Microcolony growth assay – overview of scripts and functions

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 short descriptions of all scripts, functions and files associated with the microcolony growth assay. It also contains a flowchart of the automatic experimental processing.
- For a general protocol for doing microcolony growth experiments, read MicrocolonyDetails.docx.
- For more information about the different experiment types, assumptions and usage of scripts and functions, read MicrocolonyExperiments.docx
- For information about how to set-up the assay in a new lab, read MicrocolonyAssay.docx

Table of Contents

The code folder contains the following folders:

unix: all unix scripts

matlab: all matlab scripts

R: all R scripts

microscope: copies of scripts found on the microscope computer and associated files

randomization: files associated with randomization, including copies of scripts found on the robot computer (if relevant)

multipoints: folder that handles files associated with creating xml files containing xy microscope coordinates for fields

Flowchart of scripts and functions:

1) Used during initial set-up or when adding new microscopes or types of plates:

- Fieldsize.R

- Midpositions.R
  - xmlinput.R

2) Used on experiment day:

2a) Used for focusing routine - creating multipoints.xml file:

- NewMultipoints.sh (microscope)
  - AutoMultipoints.sh
    - AutoMultipoints.R
      - NewMultipoints.R
        - xmlinput.R
        - OrderWells.R
    - XMLinput.sh
      - input.c
      - input.exe

2b) Used for experimental analysis:

- Initialize.sh (microscope)
  - SetupIDfiles.sh
    - PlateIDRand.R
  - expsync.sh (cron – microscope computer)

- cronGR.sh (cron – processing computer)
  - PI.m
    - FindImages.m
    - ImportExperimentID.m
    - GetBlobs_bfc.m
    - GetBlobs_c.m (optional - depends on experiment type)
  - FC.m (optional - depends on experiment type)
    - FindImages.m
    - ImportExperimentID.m
    - GetBlobs_fc.m
  - GRpost.sh
    - CheckEmptyQueue.m
    - GRp.sh OR ATp.sh (by experiment type)

```
->GRp.sh
    ImageTimes.sh
    PlateData.R
    RS.m
        ImportExperimentID.m
        imdata.m
        Get_Colors.m
            imdata.m
        GO.m
            BBOverlap.m
        ColonyAreas.m
        ColonyAreasSimple.m (if only one time-point)

->ATp.sh
    ImageTimes.sh (GR input)
    ImageTimes.sh (AT input)
    PlateData.R
    RS_AT.m
        ImportExperimentID.m
        imdata.m
        Get_Colors.m
            imdata.m
        GetATmovement.m
            GetImageMovement.m
        GO_AT.m
            BBOverlap.m
            BBOverlap_AT.m
        ColonyAreas_AT.m
```

Both GRp.sh and ATp.sh run (after RS.m or RS_AT.m):

```
    GRpost.m
    Rsetup.R
        importfunctions.R
        load.GR.R
            importfunctions.R
        get.rates.AT.R
        get.rates2.R
        get.rates4.R
        get.mindist.R
    GRplots.R
    AutoMovies.sh
        Movies.m OR Movies_AT.m (by experiment type)
            ImportExperimentID.m
            Trace.m OR Trace_AT.m (by experiment type)
                ImportExperimentID.m
                text_overlay.m
                    text2im.m
        MoviesFluor.m (optional - depends on experiment type)
            ImportExperimentID.m
```

3) Used after experiment:

3a) For analysis of multiple experiments

get.info.R

3b) For finding appropriate image analysis parameters

TestParameters.m
- GetBlobs_bfc.m
- imoverlay.m

3c) Useful for redoing analysis from various stages

Initialize.sh – can be run from terminal on processing computer

GRpost.sh – can be run directly from terminal on processing computer

Movies.m - can be run directly in MATLAB

3d) Involved in automatic clearing and backing-up of files

clearfiles.sh (cron – microscope computer)

databackup.sh (cron – processing computer)

Explanation of scripts and functions:

/unix

ATp.sh – main script responsible for progression of post processing (after image analysis) for experiment types #5-#8.

AutoMovies.sh – script which calls various matlab Movies functions depending on experiment type.

AutoMultipoints.sh - gets called by NewMultipoints.sh from microscope computer. Converts user information about microscope, plate… into text files (stored temporarily in /Volumes/X/code/multipoints). Runs AutoMultipoints.R and XMLinput.sh

cronGR.sh - CRON script – runs every 10 minutes on processing computer. Main script responsible for progression of analysis. Multiple iterations can run simultaneously each running the image analysis functions PI.m (or FC.m) based on the number of cpu-s and number of iterations of matlab running. Looks for which experiments need analyzing (by the presence of PI.txt in output folder) and runs GRpost.sh on experiments that may be done with image analysis (based on absence of images to process and time since modification of the queued.mat file).

databackup.sh - CRON script – runs every morning at 7AM on processing computer. Responsible for backing-up of output folders and clearing of original image folders and output folders.

GRp.sh – main script responsible for progression of post processing (after image analysis) for experiment types #1-#4.

GRpost.sh – script responsible for progression of experimental analysis after image analysis phase. Run by cronGR.sh for experiment folders that may be done with image processing. It checks queues and runs the appropriate post-processing script (GRp.sh or ATp.sh).

ImageTimes.sh – script which obtains times input images were taken. Used for growth rate calculation.

SetupIDfiles.sh – gets called by Initialize.sh from microscope computer, changes format of plate and experiment ID files and incorporates randomization via PlateIDRand.R.

/matlab

BBOverlap_AT.m – function which creates an overlap matrix which defines the overlap of microcolonies within a field between subsequent time-points. Only

used for the transition between the two experiment phases of AT experiments (experiment types #5-#8). Differs from BBOverlap only in centroid correction based on plate movement.
BBOverlap.m – function which creates an overlap matrix which defines the overlap of microcolonies within a field between subsequent time-points, used for aligning microcolonies over time.

CheckEmptyQueue.m – function run by GRpost.sh, checks if overlay queue is empty and clears if not.

ColonyAreas_AT.m – function which aligns microcolonies over time in a single field and creates matrices of information (areas, centroids…) for experiment types #5-#8. Differs from ColonyAreas only in intial combination of images and information from two experiment phases.

ColonyAreas.m – function which aligns microcolonies over time in a single field and creates matrices of information (areas, centroids…) for experiment types #1-#4.

ColonyAreasSimple.m – function which creates vectors of microcolony information (areas, centroids…) similar to ColonyAreas, for experiments with one time-point.

FC.m – function responsible for image analysis of FC portions of experiments (fluorescence at end), creates jpgs (if input images are not jpgs) and overlays, similar to PI.m with different assumptions. Assumes only fluorescent images, if only one channel uses color 2 threshold, if multiple channels, uses color 2 for c1, color 3 for c2 and color 4 for c3. Processes images in batches of 500.

FindImages.m - function called in PI.m or FC.m, creates queue for image processing.

Get_Colors.m – function which defines the 'color' of microcolonies in an image, creating .col files in colors folder. Colors are defined based on fluorescent images taken at the end of the experiment and hence are only relevent for experiments with a FC component (experiment types #3, #4, #7 and #8). For other experiment types all colors are set to 1.

GetATmovement.m – function which calculates movement of plate position between the two experiment phases of AT experiments (experiment types #5-#8) based on 30 fields.

GetBlobs_bfc.m – main image analysis function for brightfield images, called by PI.m, creates overlays. Performs bottom-hat/top-hat filtering and thresholding to obtain which areas of the image are cells (white islands with black rims).

GetBlobs_c.m – image analysis function for fluorescence images that are taken during the experiment (not at the end), called by PI.m, performs thresholding and creates overlays. Has the potential for being used for a new type of analysis (like calling subcellular structures) but has not been used for this purpose.

GetBlobs_fc.m – image analysis function for fluorescence images that are taken at the end of the experiment, called by FC.m, creates overlays. Performs thresholding to be used later for determining the color of each micro-colony.

GetImageMovement.m – function which calculates shift of cell positions between the two experiment phases of AT experiments (experiment types #5-#8) in a single field.

GO_AT.m – function important for aligning microcolonies over time for experiment types #5-#8, creates a structure of overlap matrices, each defining the overlap of microcolonies between subsequent time-points.

GO.m – function important for aligning microcolonies over time for experiment types #1-#4, creates a structure of overlap matrices, each defining the overlap of microcolonies within a field between subsequent time-points.

GRpost.m – function which compiles matout files and plate data files into txt files (areas, times..) containing information for the entire experiment, will be used later by R.

imdata.m – function which parses the names of a series of tiff files, creating vectors of field numbers (xy), timepoints (t) and fluorescent channels (c).

imoverlay.m – function which creates a mask-based image overlay.

ImportExperimentID.m – function called in many scripts, imports parameter values from experimentID.

Movies_AT.m – main function for creating movies for experiment types #5-#8. On the left are original brightfield images and on the right are the images with colors indicating tracked microcolonies.

Movies.m – main function for creating movies for experiment types #1-#4. On the left are original brightfield images and on the right are the images with colors indicating tracked microcolonies.

MoviesFluor.m – function for creating movies for experiments with one phase and flourecent images at each time point (experiment type #2 and #4). Shows brightfield and flourecent images, also shows florecent images with color indicating overlays if flourecent overlays were created.

PI.m – main function responsible for image analysis, creates jpgs (if input images are not jpgs), overlays and rprops. Assumes brightfield images, if there are multiple channels, assumes c1 images are brightfield. Processes images in batches of 500. Will create overlays for fluorescence images only if parameter is set in experimentID file, using specified treshold. Will process both GR and AT images.

RS_AT.m – main function which aligns microcolonies over time for experiment types #5-#8. Creates matout files which summarize data for each field. The function will not recreate existing matout files.

RS.m - main function which aligns microcolonies over time for experiment types #1-#4. Creates matout files which summarize data for each field. The function will not recreate existing matout files.

TestParameters.m – script used to test image analysis parameters, you should modify (specifying folder names and image analysis parameters) and run the script directly. It will create a movie with a random set of images alongside the processed images (with all called cells/colonies shown in purple).

text_overlay.m – function which adds text onto an image.

text2im.m – function which converts text to an image.

Trace_AT.m – function which creates a structure containing images with traces (colors indicating tracked microcolonies) for experiment types #5-#8.

Trace.m – function which creates a structure containing images with traces (colors indicating tracked microcolonies) for experiment types #1-#4.

/R

AutoMultipoints.R - gets called by AutoMultipoints.sh. Reads information about microscope, plate.. and uses the NewMultipoints function to create a file containing information for the new multipoints file.

Fieldsize.R - script where well and field dimensions are defined, creates /multipoints/fieldsize.Rfile. Used once at set-up and updated with addition of new types of plates or microscope cameras/objectives.

get.info.R – function that can be used after the automatic analysis in order to compile and organize data from multiple experiments. See MicrocolonyExperiments.docx for information about usage, and defaults.

get.mindist.R – function which caluculates minimum distance between microcolonies and creates the ‘md’ variable of md.Rfile.

get.rates.AT.R – contains functions which calculate growth rates for AT experiments. Creates ‘gr’ and ‘grpost’ variables of the gr.Rfile and grpost.Rfile.

get.rates2.R – function which calculates growth rates when the parameter ‘curve.fit.method’ in the experimentID file is set to 1. Calculates growth rate as the slope of a linear regression of log(areas) on time. Uses all points to fit one regression. Creates the 'gr' variable of gr.Rfile.

get.rates4.R – function which calculates growth rates when the parameter 'curve.fit.method' in the experimentID file is set to 2. Calculates growth rate as the slope of a linear regression of log(areas) on time. Fits multiple regressions in a sliding window approach. Creates the 'gr' variable of gr.Rfile.

GRplots.R – A script which produces a number of default plots for initial assessment of the experiment.

importfunctions.R – script contains two functions for importing ID files (experimentID and plateID) into R.

load.GR.R – function which compiles the 'd' variable of the d.Rfile.

Midpositions.R - function that sets up midpositions file (.Rfile) in /multipoints/. Used once at set-up for each combination of plate and microscope, takes as input an xml file with 16 points defining coordinates of edges of plate corners – look at script for details.

NewMultipoints.R – main function for creating information for a new multipoints file. See MicrocolonyExperiments.docx for information about usage, and defaults.

OrderWells.R – function that finds shortest route between used wells/fields, used by NewMultipoints.R.

PlateData.R – script which extracts data from multipoints.xml and saves it in three txt files.

PlateIDRand.R - script which rewrites the plateID.csv file in the output folder to incorporate randomization

Rsetup.R – main script which calculates growth rates and compiles data from the expeiment into three lists (d, gr and md). Each variable is saved as a separate .Rfile.

xmlinput.R – function that parses xml files, used by Midpositions.R and NewMultipoints.R

/microscope

These files are copies or examples of files found on the microscope computer

clearfiles.sh - CRON script – runs every morning at 7AM on microscope computer. Removes experiment folders (containing original images) older than a prespecifed amout of time.

corners.1.24.xml & corners.1.96.xml & corners.1.384.xml – xml files containing 16 points, four points for each corner well of the plate, defining edges. Files were used at set-up as input to the Midpositions R function to create midpositions files.

expsync.sh - CRON script – runs every 20 minutes on microscope computer, rewrittin everytime Intialize.sh is run. Uses the unix function 'rsync' to copy images from the microscope to processing computer.

Initialize.sh - USER script – run by user after experiment starts; user supplies experiment and plate IDs. Creates output folder on processing computer which will kick-start analyisis via cronGR.sh, starts image transfer by creating and running expsync.sh.

NewMultipoints.sh - USER script – run by user to obtain multipoints file. User gives information regarding microscope, plate…The script runs AutoMultipoints.sh on main processing computer.

/randomization

/multipoints

fieldsize.Rfile – file that can be loaded into R, contains two variables (wellsize and fieldsize) which contain information about possible well and field sizes. Created at set-up by modifing and running Fieldsize.R.

input.c & input.exe – called by XMLinput.sh, c script, converts multipoints to xml file.

midpositions.1.24.Rfile & midpositions.1.96.Rfile & midpositions.1.384.Rfile – files containing x and y corrdinate for well midpositions, specific for microscope and plate. Created at set-up by using Midpositions R function.

multipoints.xml & template.xml – current multipoints and template xml files, rewrittin everytime NewMultipoints.sh is run on microscope computer (every experiment).

XMLinput.sh – script where information for a new multipoints file (saved as a txt file) is converted to an xml file (using input.exe) that can be read by the microscope software.