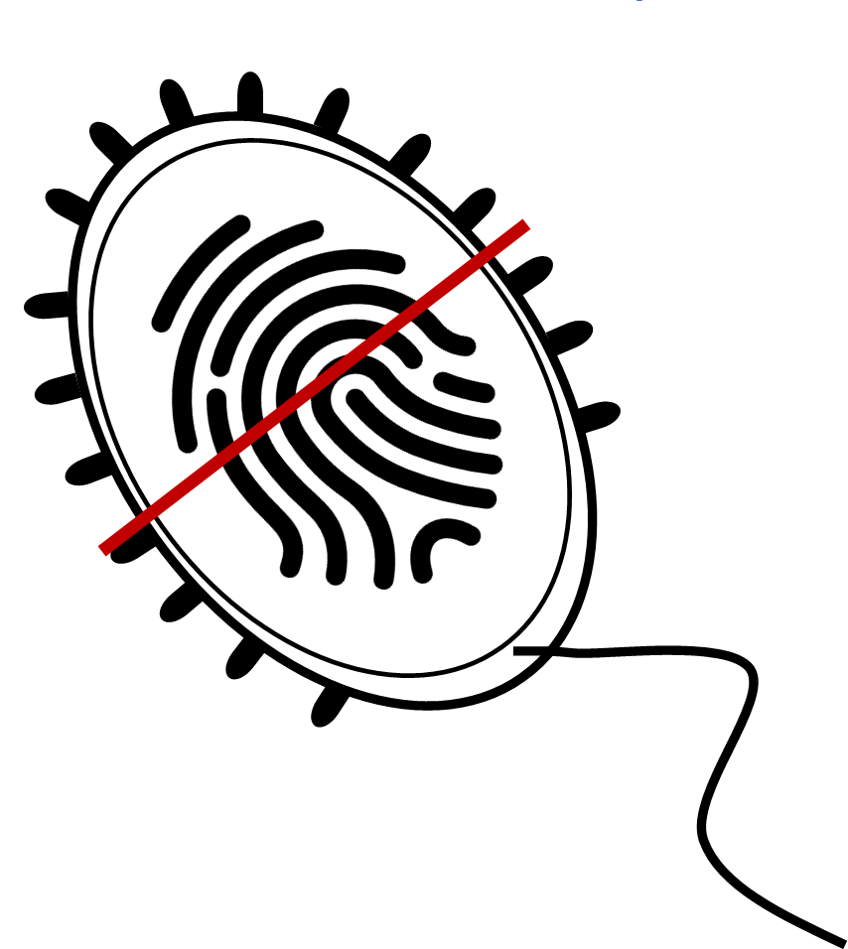
**CellScanner**



Let’s predict what is

in your medium

*Help File*

Table of Contents

[Introduction: 3](#_Toc55408904)

[I- Installation: 3](#_Toc55408905)

[Windows 3](#_Toc55408906)

[Linux /Mac 3](#_Toc55408907)

[II- Main Window: 4](#_Toc55408908)

[III- Database and accesses: 5](#_Toc55408909)

[Flow cytometer button: 5](#_Toc55408910)

[Update data function 5](#_Toc55408911)

[IV- Tool Analysis: 6](#_Toc55408912)

[V- New prediction: 7](#_Toc55408913)

[VI- Clustering & clustering analysis: 7](#_Toc55408914)

[VII- Settings window: 8](#_Toc55408915)

[IX- The Program (a refaire) 9](#_Toc55408916)

[Parameter selection: 14](#_Toc55408917)

[LEXICON: 15](#_Toc55408918)

[FUNCTIONS: 15](#_Toc55408919)

# Introduction:

CellScanner has been developed to analyse bacterial flow cytometry data for biologists, and predict the content of samples based on already known reference data. The program takes .fcs or .csv files as an input. From monoculture information called reference files, CellScanner can identify the most representative parameters of each species. Based on supervised machine learning methods, the input data will serve to create a model able to predict the species present in a monoculture or a community (in-vitro or in-silico). The potential species in the community need to be known to train the model. More species you add in the prediction, more likely the prediction will be altered. Be aware that the quality of the prediction will depend on the quality of the reference information. Two main function allows the user to assess the quality of prediction for an in-silico community, then helping the user to interpret result on a in-vitro community prediction.

This handy interface has been developed for laboratory personnel. The goal was that people with no informatic background could use this program to have information from flow cytometry data. All steps are clickable, limiting the mistakes possibilities.

The program contains a database which makes it adaptative to the user needs. If the database already contains the class ‘Species’ with different records (a particular species) and reference files (monoculture link to a specific record) , you can add new classes and new species. A class doesn’t have to contain species. It can be defined by a group of cells such as the shape type or the gram type.

This help file will explain how to use the program and how it works. All parameters are referenced here. We also describe what are the best parameters and in which condition. Parameters helps the user to go always further with the tool.

# I- Installation:

## Windows

Download the installer from the <https://github.com/Clem-Jos/CellScanner> web page and run it. Chose the location of the program. The directory created in the chosen location contains:

* The executable file to run the program
* The database bd.db
* The ‘References’ directory where the reference monoculture information is stored.

When the user import reference data, files are copied in the References directory. This directory should not be modified by the user other than from the database management function via the tool.

* The ‘Results’ directory where the prediction and analysis file will be stored.

The result for all analysis and prediction running will be stored in the Result directory.

* The Help.pdf File

## Linux /Mac

Package installing:

* Install Python 3.7 if not already installed in the computer
* Check for pip3 or pip in the cmd
* pip install pyqt5
* pip install sklearn
* pip install fcsparser
* pip install matplotlib.pyplot

Download or Clone the directory containing, the scripts, database file, Result directory References directory and the Help.pdf file.

# CellScannerII- Main Window:

The main window allows launching the different functions of the tool.

At the bottom part of the window, 5 buttons resume the principal functions:

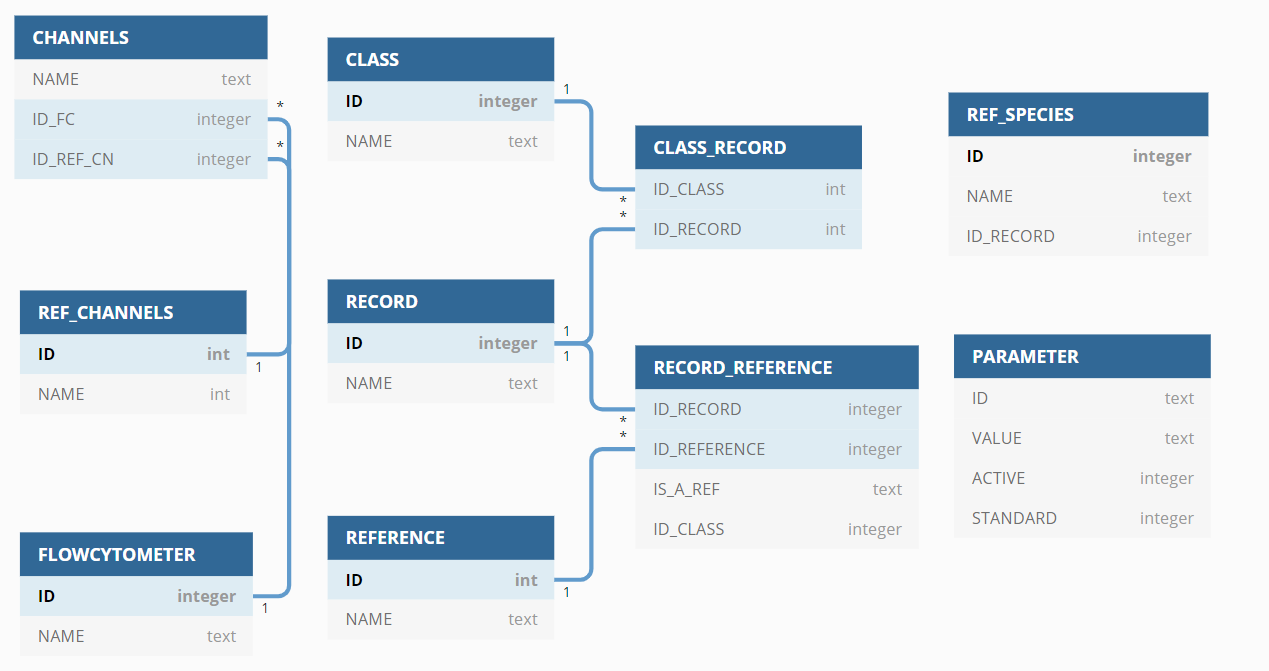
* **New Prediction**: Launch the program function for prediction(s) within *in-vitro* data files.
* **Tool Analysis**:  Launch the program function to assess the program ability to differentiate species(classes) by building an *in-silico* community.
* **Clustering:** Launch the program function to find cluster in an *in-vitro* community, and labelling them according to reference file.
* **Clustering Analysis:** Launch the program function to assess the program ability to find cluster and label them correctly by building an *in-silico* community.
* **Update Data:** Give access to the database. This function allows the user to define, modify and complete the reference data information. Setting the reference data can be necessary before running a tool analysis or a prediction.

At the top of the window the toolbar contains 3 buttons:

* **Exit** button: to quit the program
* **Parameter** button: Give access to the advanced parameter
* **Flow cytometer** button: Give access to the flow cytometer parameter, the user can define which parameter to take in account within a specific cytometer.

# III- Database and accesses:

The application is linked to a database which contains needed information to run predictions on the data from flow cytometry. The database is built as followed:

            Figure x: Database structure

The database is divided in 3 parts. The first one contains the tables ‘CHANNELS’, ‘REF\_CHANNELS’ and ‘FLOWCYTOMETER’ and is accessible by the user through the **Flow cytometer** button. The second part contains the tables ‘CLASS’, ’RECORD’, ’REFERENCE’, ’CLASS\_RECORD’, ’RECORD\_REFERENCE’, and ‘REF\_SPECIES’ and describes the reference data information. This part is accessible through the **Update Data** function. The last table, PARAMETER, contains all the advanced parameter information. This part is accessible via the **Parameter** button.

## Flow cytometer button:

According to flow cytometers used by the user, the channel names can change. The program needs to know which channels are present, and which one should be taken in account for the analysis.

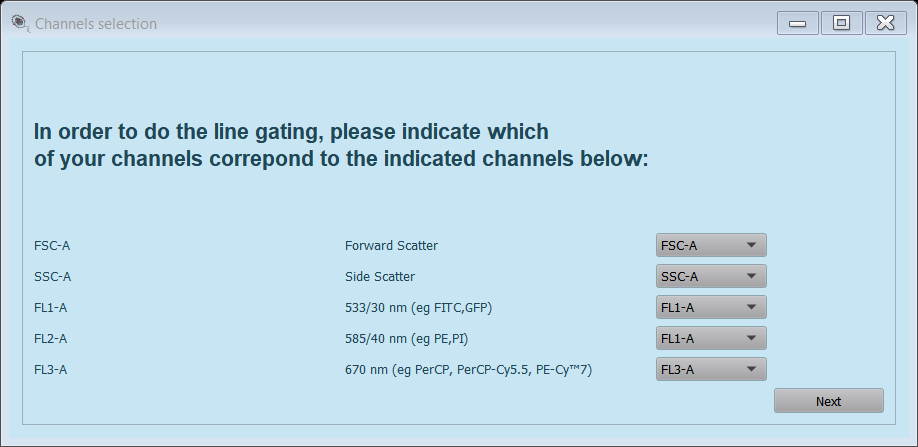
In fact, the user will decide which channels to use when creating a flow cytometer. Reducing the number of channels selected, reduce the computing time.

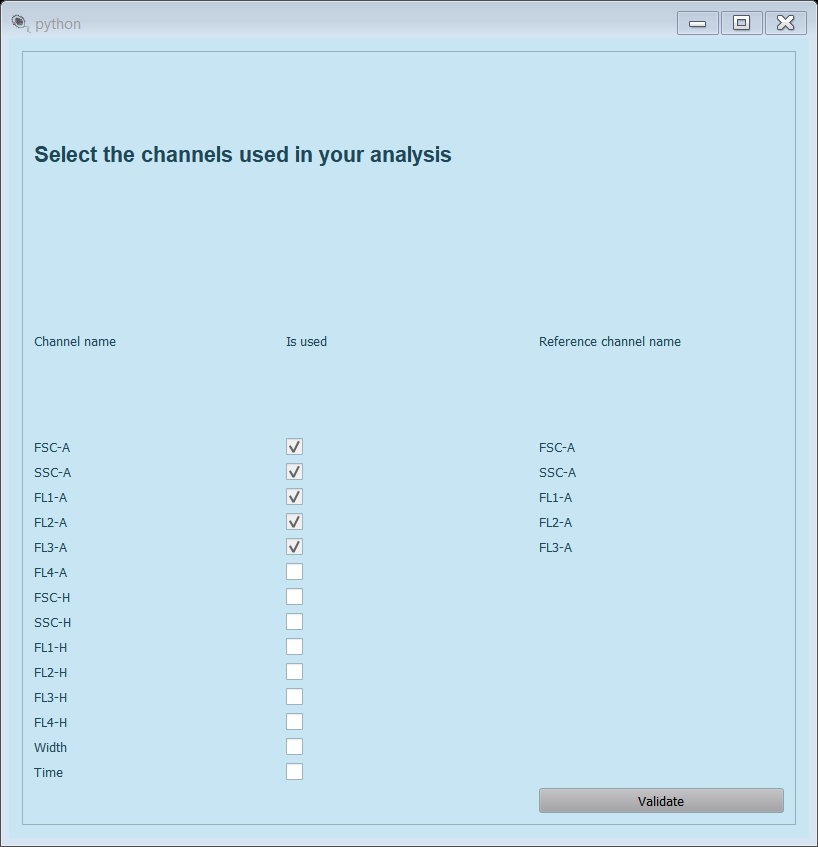
Furthermore, some channel, needs to be identified to perform the line gating (cf program/gating).

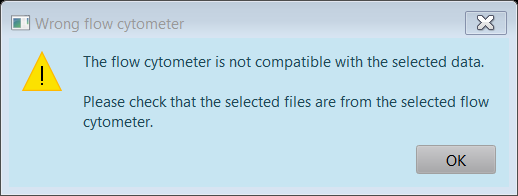
The flow cytometer window allows the user to manage flow cytometer information. On the top, the user can select the active cytometer, by choosing in the list, and clicking the ok button. The user can create, update or delete a flow cytometer by using the buttons below the list.

* Add a flow cytometer

When creating a new flow cytometer, the user has to select a name, which is not already existing on the list, or an error message will pop up. A value is added in the FLOWCYTOMETER table in the database. After confirmation of the cytometer name, a window allows the user to select a csv of fcs file, produced by the target cytometer. The program will extract the column names from this one. The next step, ask the user to identify channels from his file corresponding to reference channels according to the description. This reference channels will be used by the program to perform the line gating.



Once the reference channels are indicated by the user, he can decide which channels to use for the predictions. Channels information is saved in the CHANNELS table in the database.

The active cytometer needs to match the reference data files, and the files selected for the prediction. If it is not the case, the program will alert the user and cancel the prediction.

CAUTION: Line gating has been developed for Accuri and Cytoflex flow cytometer. Please include one of these names in your cytometer name, if you want to perform one or the other line gating.

## Update data function

A class is defined by several records. Each record should have at least one reference characterized by a flow cytometry data file. For example, in the tool, you can find the class ‘Species’, containing one record for each species.

A class cannot contain two times the same record, but the same record can be found in different classes. In this case, the references for a record will depend on the class. You can then set several classes depending on your experiments, with different reference files. Which will facilitate your prediction step.

You can complete the database by creating new classes to dissociate events or profiles (species, Gram, shape, Experiment). The name of the new class cannot be used twice.

The Update data window allows you to:

- Create a class

After clicking the button, you can enter the name of the new class, after validation, the new class is created.

- Delete a class:

After clicking the button, you have to select the class/es that you want to delete. After validation the class/es is/are deleted.

- Update a class:

Select a class and click the button update. A new window appears where you can alter the record and the reference for the selected class.

The Class species is already completed with some data. You can add or delete a record. For the Class Species, an auto-completion system exists and can help you create your species’ record. Then for each species, you can add one or several monoculture files as a reference, by using the ‘Add a Ref’ button in front of the specific record. You will be able to select your reference files through your directories. These will be copied in the reference directory of the program. Please remember that the name of your file will help you to recognize which kind of condition you used. So, we advise you to name it completely.

Once added in the database, the file can be settled as a reference for the prediction and tool analysis, by checking the file in the popup list next to the record name. Several files per species can be used as references.

Be aware that if no reference is selected for a record, it will not be available in the prediction or assessment function.

Once you completed the database, you can run a prediction or assessment.

# IV- Tool Analysis:

Tool analysis function has been developed to assess the prediction accuracy with a set of known information. The program will predict the composition of a fake coculture created with monoculture file.

- Select the Class

- Select the number of record maximum you expected. Select the record/’species’ you expected from the list. If the list is incomplete, be sure you furnished all information through the function update data. The reference information is taken from the database. You can change the reference file via the Update function on the main window.

- Select one, several or none file for each species (at least one file in total)

- Click OK

The running time can variate depending on your computer and the parameters of the prediction. It can take some minutes to 20 min. Once the calculation is done, a report appears with the information about the result. Depending on the parameter you selected, some 2d or 3d graph and matrix can appear on the screen. You can save them on your own or only close them. The calculation will continue after closing the graph.

2 steps are presented in the result. The training step allows the user to create a predictive model. It only takes a part of the reference files to train the model and a part to predict. The predicted results are presented via a3d graph, a confusion matrix, the References and Learning files. Goes on the chapter Result Analysis to understand how they are built.

If the result if good enough for your study, you can run a prediction with the same species.

# V- New prediction:

The principle is close to the tool analysis method unless there is no information about the content of the given files.

- Select the Class

- Select the number of ‘species’ or record maximum you expected. Select the record/’species’ you expected from the list. If the list is incomplete, be sure you furnished all information through the function update data. The reference files are already indicated in the database.

- Select one, several files that you want to analyse. (at least one file). You will have a report for each selected file.

You can now run the prediction. The running time depends on the number of ‘species’ you gave to analyse and on the size of the files.

Files related to the prediction

If the app is not sure about the species from an event it can be labelled unknown. In this case, the app hesitated between closest species. Be aware that if you don’t indicate all possible species in your medium, the app will identify it as a species you specified and not always in unknown.

# VI- Clustering & clustering analysis:

This part of the program is additional. To complete the program with a further function, we used a sklearn function to identify clusters and suggest a species annotation. As for the prediction and the tool analysis, the clustering function performs a prediction and the analysis assesses the clustering method, by creating a fake coculture from monocultures.

These functions are useful for species creating distinct clusters, and

# VII- Settings window:

The settings windows give access to the database by selecting the parameter needed for the run of a prediction or an analysis.

- **Number of runs**: number of model and prediction done for one file.

- **Create average prediction from runs**:

- **Percentage minimum of appearance to validate a prediction**

- **Figures view mode**:

the user can decide if the figure pops up on the screen, which allows the user to move the 3D graph to the best position. It can select save, then the figures will automatically be saved on the result directory, or the user can select none, and no graph is produced during the experiment.

- **Select 3D graph axes**: allows the user to select the 3 axes on the 3D graphs

- **Cluster distance %:** percentage of distance maximum to address a clsuser to a species.

- **Gating type**: select the gating method between None, line or machine.

None doesn’t proceed any gating. We advise adding Blank as a species during the prediction.

Line: the program will use rules described in the script on the *gatingFunction* function on the *machineLearningPackage*package, remove cells responding at the following conditions:

Accuri

Cytoflex

If the selected cytometer is different from Accuri or Cytoflex (doesn’t contain either name), none gating is performed

Machine: The program uses reference of species called Blank (or containing [‘Blk’, ‘blk’, ‘Blank’, ‘blank’]) To identify blank into every other reference file (and monoculture for tool analysis). If you use this method, remember to select Blank as a species during the prediction.

- **Showing gating effect**:

This parameter shows or save 2D graph showing the effect of gating on data with the line gating. In the same way, a 3d graph is produced with the machine learning gating.

- **Training**: **number of cells per record**:

The number of cells randomly selected per record, after merging the different reference from the same record. If the file contains less cells, the program will select the maximum number.

- **Tool analysis: number of cells per record**:

The number of cells randomly selected per record after merging the different input files from the same record. If the file contains less cells, the program will select the maximum number.

- **Machine learning method**:

The prediction can be run with different sklearn machine learning method.

* + Neural network: cf program
  + Random forest: cf program
  + Logistic regression: cf program
  + Random guessing: function which attribute randomly a record to a cell. Allows you to differentiate the effect of a predictive program and an artefact prediction.

- **Ratio Learning/Test**:

From the number of cells selected for the machine learning training, the tool selects a part to train the model and the rest to test the model. This ratio is by default set at 6/7 for the training and 1/7 (0,14) for the test.

The *Back on the default* button allows the user to reset the default settings. If the settings are modified, the changes are saved using the *save settings* buttin. You need to press the button save the settings. The settings are kept in the database until you change them or you go back to default values.

# IX- The Program (a refaire)

The program is a python-based program using pyQt5 and scikit-learn packages. The scikit-learn function are used to process data from flow cytometry via machine learning methods such as neural network, random forest, logistic regression or again clustering algorithm.

The program used the supervised classification to classify event into different classes. But a unsupervised clustering function as been added, in case you don’t have relevant reference information from the different classes. To label the clusters, the program use reference, not as a training but as an indication about where a cluster could be expected. In this case, the species are labelled.

The interface allows reduce the manual entry by the user, and then limiting the errors. Two principal functions are called by the interface to run the classification or clustering experiments: The developed python function can be used by an advanced user. As a command line.

* Prediction
* Clustering (this function can only be used if the cluster are distant enough to be recognise as different clusters)

The prediction function:

the program takes files names, for the references and associated species, files for analysis or prediction, and species if known. It also needs more parameters that are available from the advanced parameter such as the number of repeats for a prediction, ratio learning prediction, number of referenced events for each record.

######.

Files: List of reference files

Species: list of species (same length as files, because one ‘species’/ profile per files)

Files2: list of files to analyse. In the case of assessment, files will be merged, not in prediction.

nbC: number of records used as reference per species default 1000

Gating: type of gating performed on referenced and files for prediction.

A blank can be considered as a profile and gating can be avoided.

Average: True/False

If the average parameter is set as True, the average prediction is calculated for each record in the file from the several repeat predictions. A new prediction called average is produced. If the parameter is set as False, the average is only calculated with the final ratio. *Capture again the function and describe it again.*

**In the case of an analysis:**

What does the function

The steps:

1- The option.txt file is created, and the result directory is created in the CellScanner/Results/ directory. Its name is composed of the date and time when the analysis starts.

Ex: I ran an experiment the 23 of April 2020 at 13h32 and 53 sec, the result of my experiment will be saved in the directory 230420-13\_32\_52

2- Import data from files, convert the column name if necessary and proceed the gating.

3- Run n times the following steps (number of repeats):

* 1. Treat the data from the reference files (merge record, randomly select a specific number of cells for the reference and the tool analysis, tag each cell with the record name) merge all files data into one array.
  2. Train a model and test the model with one of the different methods. (neural network, random forest, logistic regression, random forest).
  3. Produce a confusion matrix for the test results
  4. Produce a 3D graph for the 1 run only with expected results and predicted results.
  5. In case of prediction, for each file selected for prediction, the model is run on the file, and a prediction is produced for each

In the case of tool analysis, the record from the same species are merged and a specific number of cells is randomly selected per record, depending on the parameter. If the parameter is set as None (or 0 in the application) the entire file is used for prediction. All the selection is then tagged and merge into one array. This array goes in prediction using the model produced by the reference files. A confusion matrix is also produced for each turn

* 1. If the average option is not selected, a 3D graph is produced for the first run of each prediction.

4- In the case of 10 runs, an average confusion matrix is calculated from the 10 reference confusion matrixes produced. A figure is produced.

5- If the average parameter is checked, for each file to predict or for the merge files of the tool analysis function, an average prediction is calculated based on the different runs as explain in the chapter about average prediction.

From these average predictions, 3D graphs are produced. In the case of the tool analysis function, an average confusion matrix is also produced.

6- Statistic files are produced and save in the result directory

7-

 If the average parameter is not checked, for each file to predict or for the merged files created with the tool analysis function, and for each run, the statistics are kept in the produced files (see results files in this case). In the case of tool analysis, an average cm is calculated from the different runs. The figures are saved or shown depending on the parameters.

**In the case of a prediction:**

The steps are the same for every steps except the following:

The clustering function:

##################

**Tool analysis description for a number of runs equal to 10**

The programme gave the possibility to create an average prediction from the different runs. It does mean that the program will not calculate the average predicted number of cells per record and per file, but decide for each event in a file what is the average prediction. It is, in this case, possible that an event is most of the time predicted as a specific record. For that, we analyse all the predicted result from the different runs.

  For each cell we look at the 10 predicted value, we count the occurrence per record and define the most prevalent as the predicted record. We can add another parameter on this function, which is the minimum of appearance needed. The default value is 70%. If within 10 runs a cell is recognized more than 7 times as the same record then the cell is tagged with the species name. If it is less than that the record is tagged as an unknown species.

**Average and minimum ratio of prediction for tool analysis**

You can use this parameter to see if the program can strictly differentiate the species or if there is some overlapping space between the species. With this function, you will decrease the sensitivity, but you will also increase the precision.

Be aware that the number of cells in the community will also have an impact on the number of cells predicted as unknown. If you put 50% as a minimum of appearance for a 2 records community, you are asking to the model to select species predicted at least one more time for a record than the other. It is close to the random guessing. In a case of 4 records, if you also put 50% of the prediction minimum, you expect two times more prediction than in random guessing which is in this case 25% for each record.

When you have a bigger community, the overlap space is likely increased. So, the unknown part will also increase.

**Result Files and figures.**

All produced files and figures can be found in the Result directory of the tool.

-       **Reference**.csv:

Statistics from the training set on the model creation. Accuracy, F1, and their standard deviation. Then, for each species the sensitivity and the precision with the standard deviation

-       **Learning**.csv:

  Prediction tables for the training set of the model creation.

One table is created per run.

-       **Prediction**.csv:

If Prediction:

For each file given to prediction, one or several tables are given, depending on the average parameter and the number of runs. The table contains the number or cell predicted in the file for each record selected as a possible alternative for the prediction.

In the case of the average prediction is calculated, you will also visualize the unknown record. Depending on the setting of the average minimum prediction rate parameter, it can contain more or less predicted results.

If Assessment: #TODO should I had the number of cells predicted per species?

              The file contains a table containing the statistics for the prediction. It contains the same data than for Learning.csv. If the average parameter is selected, or if the number of runs is 1, the file will only contain one table. If not, the file will contain as many tables as the number of runs.

-       **Assessment**.csv (Tool analysis only):

This file contains the prediction statistics for the fake community. Created with the selected file. It the same composition than for the Reference.csv.

-       **option**.txt:

The options file is produced to save the selected parameter for your prediction or tool analysis. It contains, the files selected as references, the file used for the prediction, the number of cells used for the model creation, and the number of cells used per species for tool analysis fake community reconstruction. It also contains the parameter selected for the machine learning program and the calculation. It traces the way you run the program for a prediction or analysis.

- Cm.xml:

The cm file contains all the confusion matrix (raw values and ratios)

**Confusion Matrix:**

A confusion matrix is a table indexing all the result for a prediction when the expected result is known. A matrix is only produced in case of a test of the model and for a tool analysis prediction. The matrices produced by the model is normalized. The results are shown in percentage per species.

Example:

 In this case, a confusion matrix is produced from the reference files on purpose to assess the model created with 1/7 of the given data. Here the class selected is Species and the records are Bacteroides uniformis and Bacteroides thetaiotaomicron. The True Labels are the expected result, and we can see that 66 % of the Bacteroides uniformis is well identified but 34% is predicted as Bacteroides thetaiotaomicron.

With this matrix, you can easily visualize if the species can be differentiated.

**3d Graph:**

If the Figure view mode is different from 'None', 3D graphs are produced for the step of Training, with the reference file, and for the step of prediction for both function (prediction and tool analysis).

The axes can be chosen in the parameter window. The window is divided into 2 graphs: The expected results on the left and the predicted results

# Parameter selection:

How to choose parameters:

- Gating :

Machine gating way longer to perform. On some case (few events, blank easily separable by line gating) it is more efficient to use the line gating. If the flow cytometer detects more particles, we advise using machine learning gating.

- Number of cells:

The number of cells initially selected in the parameter affects the learning step of the predictive model (Tool analysis and prediction). More cells you’re selected, more cells will be used to train the model, according to the testing/learning ratio(1/7).

With a number of run >=10, the number of cells efficient is between 1000-5000. Over 5000 cells, the calculation time increases significantly(5 to 10 min per run).

- Ratio

By default, the ration is 1/7 according to sklearn function

- Unknown

Result with the unknown will variate with the number of runs, and the number of species. More you have species, more the species will overlap. If you selected 70% minimum prediction to validate the value, you will have more unknown than with 50%.

If you want to follow the present species for high communities(at least 3 species) we advise to use the unknown parameter between 50 and 70 %.

If you want to follow a small community you can use unknown for 70 to 90% if you want to reduce false positives. If you want to compare ratios in a community, not using the unknown parameter is more efficient.

-

# LEXICON:

**Reference file**: flow cytometric file where the composition is known and used to train the model. In the case of species, it has to be a known, monoculture. One reference File has to contain information about one record from the selected class.

**Class**: Name of a group of records, use to create an analysis. By default, the program contains the class Species.

**Record**: element from a class which as to be identified during analysis. By default, a record from a class species is a bacterial species (e.g esherichia.coli).

# FUNCTIONS:

machineLearningPackage:

This package contains all individual functions useful for all the other packages.

***addSpeciesTag***(narray, species):

***gatingFunction***(narray, save=None, filename='', cwd='', fc='Accuri'):

***randomSelection***(narray, nbC=1000, random\_state=None):

***splitInformation***(narray):

***logisticRegression***(data, target, ratio, logReg='l2', asolver='lbfgs', random\_state=None):

***neuralNetwork***(data, target, ratio, activation='relu', solver='lbfgs', max\_iter=1000, random\_state=None):

***randomForest***(data, target, ratio, n\_estimators=200, criterion='gini', random\_state=None):

***randomGuessing***(data, species, ptype='train', target=None, ratio=1 / 7, random\_state=None):

***scalerTest***(data, scaler, logisticReg):

***statAnalysis***(predicted, known, species): # TO OPTIMIZE I can directly extract the matrix

***exportPrediction***(predictedLbl, samples, cwd, typeF, run, typeP='AVERAGE', repeat=0):

***exportStatistics***(statistics, samples, cwd, typeF):

***constructStatDF***(nbB):

**constructRatioDF**(nbB, species):

**meanPrediction**(data, samples, SP):

**calculateStatDataFrame**(data, samples, align):

**graph3d**(data, predict, target, species, param=['FL1-A','FL3-A','FSC-A'], statistics=None, show='show', cwd='', repeat=1, name='', predtype='analysis', clust=False):

**graph3dRef**(data, predict, target, species, param, statistics, show, cwd, refdata, reflabel, repeat=1, name='', predtype='analysis', clust=False):

**mergeSameSpecies**(arrays, species):

**assessmentValue**(statValues, species, cwd, sample, typeF):

**fileOption**(cwd, files, species, files2, species2, nbC, nbC2, gating='line', predAn='prediction',

predtype='neur', ratio=1 / 7.0, repeat=1, average=True,

doubt=0):

**createCombination**(init\_condition, nbComb):

**meanPrediction**(data, samples, SP):

**calculateStatDataFrame**(data, samples, align):

**unique**(list1):

**deleteBis**(valbis):

**position**(valuesL):

**isBetter**(clusterSP, dicSP, spp, dsp, var=0.05):

**distance**(pav, rav, column=['FSC-A', 'FL1-A', 'FL3-A']):

clusterScript:

This package contains a function used for the clustering functions: Clustering and Clustering analysis.

**assignCluster**(dist, sp):

**annotation**(data, prediction, clust\_nb, nmax, mpRef, param, species):

**clustering**(refFiles, species, predFiles, sppred, predAn, param, nbC=1000, nbC2=None, save='show', var=0.05, gating='line', showgat=None, method=1, fc='Accuri'):

machineGating:

This package contains functions used for machine learning gating used for the *New Prediction* and *Tool Analysis* function.

**getSorted**(aSp, arrays, species)

**gate**(label, array):

**machineGating**(refArrays, species, predArrays, species2, predType='neur', ratio=1/2, random\_state=None, show='save', cwd='', rept=10, name='ref'):

runScript:

This package contains the main functions used to execute the *New Prediction,* and the *Tool analysis*functions.

**importFile**()

**treat**(someArrays, species, nbC, mode='pred', cluster=False, random\_state=None):

**learning**(predType, data, target, ratio, random\_state, species=None):

**predict**(predtype, scaler, classifier, data, species):

**averageConfM**(ConfMList):

**bestPred**(predict\_lbls, doubt=0):

**plotConfusionMatrix**(cm, classes, save, cwd, name='', normalize=False,

title=None, cmap=plt.cm.Blues, predAn='prediction'):

**predictionMultiple**(files, refArrays, species, files2, Data, target2, nbC, repeat=0,

param=None, predAn='prediction', predType='rand', ratio=1 / 7.0, random\_state=None,

save='show', cwd='', average=True):

**predictions**(files, species, files2, species2, nbC=1000, nbC2=None, gating='line', showgat=False,

                predAn='prediction',

predtype='neur', ratio=1 / 7.0, repeat=1, average=True,

doubt=0, random\_state=None, save='save', fc='Accuri',

param=None):

runCalculation:

This package contains the function linking the interface script and the calculation scripts.

run(self):