Colony growth tracking

The purpose of the applications described below is first to prepare the raw images for convenient image analysis, then to analyze each Petri dish and finally to reconstruct the statistics of appearance time and growth rates of all dishes that belong to the same strain/condition.

Pre-Processing

Each scanner produces an image of up to 6 Petri dishes. Each Petri dish is analyzed separately (because each dish may contain a different condition). The preprocessing stage purpose is to align all the images of a specified scanner, and to separate the 6 Petri dishes. It is assumed that the raw images are similar to the example shown below, namely that each scanner monitors up to 6 standard Petri dishes that are placed in a fixed position on the scanner.
All functions are in subfolder 'Prepare'.

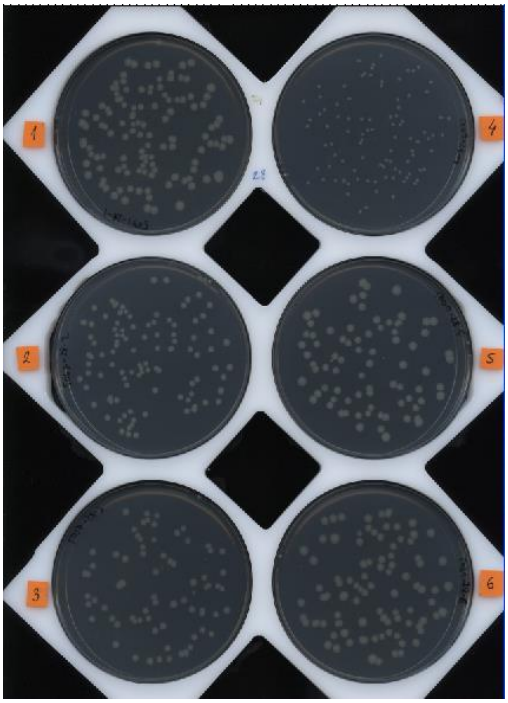


Figure1 - Typical raw image of 6 standard Petri dishes on one of the scanners. The 6 plates are placed inside a "Scanning board" – a rigid board with 6 round slots that keeps the plates in a fixed position on the scanner.

CropROI(SourceName,DestDirNames,BoardFileName,Plates2Cut,isSpecific,changeAlign)		
Purpose	Cut the image into 6 separate Petri dishes	
Parameters	`SourceName`	The scanner's images in a general format including the path directory and the format. (<FilesPath\GeneralFileName*.FileType>). **Example**: Suppose "A" is the folder where all images are, and it contains images of several scanners. If scanner 1 images are called `im_S1_*.tif` and scanner 2 images are called `im_S2_*.tif` etc. then to crop images of scanner 2 write: `'D:\Scans\A\im_S2_*.tif'`
	`DestDirNames`	A vertical cell array of destination folders for plate images. **Example**: `{'D:\Scans\A\A_S2_P1';'D:\Scans\A\A_S2_P3'; 'D:\Scans\A\A_S2_P4'}`
	`BoardFileName`	The board hint file name (<FilePath\FileName>). This file is created once for each plate holder using `CreateBoardHint`. **Example**: `'C:\Boards\BoardHint2.mat'`
	`Plates2Cut`	An array of the wanted Petri dishes **Example**: [1;3;4]
	`[isSpecific]`	Aligns a plate individually (default 0).
	`[changeAlign]`	Let you choose the alignment rectangle (default 0).
Description	The function aligns the images of a specific scanner specified in `SourceName`, finds the Petri dishes in the image using the `BoardFileName`, and cuts each Petri dish. The images of each separate Petri dish are placed in `DestDirNames`.	
Output	• Subfolders for each plate as specified in `DestDirNames` that contain the separated plates and a data file. • A file containing the movement corrections for the scanner (`'D:\Scans\A\A_S2_<firstimage>_motions'`).	

CreateBoardHint(inputImage,NumberOfPlates,FileName,PlateDiameter)		
Purpose	Defining a new "Scanning board". A "Scanning board" is a rigid white board with 6 round slots for the Petri dishes, as shown in the figure 1.	
Parameters	`inputImage`	An image or a name of and image of the plate holder (Figure 1) **Example**: `'D:\Scans\A\im_S2_20170425_1033.tif'`
	`NumberOfPlates`	Number of holes for Petri dishes to be found **Example**: `6`
	`FileName`	The board hint file name that will be created (<FilePath\FileName>). This file is created once for each plate holder and used by `CropROI`. **Example**: `'C:\Boards\BoardHint2.mat'`
	`[PlateDiameter]`	The diameter [mm] of the Petri dish. (default: 90) **Example**: 90
Description	The function finds `NumberOfPlates` circles and saves their center and radius in `FileName`. It assumes bright board and dark plates (Figure1). This is an iterative and relatively long process, and it is done once for each new board.	
Output	• The board hint file named `FileName`.	

Analyzing the images

The purpose of this stage is to identify colonies in the images of the dishes, and to track them in time. This is done for each Petri dish separately using `ProcessPlates`. All functions are in subfolder 'Track Colonies'.

`setMaskApp(DirName)`		
Purpose	Defining the area of analysis inside a plate	
Parameters	`DirName`	A directory of plate files **Example**: `'D:\Scans\A\A_S2_P1'`
Description	Shows the last plate and the area to analyze. The user can interactively change this area.	

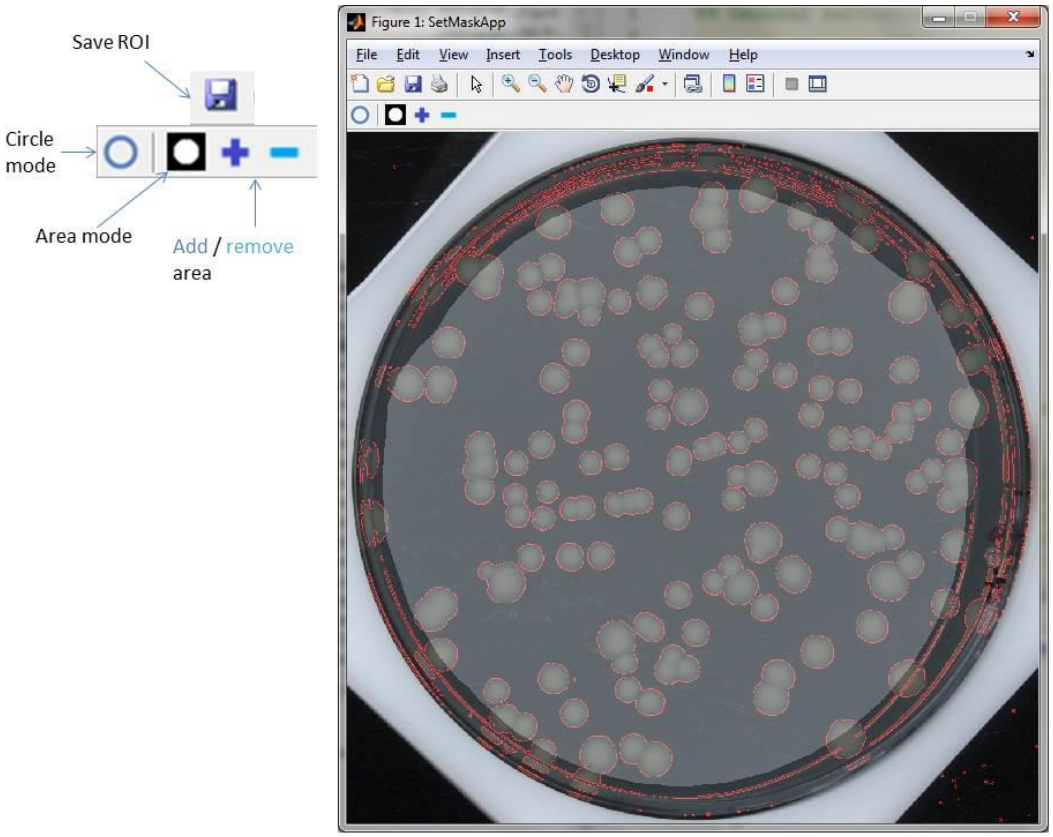


Figure 2 – SetMaskApp - Defining the area of analysis inside a plate. You can change the default inner circle area by clicking the area to be included/ excluded using the Add (+) or Remove (-) buttons.

ProcessPlates(SourceDirs,LogFiles,Descriptions)		
Purpose	Finding the colonies appearance time and are in time.	
Parameters	SourceDirs	A vertical cell array of plate folders. **Example**: {'D:\Scans\A\A_S2_P1';'D:\Scans\A\A_S2_P3'; 'D:\Scans\A\A_S2_P4'}
	[LogFiles]	A vertical cell array of log files. The starting time of the experiment is extracted from there. **Example**: {'D:\Scans\A\logFileName.log';'D:\Scans\A\logFileName.log';'D:\Scans\A\logFileName.log'}
	[Descriptions]	A vertical cell array of plate folders **Example**: {'Strain 1';'Strain 1';'Strain 2'}
Description	For each plate in SourceDirs it finds the colonies in time. Entering LogFiles enables to calculate the time since the beginning of the experiment. It can also be done later using SetStartingTime. If you enter Descriptions for each plate, it will later be used as a title when showing the plate or the area graph. It can also be done later using SetDescription.	
Output	• A data file in each plate folder. **Example:** 'D:\Scans\A\A_S2_P1\A_S2_P1_data'	

SetStartingTime(SourceDir,StartTime,LogFile)		
Purpose	Setting the time of experiment begining	
	SourceDir	The plate folder **Example**: 'D:\Scans\A\A_S2_P1'
	StartTime	Write '' (blank) if you want the starting time to be set from the log file. Othereise write the starting time in 'yyyy/mm/dd HH:MM:SS' format **Example**: '2017/11/28 15:10:00'
	LogFile	The log file is written when the first image is taken, and usually the experiment starts then. **Example**: 'D:\Scans\A\logFileName.log'
Description	Changes the starting time of the plate in the data file	

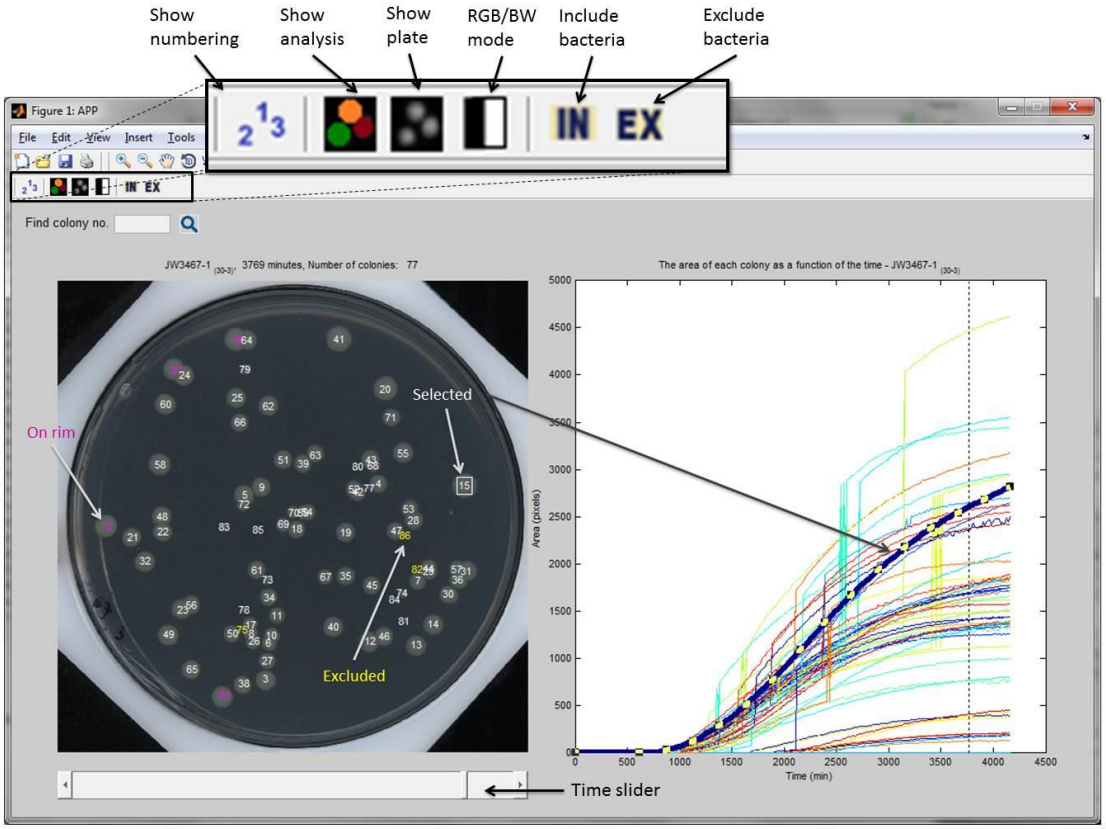
SetDescription(SourceDir,Description)		
Purpose	Sets the Description of the plate	
	SourceDir	The plate folder **Example**: 'D:\Scans\A\A_S2_P1'
	Description	**Example**: 'Strain 1'

Data analysis

The purpose of this stage is to extract the statistics of appearance time and growth rates of all Petri dishes that belong to the same strain/condition. Some functions works on one Petri dish (e.g. `PlateAnalyzer`), while other functions use the data from all Petri dishes with the same condition (e.g. `getAppearanceTime`, `getSurvivalCurve`, `getAppearanceGrowth`, `getStatistics`). All functions are in subfolder 'TrackColonies' or 'ConditionAnalisys'

Major functions list:

`PlateAnalyzer(FileDir,Time)`		
Purpose	A tool for investigation of the automatic analysis	
Parameters	`FileDir`	The root directory of the Petri dish (full path) **Example**: `'D:\Scans\A\A_S2_P1'`
	`[Time]`	if no starting time was found in the data file the time argument will be used (with default 0)
Description	Displaying the Petri dish alongside with the graph of the colony size versus time. Using this tool you can associate a colony with its curve on the area graph, you can locate a colony, and exclude analysis defects (or include). Using the time slider you can scroll through the experiment at different times and learn about the development of the colonies. When show analysis in on, the colonies are painted with their colour, similar to the colour of their curves. The RGB/BW mode switches between the original plate image and the black and white image after background subtraction. Colour coding of the text IDs of the colonies: The ID number is red for new identified colonies, blue for something identified (dirt) that doesn't appear in the next frame, yellow for manually screened colonies, and purple for colonies that are on the rim of the analyzable area of the plate.	



[AppearanceTime] = getAppearanceTime(SourceDirs,BeginTimes)		
Purpose	Gets the appearance times pf all colonies	
Parameters	`SourceDirs`	A vertical cell array of plate folders. **Example**: `{'D:\Scans\A\A_S2_P1';'D:\Scans\A\A_S2_P3'}`
	`[BeginTimes]`	if no starting time was found in the data file the `BeginTimes` argument will be used (with default 0)
Returns	`AppearanceTime`	A structure array of appearance times (in minutes) of all the colonies (`AppearanceTime.time`). This array also gives the unique ID (`AppearanceTime.plate + AppearanceTime.id`) you can later identify a specific colony.
Description	The function returns a structure array of all colonies in all plates specified. It contains an ID of the colony (the plate and the colony number), and the time it appeared. Use histogram on `AppearanceTime.time` to show the appearance distribution.	

[SurvivalCurveX, SurvivalCurveY] = getSurvivalCurve(SourceDirs,BeginTimes)		
Purpose	Gets a survival curve	
Parameters	`SourceDirs`	A vertical cell array of plate folders. **Example**: `{'D:\Scans\A\A_S2_P1';'D:\Scans\A\A_S2_P3'}`
	`[BeginTimes]`	if no starting time was found in the data file the `BeginTimes` argument will be used (with default 0)
Returns	`SurvivalCurveX`	Appearance times
	`SurvivalCurveY`	Cumulative appearances
Description	The function returns how many colonies have not yet appeared till that time. **Example**: `plot(SurvivalCurveX,SurvivalCurveY/TotalN)`	

[AppearanceGrowth,NotBigEnough,Merged] = getAppearanceGrowth(SourceDirs, lb, ub,BeginTimes)		
Purpose	This function calculates the growth rate and appearance time for each relevant colony	
Parameters	`SourceDirs`	A vertical cell array of plate folders. **Example**: `{'D:\Scans\A\A_S2_P1';'D:\Scans\A\A_S2_P3'}`
	`lb`	lower bound size in pixels
	`ub`	upper bound size in pixels
	`[BeginTimes]`	if no starting time was found in the data file the `BeginTimes` argument will be used (with default 0)
Returns	`AppearanceGrowth`	A structure of arrays with a unique ID of a colony (plate+colony number) and the time it took to reach from `lb` to `ub`.
	`NotBigEnough`	colonies with last area < ub (not in statistics)
	`Merged`	colonies that merged before exceeding the ub (not in statistics)
Description	This function calculates the growth time fo reach relevant colony. Relevant colonies are the colonies that are not excluded by the user, and are far enough from the border. If several colonies merged into one, we take data only till the first merging time. **Example**: `AppearanceGrowth.growth*60*24` gives the growth time in minutes.	

[Stat] = getDistrStatistics(Distr)		
Purpose	checking statistical parameters for a histogram	
Parameters	`Distr`	appearance time for each bacteria (not a histogram).
Returns	`Stat`	A structure of statistical values: total, Avg, std, skewness, mode, median, stdMedian.