Microcolony growth assay – Experiments

Naomi Ziv – 9/2015

If you have any questions or encounter any problems, feel free to contact me at nz375@nyu.edu

- This document contains information about the different experiment types, assumptions and usage of scripts and functions.
- For a general protocol for doing microcolony growth experiments, read MicrocolonyDetails.docx.
- For more information about the underlying code, read MicrocolonyScripts.docx
- For information about how to set-up the assay in a new lab, read MicrocolonyAssay.docx

Table of Contents

Experiment types

There are a number of different ways to set-up microcolony growth experiments. The basic experimental type consists of only brightfield images taken at each time-point. There are three additional features that can be used in combination. In addition to brightfield, fluorescent images can be taken at each time-point, this feature allows to compare average fluorescent intensity between microcolonies or over time. Fluorescent images can also be taken only at the end of the experiment with the purpose of differentiating between microcolonies of different genotypes within the same well (also known as FC experiments or the "color" of a microcolony). This feature has been used for normalizing between wells by introducing an identical fluorescent strain into each well. The third feature is the ability for an experiment to have two phases (also known as AT experiments), separated by a treatment for which the plate is removed from the microscope (such as heat-shock). The experiment type is an important parameter (defined as a number, 1-8) that is set in the experimental ID file (under Experiment.Type). The values are defined as:

1) (GR) Only brightfield
2) (GR-fluor) Brightfield + fluorescence at each time point
3) (FC) Main experiment (brightfield) + fluorescence at end
4) (FC-fluor) Main experiment (brightfield + fluorescence) + fluorescence at end
5) (AT) two experiment phases with a treatment in-between, only brightfield
6) (AT-fluor) Same as #5, with brightfield + fluorescence
7) (AT-FC) Combination of #3 and #5
8) (AT-FC-fluor) Combination of #4 and #6

Defaults and assumptions

NewMultipoints.R

This function is the principle function responsible for creating a multipoints.xml file that can be loaded into the microscope software NIS-elements. It can work in a number of different ways controlled by the user input to NewMultipoints.sh. Generally, it will load information about possible well and field sizes (saved in fieldsize.Rfile) and determine the relevant sizes based on the user answers to NewMultipoints.sh (concerning plate, objective and magnifier). Based on these sizes, there is a limit to how many fields can fit in a well. Specifically, the height or width of the field divided by the height or width of the well, rounded down, specifies the maximum number of rows or columns respectively of a grid of fields in a well. To further ensure the well edge is not imaged, well sizes are generally reduced by ~300 um (see Fieldsize.R).

After determining the well and field sizes, the function will create a grid of field coordinates based on the number of fields specified by the user input to NewMultipoints.sh. It will first try to create a symmetric grid of fields

surrounding the well mid-position (e.g., for 8 fields, it will create a 3X3 grid, for 10 it will create a 4X4). If not all fields in the grid are needed, it will randomly sample which fields in the grid are provided. The sampling will occur only once, so each well has the same relative position of fields. If more fields are requested than can fit in the maximum symmetric grid, the function will create the maximum grid that can fit in the well (this may have more rows or columns depending on the relative shapes and sizes of wells and fields). If more fields are requested than can fit in the maximum grid, all fields in the maximum grid will be returned, overriding the number of fields requested.

You can override this entire procedure by supplying a wellmask (a matrix of 1s and 0s, similar to platemasks and saved in the platemasks folder). This will determine directly the size and configuration of the grid of fields within each well. Care should be taken to know the size of the field and well you are using, otherwise fields can be returned between wells or in neighboring wells while the field is still considered part of the original well during naming and later analysis.

The typical focusing procedure involves first creating template.xml, an xml file with one field per well, where the perfect focus offset has been manually adjusted. When using this option, all fields in the well will be given the adjusted value. When appropriate, you can also use a platemask when creating the template (so that manually adjusted values are only given to some wells). In this case, at least two wells (the first and last) in each row have to be specified (typically, wells from four columns are used). Wells that appear in the template, will receive the adjusted value for all fields, others will receive a weighted average dependent on the values in the wells to either side that appeared in the template. In all cases, all fields in a given well receive the same perfect focus offset.

expsync.sh

This script is responsible for the automatic transfer of images between the microscope and processing computers. It is created the first time Intialize.sh is run and rewritten every time Intialize.sh is run afterwards (replacing the name of the experiment folder to be transferred). The expsync.sh script runs every 20 minutes (via the cron-tab), irrespective if an experiment is running. The script contains rsync commands with multiple arguments specified to ensure all attributes of the files are maintained (importantly, time created, which will be used for growth rate analysis). The script avoids running two iterations of itself by first looking for the file expsyncDone.txt, which it removes and later creates at the end of the script. A potential issue occurs if the script was disrupted in a way that stops the rsync command but does not create the expsyncDone.txt file. The script will not resume transfer of images (either in a current or future experiment). If you identify that this has happened, create the file by running (touch ~/expsyncDone.txt) on the microscope computer.

Image analysis

Image analysis is based on identifying the darkest and lightest pixels in the field. It performs top-hat (to enhance center of cells) and bottom-hat (to enhance black rims of cell) filtering each followed by thresholding. Each thresholding is followed by a set of filling, dilation and erosion operations in order to combine the cells of a particular microcolony into one blob. The results of both approaches are combined into the resulting overlay. The relevant parameters defined in the experimentID file are Black.Dilate.Size, White.Dilate.Size, Black.Threshold, White.Threshold and BW.Blur.Size.

Parallel image analysis

Image analysis is parallelized by the following procedure. Every time PI.m (or FC.m) is run it compares the image and overlay folders, producing a list of images that have not been analyzed. This is compared to queued.mat (file in overlays folder), which contains the list of images currently being processed. The functions will take up to 500 images for processing and append this list to queued.mat (resaving the file). After completion of the batch of image analysis, the function will load queued.mat and remove the names of images it has processed and resave the file. PI (or FC) functions can potentially start running every 10 minutes (called by cronGR,sh which runs every 10 minutes via the cron-tab). The total number of iterations running at a given time depends on the number of MATLAB iterations running and the number of processors on the computer. Images can be processed for several experiments at once.

Color thresholds

The Color.Threshold parameters of the experiment ID file are used either for creating overlays for fluorescent images taken at each timepoint or for creating color_overlays for FC-type experiments. For experiments with fluorescent images, overlays are only created if the parameter Use.Color.Threshold is set (otherwise the files created are just small placeholders). The threshold is used by the matlab function GetBlobs_c. This could potentially be used to specify certain sub-cellular structures and combined with additional analysis to analyze the size, shape and number of these structures.

For FC experiments, the thresholds are used by the matlab function GetBlobs_fc, which creates color_overlays. The proportion of a microcolony area that is part of a blob in the corresponding color_overlay determines colors. The Color.Call.Cutoff parameter set in the experiment ID determines the required proportion. You can have multiple fluorescent channels as part of an FC experiment (and set the Color.Threshold independently). The color given is defined as 1+the channel number (and combined for multiple fluorescence). For an FC experiment with two fluorescent channels, microcolonies can be 4 colors (1-not fluorescent, 2- fluorescent in the first channel, 3-flourecent in the second channel and 5-flourecent in both).

Microcolony alignment

Alignment of microcolonies happens between every two subsequent time-points. Colonies are aligned based on their centroid position and bounding boxes. Specifically, blobs are defined as overlapping if the distance between centroids is less than half the width and height of either of the bounding boxes. Merged colonies are identified as multiple colonies (different blobs in one timepoint) aligned to the same blob in the next timepoint.

Growth rate calculation

There are currently two ways to calculate growth rates (set by the Curve.Fit.Method parameter). Both methods calculate growth rate as the slope of a linear regression of ln(area) over time. The first method uses all available data points for each microcolony. It incorporates lag duration by setting a cutoff of log growth between subsequent points (Gr.Lag.Cutoff), under which a microcolony is considered in lag phase. Lag time-points are not included in the growth rate calculation, and lag values are discreet. The second method uses a sliding window approach, fitting multiple regressions. The regression with the highest slope is chosen to represent the exponential phase of growth. Lag duration is calculated as the intersection of the line defining the growth phase with a horizontal line defined by the initial cell size. Lag duration using this method takes on continuous values.

In the last column of the gr$colparam variable, the first method reports 'points', the number of points used for the growth rate calculation. The second method reports 't-last', the last time-point the colony was tracked (as the number of points used for the growth rate calculation is constant, depending on the size of the sliding window).

AT experiments

AT experiments are experiments with two phases separated by a treatment for which the plate is removed from the microscope. The matlab function GetATmovement calculates the potential shift in plate position after the treatment. An inflexibility in the code results in a requirement for at least 166 fields in the experiment. The d variable should contain extra elements (called phase1 and phase2) that contain the number of time-points in each experiment phase. Growth rates are calculated for the first experiment phase (gr.Rfile), the second experiment phase (grpost.Rfile) or both phases (grall.Rfile). Only the first growth rate calculation method (see above) is implemented for AT experiments.

Problems and solutions

Forgetting to move multipoints file

Problem: The multipoints.xml file used for the experiment was not copied to experiment folder on the microscope computer (this is done manually when the experiment starts).

Identification: Overlays and matout files were created but .Rfile files and certain datafiles were not. Log files contain words such as: "NO MULTIPOINTS FILE IN INPUT OR OUTPUT FOLDER, YOU WILL HAVE TO FIX THAT".

Solution:

1) Copy multipoints.xml into the output folder, you can either:
    a. Find the original file on the microscope computer if it has not been overwritten.
    b. Create a new file with the same number of points (and platemask if used), making a note that the field coordinates and PFS values may differ from the experimental values.
2) Open terminal on the processing computer and type GRpost.sh followed by the experiment name (e.g., GRpost.sh NZ_150908_1_GR).

Bad image analysis

Problem: Image analysis parameters are not appropriate for images taken.

Identification: When looking at movies, traces (colors indicating tracked microcolonies) under- or over- estimate the size of the cell/microcolony.

Solution:

1) Find optimal image analysis parameters by using TestParameters.m.
2) Modify your experiment ID in the experimentsIDs folder. The relevant plateID should also be in the plateIDs folder.
3) Delete or rename your entire output folder.
1) Open terminal on the processing computer and run Intialize.sh (this can also be done on the microscope computer but should be avoided if there is currently another experiment running). Choose the correct experiment and plate ID files.

Bad FC image analysis

Problem: Color threshold parameter used is not appropriate for fluorescent images in FC type experiments.

Identification: color_overlays do not mark fluorescent microcolonies correctly when comparing to the original FC images. When analyzing data, expectations are not met concerning the color of microcolonies (e.g., all colonies receive one color).

Solution:

2) Assuming there is no problem with regular image analysis (see 'Bad image analysis')
3) Delete entire color_overlays and matout folders in your output folder.
4) Modify your experiment ID in the experimentsIDs folder. The relevant plateID should also be in the plateIDs folder.
5) Open terminal on the processing computer and run Intialize.sh (this can also be done on the microscope computer but should be avoided if there is currently another experiment running). Choose the correct experiment and plate ID files.

Some images were not processed

Problem: The most likely reason is a power shutdown on the processing computer during image analysis phase. Also may be necessary in the event that the processing computer has run out of space and after you clear up some space, image processing does not resume.

Identification: PI.txt is still in output folder despite sufficient analysis time, no post-processing has occurred (no matout, datafiles, Rfiles or movies). Images exist in the input folder but no corresponding overlays have been created (this may occur for GR, FC or AT images).

Solution:

1) Delete queued.mat in the overlays folder (or color_overlays / at_overlays if relevant).
2) Wait, processing of missed images (leading eventually to post-processing phases) should resume within ten minutes.

Some images were not transferred

Problem: Likely a result of limited space on processing computer or disruption to microscope computer while running expsync.sh.

Identification: Analysis completes with fewer timepoints than expected or stalls at GRpost.m (meaning there are no Rfiles or some datafiles) with errors concerning unequal number of columns in the log file . Images exist on the microscope computer but do not exist in the input folder on the processing computer.

Solution:

1) Ensure there is available space on the processing computer
2) Determine if images will continue to transfer automatically:
    a. If no other experiments have been started, expsync.sh should still contain the rsync command into the correct folder, verify this by running (less expsync.sh) on the microscope computer.

   b. If no iteration of expsync.sh (no files are being transferred) is running but the home directory on the microscope computer does not have the file expsyncDone.txt (can be checked by running the ls command on the microscope computer), create this file by running (touch ~/expsyncDone.txt) on the microscope computer.
   c. If another experiment has already started or is about to start, it is better that expsync.sh transfer the new experiment, in this case run the rsync command directly on the microscope computer (e.g., rsync -vrltoDW /cygdrive/f/GrowthAssay/NZ_150908_1_GR bermanlab@britanya420-9.tau.ac.i:/Volumes/IN/GR)
3) Delete datafiles/imagetimes.txt (or at_imagetimes.txt) in the output folder
4) Delete queued.mat and queueempty.txt in overlays folder (for experiment phases that had missing images)
5) Delete any matout files that will not be complete (fields had missing images).
6) Copy/create PI.txt file in output folder.

Some images were not taken

Problem: A result of limited space on microscope computer or disruption to microscope computer during the experiment (like a power shut-down).

Identification: Analysis completes with fewer timepoints than expected or stalls at GRpost.m (meaning there are no Rfiles or some datafiles) with errors concerning unequal number of columns in the log file. Images for entire or partial timepoints are missing on the microscope computer and in the input folder on the processing computer.

Solution:
1) In the case that entire timepoints are missing, the analysis should have completed with the available timepoints and as images for the missing timepoints do not exist, there is nothing further that you can do.
2) In the case that there is a partial timepoint (this will result in the stalled analysis described).
   a. Delete the partial timepoint images from the microscope computer and the input folder on the processing computer.
   b. Delete the partial timepoint overlays, jpegs and rprops from the overlays, jpgGRimages and rprops folders (or corresponding AT files).
   c. Delete datafiles/imagetimes.txt (or at_imagetimes.txt) in the output folder.
   d. Delete any matout files that have been created.
   e. Open terminal on the processing computer and type GRpost.sh followed by the experiment name (e.g., GRpost.sh NZ_150908_1_GR).

Bad growth rate calculation

Problem: Growth rate analysis parameters are not appropriate for experiment.

Identification: When looking at plots or doing personal analysis, it is noticeable that parameters used during the automatic analysis were not appropriate (e.g., high GR.Minimum.Timepoints relative to the number of tracked timepoints results in many microcolonies without calculated growth rates).

Solution:

1) If you are interested in replacing the .Rfiles
    a. Modify the experimentID.csv file, it is better to do this in terminal using an editor such as emacs by running (emacs experimentID.csv) after changing the directory to the output folder, for example by running (cd ~/X/out/GR/NZ_150908_1_GR). After changing the parameters, save the file by hitting control-x followed by control-c and answering y to the prompt.
    b. Open terminal on the processing computer and type GRpost.sh followed by the experiment name (e.g., GRpost.sh NZ_150908_1_GR).
    c. The .Rfiles will be replaced in the output folder after not very long.
2) If you are interested in testing and comparing multiple different combinations of parameters, it may be worth to create the gr variable yourself, in R using the getrates2 or getrates4 functions. Look at Rsetup.R for usage.