile, C.

J. (2018). *In vitro* culturing and screening of *Candida albicans* biofilms. *Current Protocols in Microbiology*, *50*, e60. doi: 10.1002/cpmc.60.

Hawser, S. P., & Douglas, L. J. (1995). Resistance of *Candida albicans* biofilms to antifungal agents *in vitro*. *Antimicrobial Agents and Chemotherapy*, *39*, 2128–2131. doi: 10.1128/AAC.39.9.2128.

Kaneko, Y., Miyagawa, S., Takeda, O., Hakariya, M., Matsumoto, S., Ohno, H., & Miyazaki, Y. (2013). Real-time microscopic observation of *Candida* biofilm development and effects due to micafungin and fluconazole. *Antimicrobial Agents and Chemotherapy*, *57*, 2226–2230. doi: 10.1128/AAC.02290-12.

Kawai, A., Yamagishi, Y., & Mikamo, H. (2017). Time-lapse tracking of *Candida tropicalis* biofilm formation and the antifungal efficacy of liposomal amphotericin B. *Japanese Journal of Infectious Diseases*, *70*, 559–564. doi: 10.7883/yoken.JJID.2016.574.

Kojic, E. M., & Darouiche, R. O. (2004). Candida infections of medical devices. *Clinical Microbiology Reviews*, *17*, 255–267. doi: 10.1128/CMR.17.2.255-267.2004.

Kucharíková, S., Tournu, H., Lagrou, K., Van Dijck, P., & Bujdáková, H. (2011). Detailed comparison of *Candida albicans* and *Candida glabrata* biofilms under different conditions and their susceptibility to caspofungin and anidulafungin. *Journal of Medical Microbiology*, *60*, 1261–1269. doi: 10.1099/jmm.0.032037-0.

Kuhn, D. M., Chandra, J., Mukherjee, P. K., & Ghannoum, M. A. (2002). Comparison of biofilms formed by *Candida albicans* and *Candida parapsilosis* on bioprosthetic surfaces. *Infection and Immunity*, *70*, 878–888. doi: 10.1128/IAI.70.2.878-888.2002.

Lind, R. (2012). Open source software for image processing and analysis: Picture this with ImageJ. In L. Harland & M. Forster (Eds.) *Open Source Software in Life Science Research* (pp. 131–149). Cambridge, UK: Woodhead Publishing.

Lohse, M. B., Gulati, M., Valle Arevalo, A., Fishburn, A., Johnson, A. D., & Nobile, C. J. (2017). Assessment and optimizations of *Candida albicans in vitro* biofilm assays. *Antimicrobial Agents and Chemotherapy*, *61*, e02749–16. doi: 10.1128/AAC.02749-16.

Mayer, F. L., Wilson, D., & Hube, B. (2013). *Candida albicans* pathogenicity mechanisms. *Virulence*, *4*, 119–128. doi: 10.4161/viru.22913.

McCall, A. D., Pathirana, R. U., Prabhakar, A., Cullen, P. J., & Edgerton, M. (2019). *Candida albicans* biofilm development is governed by cooperative attachment and adhesion maintenance proteins. *Biofilms and Microbiomes*, *5*, 21. doi: 10.1038/s41522-019-0094-5.

Melo, A. S., Bizerra, F. C., Freymüller, E., Arthington-Skaggs, B. A., & Colombo, A. L. (2011). Biofilm production and evaluation of antifungal susceptibility amongst clinical *Candida spp.* isolates, including strains of the *Candida parapsilosis* complex. *Medical Mycology*, *49*, 253–262. doi: 10.3109/13693786.2010.530032.

Nagy, G., Hennig, G. W., Petrenyi, K., Kovacs, L., Pocsi, I., Dombradi, V., & Banfalvi, G. (2014). Time-lapse video microscopy and image analysis of adherence and growth patterns of *Candida albicans* strains. *Applied Microbiology and Biotechnology*, *98*, 5185–5194. doi: 10.1007/s00253-014-5696-5.

Nett, J. E., Cain, M. T., Crawford, K., & Andes, D. R. (2011). Optimizing a *Candida* biofilm microtiter plate model for measurement of antifungal susceptibility by tetrazolium salt assay. *Journal of Clinical Microbiology*, *49*, 1426–1433. doi: 10.1128/JCM.02273-10.

Pierce, C. G., Uppuluri, P., Tummala, S., & Lopez-Ribot, J. L. (2010). A 96 well microtiter plate-based method for monitoring formation and antifungal susceptibility testing of *Candida albicans* biofilms. *Journal of Visualized Experiments*, *44*, e2287.

Potera, C. (1999). Forging a link between biofilms and disease. *Science*, *283*, 1837–1839. doi: 10.1126/science.283.5409.1837.

R Core Team (2020). R: A Language and Environment for Statistical Computing. Available at https://www.R-project.org/

Rodriguez, A., Ehlenberger, D. B., Dickstein, D. L., Hof, P. R., & Wearne, S. L. (2008). Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images. *PLoS One*, *3*, e1997. doi: 10.1371/journal.pone.0001997.

Sherry, L., Kean, R., McKloud, E., O'Donnell, L. E., Metcalfe, R., Jones, B. L., & Ramage, G. (2017). Biofilms formed by isolates from recurrent vulvovaginal candidiasis patients are heterogeneous and insensitive to fluconazole. *Antimicrobial Agents and Chemotherapy*, *61*, e01065–17. doi: 10.1128/AAC.01065-17.

Sprouffske, K. (2020). Simple Metrics to Summarize Growth Curves [R package growthcurver version 0.3.1]. Available at https://CRAN.R-project.org/package=growthcurver.

Stritt, M., Stalder, A. K., & Vezzali, E. (2020). Orbit Image Analysis: An open-source whole slide image analysis tool. *PLoS Computational Biology*, *16*, e1007313. doi: 10.1371/journal.pcbi.1007313.

Weerasekera, M. M., Wijesinghe, G. K., Jayarathna, T. A., Gunasekara, C. P., Fernando, N., Kottegoda, N., & Samaranayake, L. P. (2016). Culture media profoundly affect *Candida albicans* and *Candida tropicalis* growth, adhesion and biofilm development. *Memorias do Instituto Oswaldo Cruz*, *111*, 697–702. doi: 10.1590/0074-02760160294.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R. … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*, 1686. doi: 10.21105/joss.01686.