How to use ScanGrow [PoC]

This manual is intended to provide guidance for the utilisation of the application ScanGrow Proof of Concept **version, developed by Riverwell Consultancy Services Ltd.** For help or feedback, please contact Laura Espina (laura.espina@zoo.ox.ac.uk or lespina@gmail.com) or Ross Worth (ross@bytepro.co.uk).

Installation and general use

The installer “ScanGrow Installer.msi” installs the standalone application ScanGrow [version proof of concept] on a standard PC running on Windows 7 or later. For optimal use, install and run ScanGrow in Administrator mode.

With the present version of ScanGrow, only one flat-bottomed 96-wells microplate can be analysed at a time for each scanner. If more than one scanner is connected to the PC, a separate instance of the application can use a different scanner to process more than one microplate at the same time. Each microplate needs to be positioned on each scanner as explained below.

Software requirements

- Standard PC running on Windows 7 or later with .NET Framework installed.
- For customization of the classification model, Windows Visual Studio containing the ML.NET Model Builder component is needed.
- For other customization options and to collect data from the TableView of the ScanGrow app, Microsoft Excel or any other spreadsheet manager is necessary.

Position of the microplate on the scanner

Place the microplate in the correct position for obtaining the image (see Figure 1). Using another microplate as a positioner can help. Cover the scanning area with a black cloth or pasteboard. Sources of light other than the scanner beams can introduce noise in the images so they should be minimized.

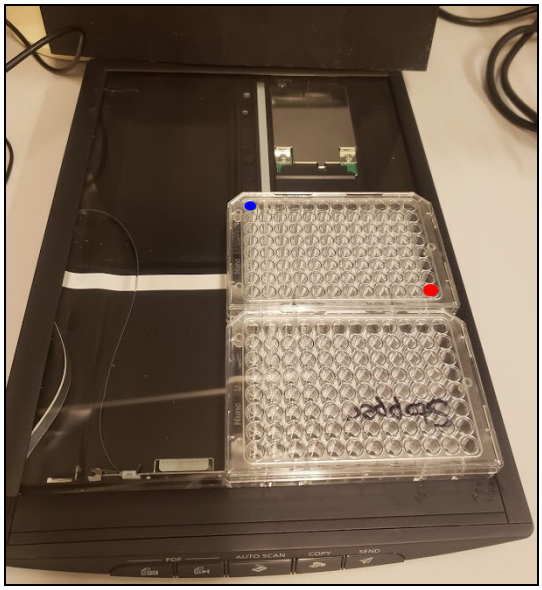

Figure 1. Position of the microplate of interest on the scanner. An additional microplate (“Stopper”) can be placed to achieve the ideal position. Red spot = well A1, blue spot = well H12.

With this disposition, the images collected by the scanner will show the plate on the right side. This type of format will be referred to as “original position”.

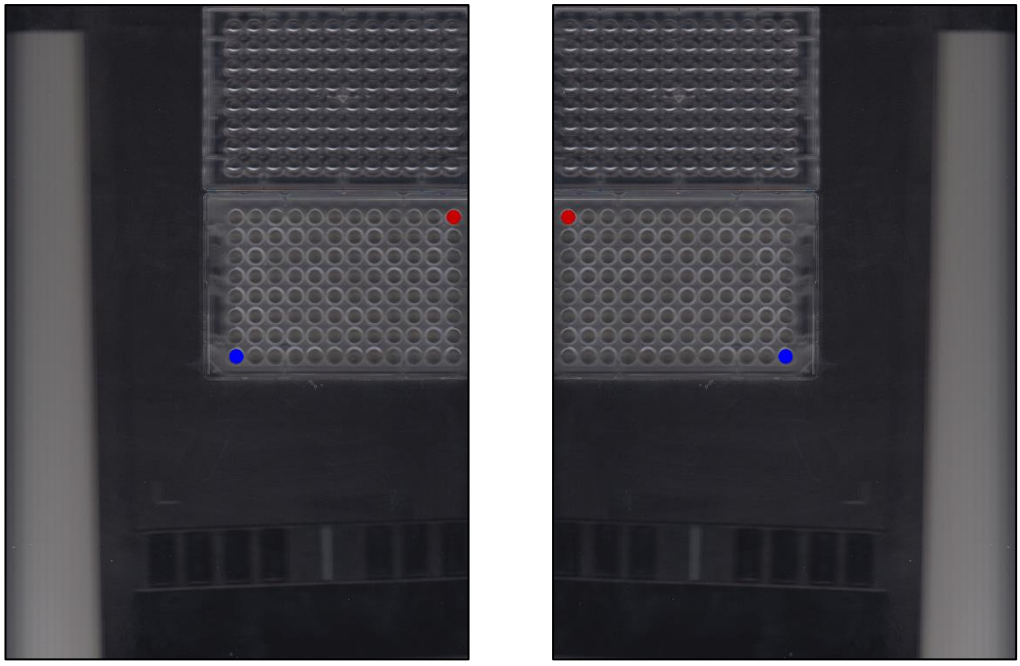

Figure 2. "Original position" (left) and "inverted position" (right). The "inverted position" is the one used for the analyses. Red spot = well A1, blue spot = well H12.

Automatic run mode

Use this mode to program automatic scans throughout a specific incubation time, while being able to see the results in real time during the run. These results consist in growth curves for each well, with the level of growth being automatically assigned to each well at each time point based on the configuration mode. To learn how to change the configuration files and the training model, go to the section “Customization”.

To begin the automatic run, place the scanner containing the microplate at the appropriate temperature and in a dark place.

When opened, the app will recognize the scanners available to use. Select one scanner and the working directory (which can be an empty folder), as well as the interval between scans and the number of scans needed. After the adequate selections, click *Start*.

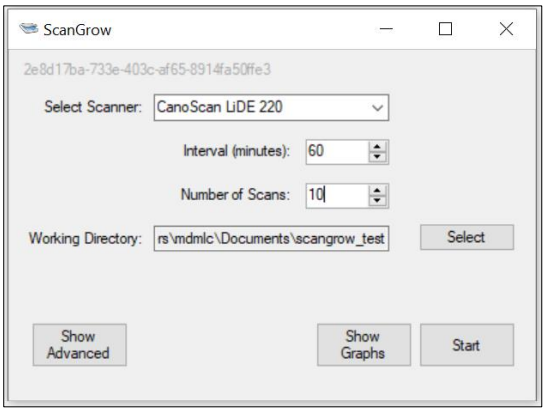

Figure 3.

At any moment during the run, click on *Show Graphs* to see the growth curves as detected by the app. It make take some minutes to read the results and process the graphs. Click on “Refresh” to update the Graph view.

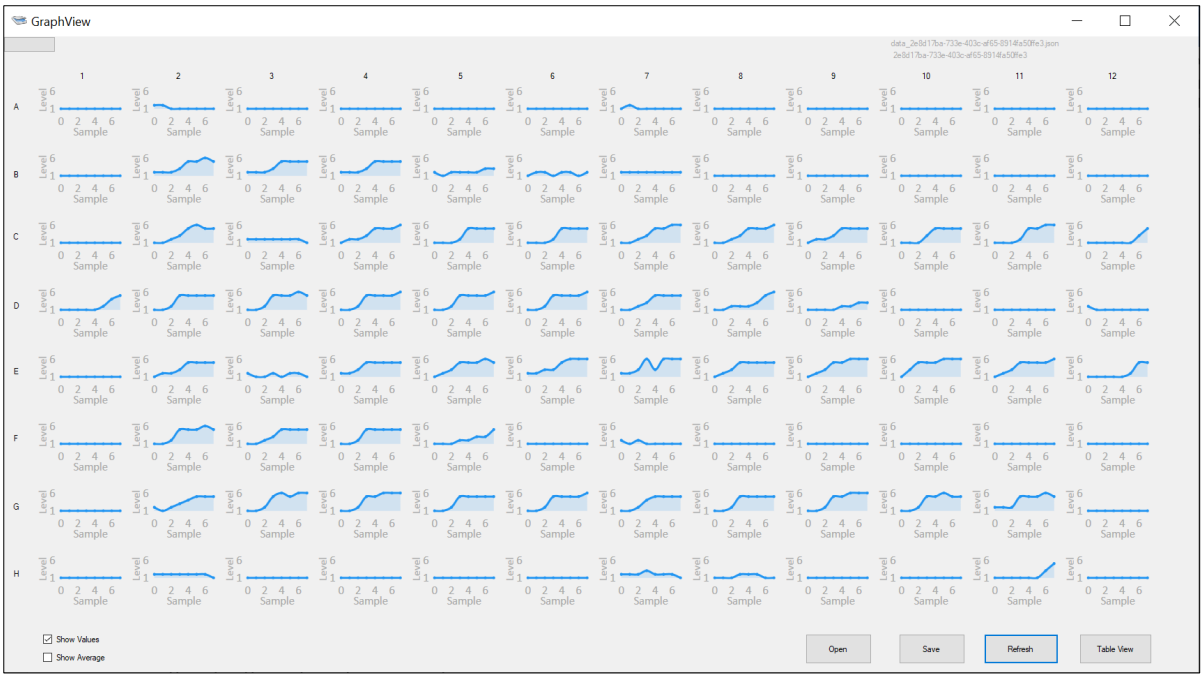

Figure 4.

If the Graph View is not showing the graphs at the correct size, close the app, change the display settings (scaling and/or resolution), and open ScanGrow again.

Results shown on the Graph view can be saved and opened in *json* format using the buttons *Save* and *Open*, respectively.

From the graphs view, click on *Table View* to visualize the data frame containing the levels of growth assigned to each well at each time point. This table can be copied to the clipboard for further analysis of the data.

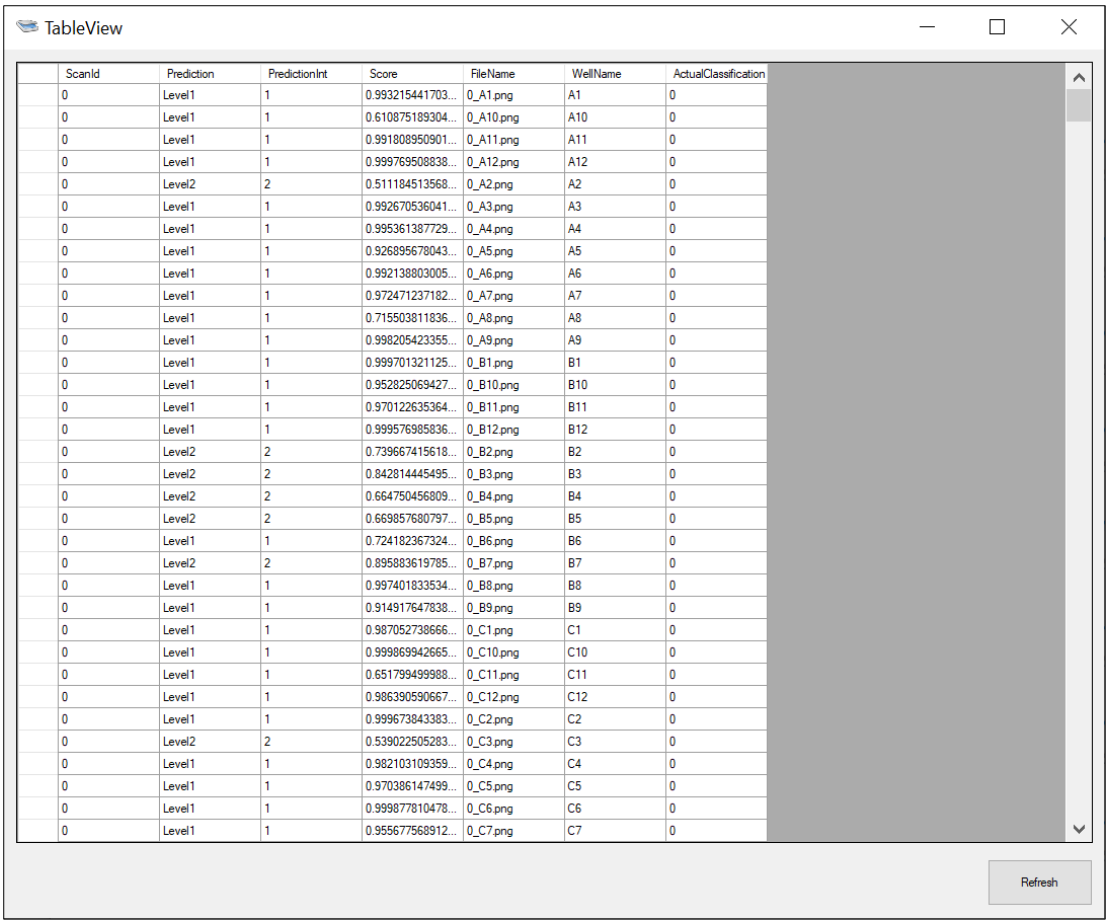
TableView
ScanId
Prediction
PredictionInt
Score
FileName
WellName
ActualClassification
0 Level1 1 0.993215441703... 0_A1.png A1 0
0 Level1 1 0.610875189304... 0_A10.png A10 0
0 Level1 1 0.991808950901... 0_A11.png A11 0
0 Level1 1 0.999769508838... 0_A12.png A12 0
0 Level2 2 0.511184513568... 0_A2.png A2 0
0 Level1 1 0.992670536041... 0_A3.png A3 0
0 Level1 1 0.995361387729... 0_A4.png A4 0
0 Level1 1 0.926895678043... 0_A5.png A5 0
0 Level1 1 0.992138803005... 0_A6.png A6 0
0 Level1 1 0.972471237182... 0_A7.png A7 0
0 Level1 1 0.715503811836... 0_A8.png A8 0
0 Level1 1 0.998205423355... 0_A9.png A9 0
0 Level1 1 0.999701321125... 0_B1.png B1 0
0 Level1 1 0.952825069427... 0_B10.png B10 0
0 Level1 1 0.970122635364... 0_B11.png B11 0
0 Level1 1 0.999576985836... 0_B12.png B12 0
0 Level2 2 0.739667415618... 0_B2.png B2 0
0 Level2 2 0.842814445495... 0_B3.png B3 0
0 Level2 2 0.664750456809... 0_B4.png B4 0
0 Level2 2 0.669857680797... 0_B5.png B5 0
0 Level1 1 0.724182367324... 0_B6.png B6 0
0 Level2 2 0.895883619785... 0_B7.png B7 0
0 Level1 1 0.997401833534... 0_B8.png B8 0
0 Level1 1 0.914917647838... 0_B9.png B9 0
0 Level1 1 0.987052738666... 0_C1.png C1 0
0 Level1 1 0.999869942665... 0_C10.png C10 0
0 Level1 1 0.651799499988... 0_C11.png C11 0
0 Level1 1 0.986390590667... 0_C12.png C12 0
0 Level1 1 0.999673843383... 0_C2.png C2 0
0 Level2 2 0.539022505283... 0_C3.png C3 0
0 Level1 1 0.982103109359... 0_C4.png C4 0
0 Level1 1 0.970386147499... 0_C5.png C5 0
0 Level1 1 0.999877810478... 0_C6.png C6 0
0 Level1 1 0.955677568912... 0_C7.png C7 0
Refresh

Figure 5.

Offline mode

Use this mode to analyse previously obtained scans. The format for the output results is the same than for the automatic run mode.

Input format: The previously obtained *.tif* scans need to be located in the same folder, in "original position" format, and with the titles in numeric ascending order.

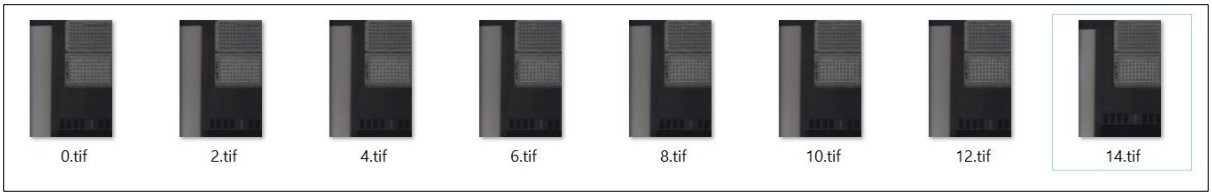

Figure 6. Example of images to be analysed in offline mode.

To run the offline mode, click on *Show Advanced*, and in the *Run offline* section, click on *Select* to select the folder where the images are. Click on *Run offline*. You will be prompted to select the first image to analyse.

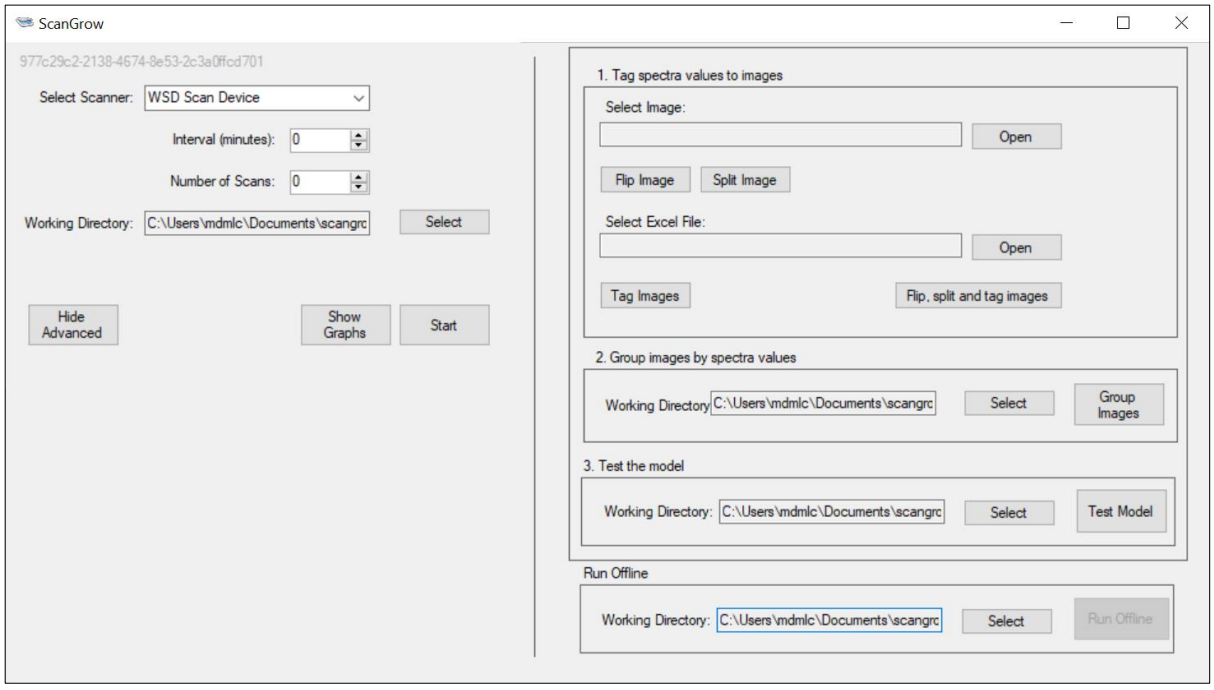

Figure 7. Offline run being processed.

When the analysis of the first image is complete, the button will appear clickable again. Click on it again to continue with the following images. At any moment in the process, click on the button *Show Graphs* to see the growth curves and store the values from the *Table View* mode.

Customization

The “Automatic run mode” and “Offline mode” run under a preconfigured setting which is defined by three customizable files:

- ImageMask.csv
- ClassificationLevels.csv
- MLModel.zip

These three files can be modified to adjust different parameters, explained below.

Customization of the position and size of the wells ("positioning grid")

File and its location:

\Program Files\Riverwell Consultancy Services Ltd\Scan Grow\Configuration\ImageMask.csv

The file “ImageMask.csv” contains the coordinates and the size of the cuts, both being customizable. For this version of ScanGrow, it is better to maintain the number of cuts (96).

Include	Row	Column	Y	X	Width	Height
TRUE	A	1	1138	82	62	62
TRUE	A	2	1138	188	62	62
TRUE	A	3	1138	294	62	62
TRUE	A	4	1138	400	62	62
TRUE	A	5	1138	506	62	62
TRUE	A	6	1138	612	62	62
TRUE	A	7	1138	718	62	62
TRUE	A	8	1138	824	62	62
TRUE	A	9	1138	930	62	62
TRUE	A	10	1138	1037	62	62
TRUE	A	11	1138	1144	62	62
TRUE	A	12	1138	1251	62	62
TRUE	B	1	1246	82	62	62
TRUE	B	2	1246	188	62	62
TRUE	B	3	1246	294	62	62
TRUE	B	4	1246	400	62	62
TRUE	B	5	1246	506	62	62
TRUE	B	6	1246	612	62	62
TRUE	B	7	1246	718	62	62
TRUE	B	8	1246	824	62	62
TRUE	B	9	1246	930	62	62

Figure 8. ImageMask.csv.

Customization of the OD range for each growth level

File and its location:

\Program Files\Riverwell Consultancy Services Ltd\Scan Grow\Configuration\ClassificationLevels.csv

The file “ClassificationLevels.csv” contains the OD ranges that comprise each level of growth (y-axis in the growth curves). The OD values can be changed, but it is recommended to maintain the number of levels (6).

Name	GreaterOrEqual	LessThan
Level1	0	0.06
Level2	0.06	0.15
Level3	0.15	0.25
Level4	0.25	0.35
Level5	0.35	0.6
Level6	0.6	100000

Figure 9. ClassificationLevels.csv.

Customization of the machine learning model

File and its location:
\Program Files\Riverwell Consultancy Services Ltd\Scan Grow\MLModel.zip

A deeper level of customization can be done by creating a new machine learning model. Through this process, a new model is trained by assigning pre-obtained OD (spectra) values to scanned images. The steps to train a new model are:

1. Tag absorbance values to images
2. Group images by absorbance values
3. Create the model with ML.NET Model Builder
4. Test the model

The whole process is explained below using the compressed folder “Trainingfiles.zip”, which contains both the scanned images and the absorbance values in Excel format.

1. Tag absorbance values to images

Input format of the scanned images and the absorbance values: Copy the images and the Excel files in the same folder. For each image (preferably in “inverted position”), an Excel file with the same title should be present.

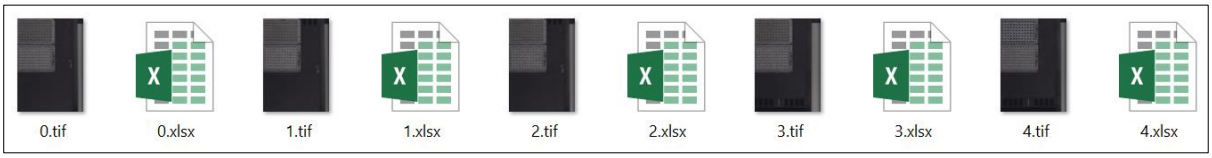

Figure 10. Example of files for training the model.

Do not change the number or names of the tabs in the Excel file, or its format, when introducing new absorbance values: it should look like in Figure 11.

	A	B	C	D
1				
2		**Well**	**Name**	**A620**
3		A1	Sample 1.1	0.033
4		A2	Sample 1.2	0.030
5		A3	Sample 1.3	0.030
6		A4	Sample 1.4	0.029
7		A5	Sample 1.5	0.034
8		A6	Sample 1.6	0.037
9		A7	Sample 1.7	0.077
10		A8	Sample 1.8	0.075
11		A9	Sample 1.9	0.198
12		A10	Sample 1.10	0.207
13		A11	Sample 1.11	0.325
14		A12	Sample 1.12	0.033
15		B1	Sample 1.13	0.029
16		B2	Sample 1.14	0.027
17		B3	Sample 1.15	0.026
18		B4	Sample 1.16	0.029
19		B5	Sample 1.17	0.033
20		B6	Sample 1.18	0.032
21		B7	Sample 1.19	0.071

Header | Results

Figure 11. Example of Excel file for training the model.

In the app, click on *Show advanced*, and in the "*1. Tag spectra values to images"* section, click on the upper *Open* to select the first image. If the image is in "inverted position", click on *Split image* to cut out the 96 wells; if the image is in "original position", click on *Flip Image* first. The position and size of these cuts can be configured by changing the file "ImageMask.csv", as previously explained. After that, select the lower *Open* to select the corresponding Excel file. Click on *Tag images* to tag each absorbance value to the corresponding image of the well.

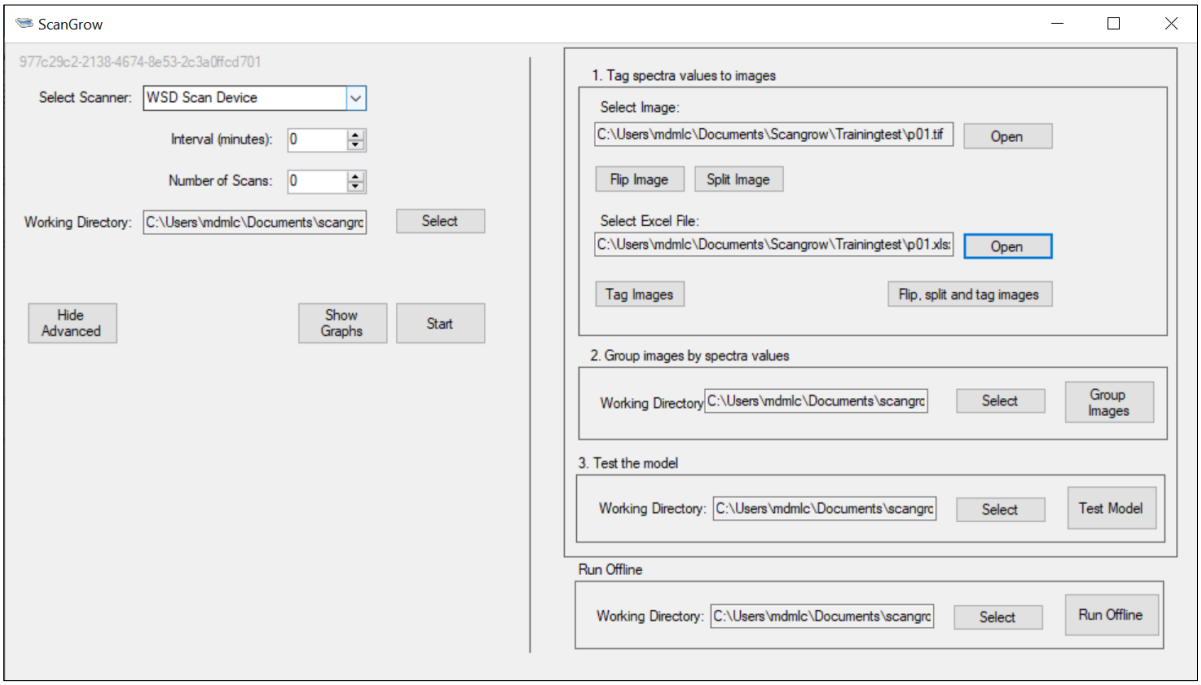

Figure 12.

In the same folder where the training files are located, 96 folders will appear, each of them containing the image of the well of the same name with its absorbance value in the title. This process can be repeated with the rest of the training files (by selecting the next image and Excel file) until each folder contains all the data for its well. The folder and the contents of each of its subfolder should look like in Figure 13.

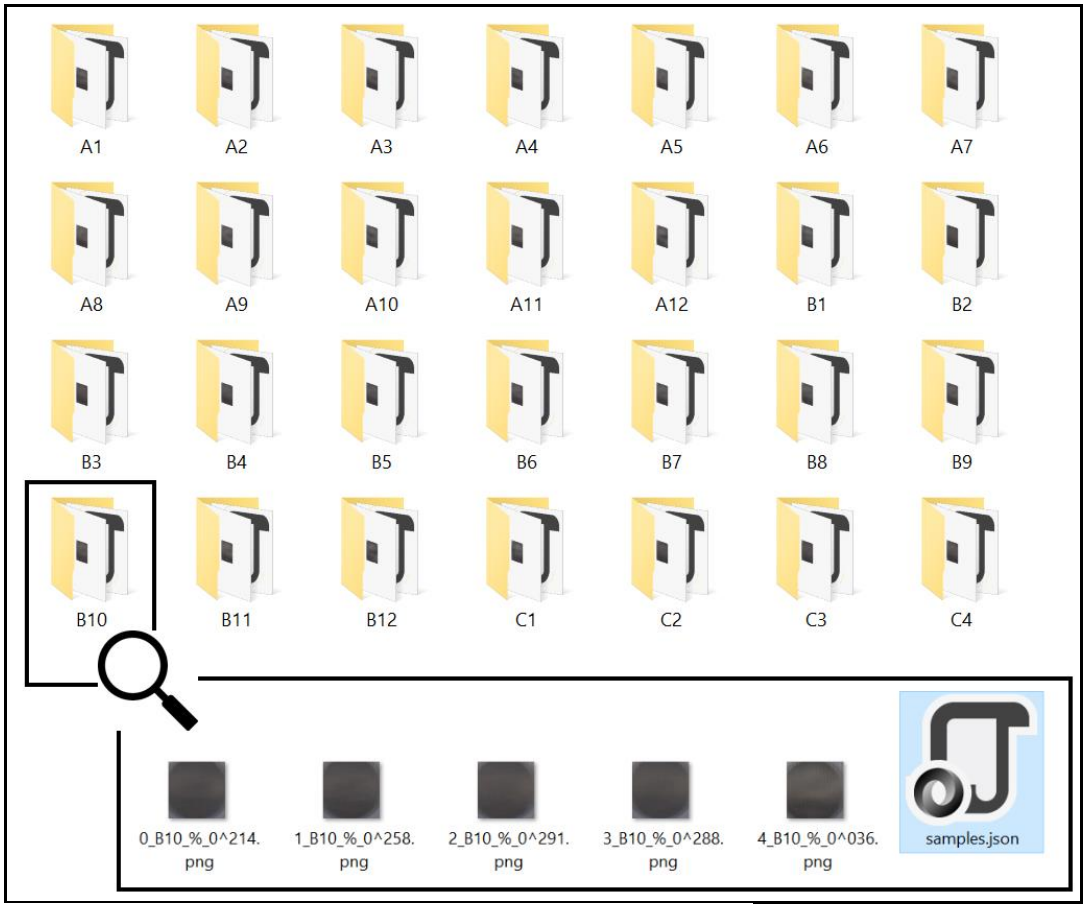

Figure 13.

2. Group images by absorbance values

When all the training images have been tagged, in the app, click on *Show advanced*, and in the "*2. Group images by spectra values"* section, click on *Select* to select the folder containing the A1-H12 folders. Click on *Group Images*. New folders (named Level1, Level2, Level3, Level4, Level5 and Level6) will appear next to the A1-H12 folders, containing the cut wells categorized by level. Move the Level1 – Level6 folders to another empty folder.

3. Create the model with ML.NET Model Builder

You must have installed Microsoft Visual Studio 2019 v.16.6.1 or later, and the ML.NET Model Builder component. In Visual Studio, create a new C# Console App (.NET core) project. Right-click on the title of the project and add Machine Learning.

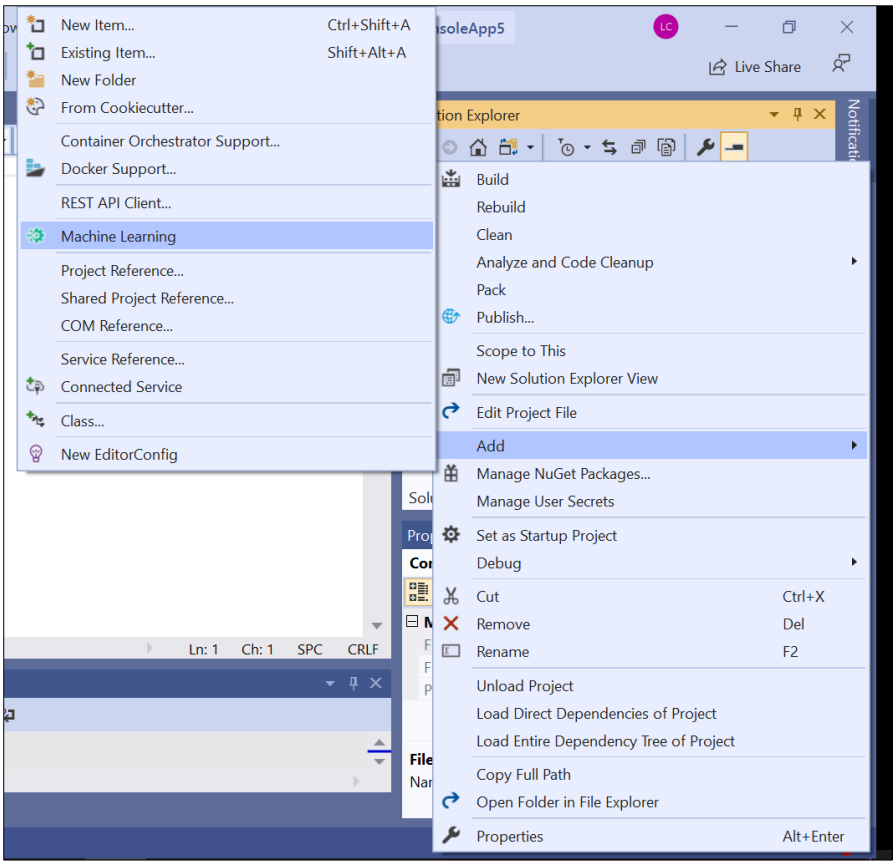

Figure 14.

Click on the “Image Classification” button. Select the folder where the Level1 – Level6 folders are located. Follow the sequencial steps up to the step “5.Code”. Once the model is created, right-click on the element “MLModel.zip” and “Open containing folder”.

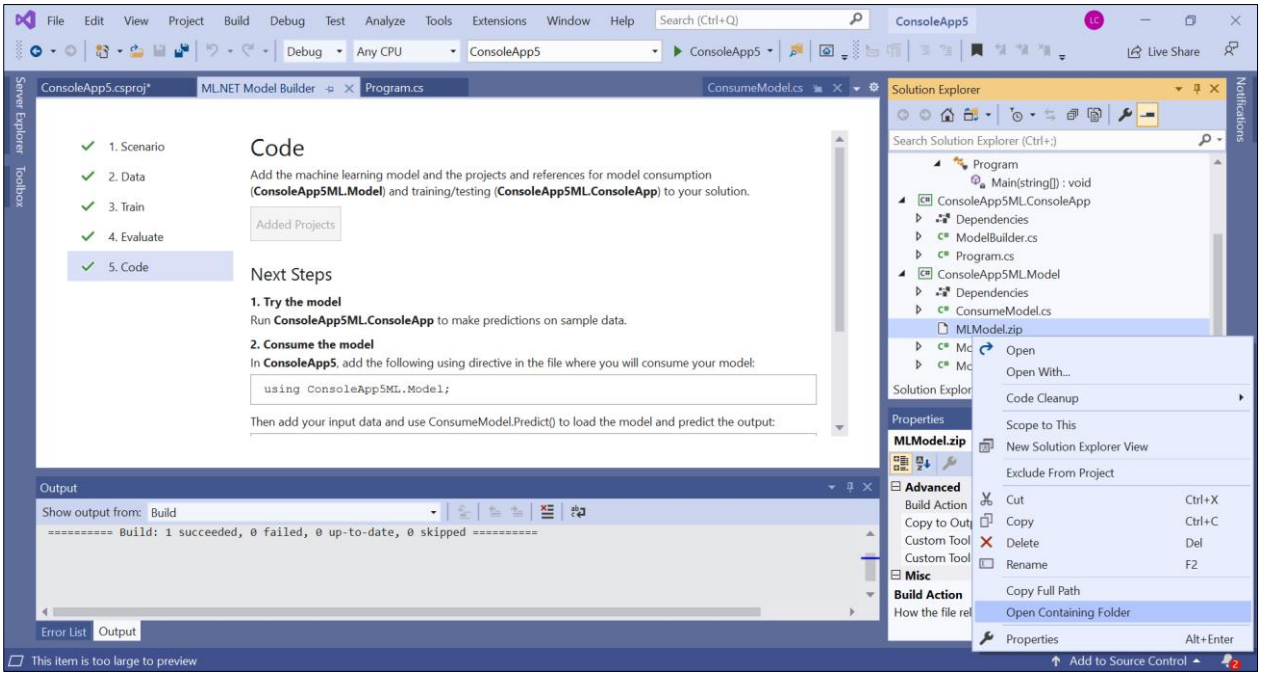

Figure 15.

Copy and substitute the file “MLModel.zip” in place of the previously present “\Program Files\Riverwell Consultancy Services Ltd\Scan Grow\MLModel.zip”.

4. Test the model

The “*3. Test the model”* section is used to compare the tagged absorbance values with the values given by the model.

Input format of the scanned images and the absorbance values: As shown in Figure 16, the working directory should contain both the original images in “inverted position” and the folders containing the cut images with their respective absorbance values (like shown in Figure 13).

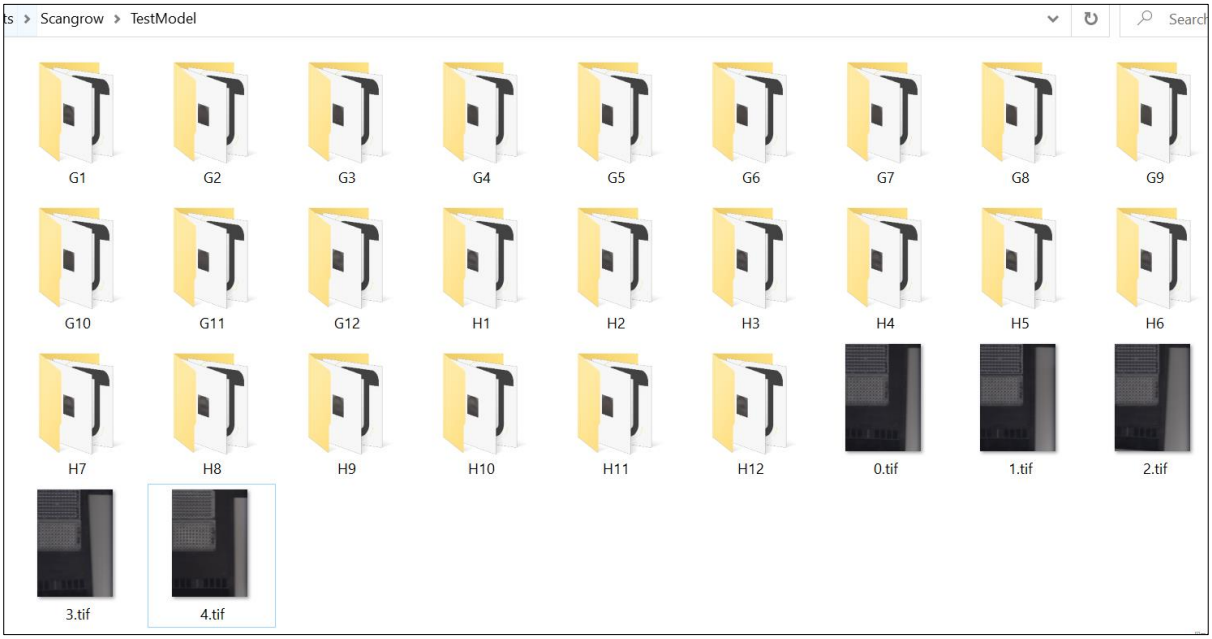

Figure 16.

In the app, select the working directory and click on *Test model*. Wait until the process is finished, and a new window will prompt you to save the results in .csv format, which should look like those shown in Figure 17.

	A	B	C	D	E	F	G	H
1	ScanId	Prediction	Prediction	Score	FileName	WellName	ActualClassification	
2	0	Level2	2	0.749462	0_A1_%_0^119.png		2	
3	0	Level2	2	0.940506	1_A1_%_0^173.png		3	
4	0	Level2	2	0.476943	2_A1_%_0^134.png		2	
5	0	Level2	2	0.903529	3_A1_%_0^072.png		2	
6	0	Level1	1	0.999871	4_A1_%_0^04.png		1	
7	0	Level3	3	0.633703	0_A10_%_0^202.png		3	
8	0	Level3	3	0.464827	1_A10_%_0^243.png		3	
9	0	Level3	3	0.518486	2_A10_%_0^275.png		4	
10	0	Level3	3	0.725688	3_A10_%_0^288.png		4	
11	0	Level1	1	0.99987	4_A10_%_0^041.png		1	
12	0	Level2	2	0.757578	0_A11_%_0^194.png		3	
13	0	Level3	3	0.85035	1_A11_%_0^249.png		3	
14	0	Level4	4	0.515775	2_A11_%_0^445.png		5	
15	0	Level5	5	0.853343	3_A11_%_0^479.png		5	
16	0	Level1	1	0.99724	4_A11_%_0^041.png		1	
17	0	Level4	4	0.56176	0_A12_%_0^229.png		3	
18	0	Level4	4	0.579197	1_A12_%_0^295.png		4	
19	0	Level2	2	0.674386	2_A12_%_0^174.png		3	
20	0	Level3	3	0.788426	3_A12_%_0^168.png		3	
21	0	Level1	1	0.999539	4_A12_%_0^044.png		1	

Figure 17.

Additional help: How to run ScanGrow with a Raspberry Pi

This quick tutorial shows how to program the automatic triggering of a scanner by controlling it with a Raspberry Pi. Scanned images can be later analysed with ScanGrow in an offline run.

Note: this tutorial was made for the scanner model Canon CanoScan LiDE 220. Other models may require different configuration settings.

- **Step 1. Control the Raspberry Pi via SSH**
  Connect your PC and the Raspberry Pi to the same WiFi network. It is recommended to run the Pi with HDMI monitor plugged in for the first time and:
  - Make sure the scanner works in Pi
  - Save the details of the WiFi connection and allow for automatic reconnection
  - Get the login and password details
  - Enable SSH (Preferences -> Raspberry Pi Configuration -> Interfaces)
  - Get the Pi's IP address of the connection by typing on the shell:

```
hostname -I
```

  The Raspberry Pi can be now connected without peripheral devices to the power outlet inside the anaerobic chamber, and with the scanner connected to one of the USB ports. Make sure the Pi is turned on and connected to the WiFi network.

  From your PC, use PuTTY or KiTTY to connect to via SSH to the Pi by introducing the Pi's IP address. When prompted, introduce the login and password details.

- **Step 2. Install SANE in Raspberry Pi**
  From the PuTTY/KiTTY terminal, make sure to have the latest version of the package listings by typing:

```
sudo apt update
```

  Install SANE:

```
sudo apt install sane-utils
```

  Check that SANE has been correctly installed and get the name of the scanner by typing:

```
sudo scanimage -L
```

You should read something like:

```
device `genesys:libusb:001:004' is a CANON CanoSlide 220 scanner
```

- **Step 3. Create script to trigger the scanner**

Go to the folder where you would like to create and store the script (eg `/home/pi/Documents`) and type:

```
nano scan.sh
```

This will open the Linux Command-line text editor. You can paste the following:

```
#!/bin/bash

for i in {1..96} #run of 48h
do
 echo "Scanning Image $i"
 #scan
 scanimage -d 'genesys:libusb:001:004' --resolution 300 --format
tiff --mode color >/home/pi/Documents/$i.tiff
 sleep 1780s #for every 30 min #3590s for every hour
done
```

Parameters to change if needed:
- Number of scans: from `96` to whatever needed
- Name of device: from `'genesys:libusb:001:004'` to the displayed name when typing `sudo scanimage -L`
- Folder to store the images: from `/home/pi/Documents` to the selected folder
- Sleep time (time between scans): from `1780s` to whatever needed

Save the script and exit.
Change the script permissions:

```
chmod 755 scan.sh
```

- **Step 4. Trigger the scanner and get the images**

To trigger the scanner every 30 min (or any other interval), from the folder where the script is saved, type:

```
sudo ./scan.sh
```

Note: it is highly recommended to use GNU Screen to save and recover sessions during the scanning process.

Install Screen by typing:

```
sudo apt-get install screen
```

Save a scanning session as “1”:

```
sudo screen -S 1 ./scan.sh
```

Now you can disconnect the Pi from the WiFi network if needed, and let the run progress. To reconnect again to the run, type:

```
sudo screen -r -d 1
```

When the run is finished, you can recover the images via SSH. A simple way is using WinSCP or any other application using SSH File Transfer Protocol. These images can be further analysed in ScanGrow using the offline run.