UHASSELT
KNOWLEDGE IN ACTION

**www.uhasselt.be**
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

# Faculty of Sciences

## *School for Information Technology*

Master of Statistics and Data Science

***Master's thesis***

***Who is who? Detecting plankton strains in a green soup***

**Johanna Hernandez**
Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Data Science

**SUPERVISOR :**
Prof. dr. Olivier THAS

**SUPERVISOR :**
Dr. Frederik DE LAENDER

2024
2025

# Faculty of Sciences

## *School for Information Technology*

Master of Statistics and Data Science

***Master's thesis***

***Who is who? Detecting plankton strains in a green soup***

**Johanna Hernandez**

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Data Science

**SUPERVISOR :**

Prof. dr. Olivier THAS

**SUPERVISOR :**

Dr. Frederik DE LAENDER

# Contents

**Abstract**

Given the role of phytoplankton in marine ecosystems, it is crucial to be able to accurately estimate their population both in their natural habitat and in experimental situations where the effects of different agents or conditions would want to be studied. Part of this process is to successfully identify the correct species and strain to which a phytoplankton cell cultivated in organic multi-cultures, i.e., different strains or species grown together, belongs. In this study, we intended to allocate the total population estimates of organic multi-strain cultures to each individual strain present in the culture. This was done using the light-intercepting capabilities of phytoplankton cells measured using a cytometer as primary predictors. Several base models were trained on synthetic multi-cultures, i.e., concatenated mono-cultures, using supervised learning; and the results of these models were further combined in various strategies as a form of ensembling. The resulting models were shown to be more accurate in distinguishing strains of different species than strains of the same species. Although differentiating strains of species V (2375 and 2524) and strains of species VIII (2383, 2434) require most cytometer outputs primarily FSC, and RED.R respectively, when differentiating strains of different species, the models mainly depended on YEL.B. Although the models were able to accurately assign phytoplankton cells to their true strains, with an accuracy ranging from 92.81% to 95.78%, the models presented need to be used with caution on organic multi-culture data given that they differ from synthetic multi-culture data, primarily for lack of inter-strain or inter-species interactions.

# 1 Introduction

Phytoplankton are photosynthetic marine microorganisms capable of capturing light energy that thrive in open waters such as lakes, rivers and oceans, and are displaced passively through water currents or actively using their locomotory organs known as flagella [1] [2].

Picoplankton is a kind of phytoplankton whose size ranges from 0.2 to 2$\mu$m. Although considered one of the smallest phytoplankton due to its size, the picoplanktonic genera of Synechococcus, with *Prochlorococcus* and *Synechocystis*, comprises 30-50% of phytoplankton biomass [2]. Furthermore, it has been shown that, along with nano- and micro-planktons, they maintain the primary productivity in the oceans [2]. In particular, cyanobacteria represent approximately 10% of the total primary production on a global scale for the period 1998–2011 [3]. Therefore, it is important to study and understand how these microorganisms thrive, grow, and interact with their environment. And part of this involves successfully identifying the exact species or strain of phytoplankton in order to correctly attribute a phenomenon or a process to the right species or strain of phytoplankton.

Several techniques and methodologies have already been developed in order to perform such tasks. One of the most basic is through microscopic identification. However, because this is an expensive and time-consuming process, not to mention limited to only phytoplankton larger than 8-10$\mu$m [4], this approach is inefficient. An alternative faster approach is pigment analysis. Analyzing pigment composition and coupling with genetic diversity or morphological variations, for example, has been helpful in categorizing picoplanktons [2]. However, this approach is not capable of distinguishing different phytoplankton strains. Thus, pigment analysis has been used increasingly with molecular methods to understand picoplankton populations [2]. However, although DNA sequencing techniques and real-time amplification methodologies are very reliable in taxonomical endeavors, they can be expensive and require highly trained personnel [5].

Another alternative is to classify phytoplankton into meaningful functional groups based on their morphology. This was shown to be a sufficient technique that captures a lot of the functional properties of phytoplankton and does not require taxonomic affiliations and can be used even for species with unknown physiological traits [6]. An example of this is using a cytometer to assess the light interception capabilities of phytoplankton cells. Given a sample culture, the machine works by sucking cells one by one and exposing them to different laser frequencies. Then the machine measures how the light refracts on the cells. And for this research, these cytometer readings or measurements are of primary interest.

### 1.1 Relevance and Stakeholders

Biodiversity plays a crucial role in the maintenance of healthy ecosystems [7]. It is therefore important to be able to measure the population of different flora and fauna existing in nature. In particular, this research focuses on phytoplankton, which can be found at the base of the aquatic food web [8]. This means that any perturbations in the population of these aquatic organisms have a direct implication on the ecosystem as a whole.

This research is relevant because it focuses on estimating the population of phytoplankton. Although this research is not concerned with directly measuring the population of phytoplankton as they naturally occur in marine environments and focuses on the phytoplankton population in controlled environments, this research will allow empiricists and biodiversity scientists to gain understanding of their behavior and how different environmental factors and situations might affect the growth or decline of their population.

### 1.2 Ethical consideration

For the experiments carried out in this study, the starting cultures were collected directly from their natural habitat or sampled from existing cultures grown in the laboratory. In the former case, given the phytoplankton population, which systematically doubles in mass daily [9], collecting a small quantity needed to start the experiment does not pose any threat nor endanger the phytoplankton community in any way. In the latter case, the samples used did not directly affect phytoplankton in the oceans.

The cultures used in the experiments were exposed to different conditions determined by varying temperatures and possibly exposure to a certain type of herbicide, atrazine. Although the different temperatures with which the cultures were kept fall within the temperature range where phytoplankton, specifically for *Synechococcus*, thrives in nature, at least above 14° [10], exposure of some phytoplankton to atrazine might be unnatural and harmful. However, studying the possible effects of this herbicide on how phytoplankton thrives in such environment is of interest.

In addition, provided that the experiments are conducted in the laboratory, they did not directly influence the marine ecosystem, thus avoiding the risk of causing any harm to the environment. Lastly, the experiment was carried out by competent and credible scientists. The setup of the experiment and the data were well documented, ensuring the authenticity of the experiment and the data collected, and making the experiment reproducible.

### 1.3 Problem Description

For this research, we examine how to allocate the population of a given organic multi-strain culture in to the different strains that are present in that culture. In order to do this, we explore different ways of classifying individual cells into their respective strains primarily using their light-interception abilities quantified using a cytometer. We create various models trained on synthetic multi-strain cultures, that is, on concatenated data of mono-strain cultures, that could be used as tools in the given experimental set-up and provide a methodology on how future experimental data can be analyzed.

## 2 Experiment Design

For this particular study, four different strains of synechococcus bacteria were selected: strains 2375 and 2524, which both belonged to species V, and strains 2383 and 2434, which belonged to species VIII.

Each synechococcus strain was grown in isolation, referred to as mono-culture, to know its light-refracting characteristics. And to replicate the co-occurrence of various strains in nature, several strains were grown together. This resulted in six duo-cultures, four tri-cultures and one tetra-culture. These will be referred to as organic multi-cultures. Classifying the cells of samples of these organic multi-cultures is one of the main objectives of this study.

These cultures were exposed to different conditions as defined by temperature (22°C and 24°C) and atrazine concentrations (0 mg/L and 0.1 mg/L). And each experiment set-up has been replicated three times, thus producing 180 cultures (15 strain combinations x 4 conditions x 3 replicates) in total. Each of these cultures was sampled over the course of 21 days. Each sample was fed to the cytometer, which exposes individual cells to 8 different light frequencies and measures how they intercept light. This in turn creates 8 different possibly correlated output measurements.

| Strain | Species | Condition | | Replicate |
|---|---|---|---|---|
| Combination | Combination | Temperature | Atrazine Concentration | |
| Mono-culture<br>2375, 2383, 2434, 2524 | V, VIII | 22°C, 24°C | 0 mg/L, 0.1 mg/L | 1, 2, 3 |
| Duo-culture<br>2375_2383, 2375_2434,<br>2375_2524, 2383_2434,<br>2383_2524, 2434_2524 | V_VIII<br>V_V, VIII_VIII<br>VIII_V | 22°C, 24°C | 0 mg/L, 0.1 mg/L | 1, 2, 3 |
| Tri-culture<br>2375_2383_2434,<br>2375_2383_2524,<br>2375_2434_2524,<br>2383_2434_2524 | V_VIII_VIII<br>V_VIII_V<br>V_VIII_V<br>VIII_VIII_V | 22°C, 24°C | 0 mg/L, 0.1 mg/L | 1, 2, 3 |
| Tetra-culture<br>2375_2383_2434_2524 | V_VIII_VIII_V | 22°C, 24°C | 0 mg/L, 0.1 mg/L | 1, 2, 3 |

Table 1: Different experiment set-ups

# 3 Data Exploration and Handling

## 3.1 Missing and Invalid Data

Although we have a balanced experiment design, where each stratum, as identified by a strain combination, environmental conditions, and replication, should produce 252 cultures, in table 2 we notice that not all cultures were sampled during the 21-day experiment period. This is particularly apparent in figure 1, which shows the aggregated population, that is, the estimated number of cells in the culture, of each strain-combination, regardless of the environmental condition. In the figure, we clearly see some drop in the population of some strain combinations, particularly on the 13th day of measurement in the duo-culture experiments, and the 11th day in the tri-culture and tetra-culture experiments. However, further data exploration revealed that these drops are not indications of a decline in population, but rather of missing population measurements for some cultures.

In each sample, there could be hundreds and even thousands of cells remaining after it is diluted to allow the cytometer to do its readings. However, not all measurements provided were as expected, and thus the data needed to be cleaned, that is, removed of these erroneous readings. This part of the data preparation has been dealt with by the scientist who

| Culture type | Number of Samples | | Number of Cells | Population Size |
|---|---|---|---|---|
| | Actual | Expected | | |
| Mono-culture | 1008 | 1008 | 600 - 5633 | 108.5905 - 175150.5097 |
| Duo-culture | 1482 | 1512 | 525 - 8830 | 2258.5884 - 789144.1235 |
| Tri-culture | 971 | 1008 | 629 - 2432 | 1582.2040 - 196252.2118 |
| Tetra-culture | 243 | 252 | 680 - 2032 | 2216.1155 - 171255.3544 |

Table 2: Number of actual samples obtained and the range of number of cells clearly identified in each sample. The expected number of samples is calculated as the product of the number of strain-combinations, number of days, number of treatment, and number of replicates.

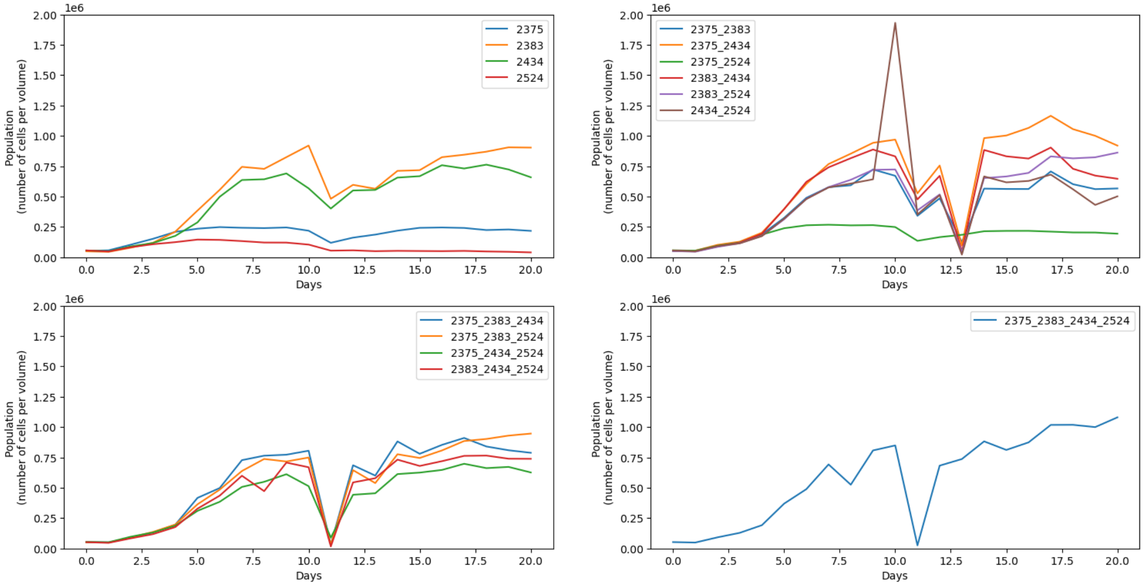

Figure 1: Aggregated population trend of each strain combinations during the 21-day experiment period in mono-cultures (upper-left), duo-cultures (upper-right), tri-cultures (lower-left), and tetra-culture (lower-right). The legend indicates the strains present in the multi-cultures.

performed the experiment, since he knows the possible range of expected values that the cytometer should output. These erroneous measurements could be caused by a debris or by two or more cells being exposed to the selected light frequencies at the same time.

## 3.2 Cytometer Output

Recall that each cell in each sample was exposed to eight different light frequencies, resulting in eight different measurements. In tables 3 and 4, we can see some summary statistics of cytometer outputs in all cultures, and it seems that the measurements at any given frequency for all cultures are close to each other. Note that the summaries were calculated after having concatenated all the data for each culture type. For instance, in the mono culture data, we have calculated the mean and the standard deviation of all four strains combined. Similarly, for the duo-culture, we have combined the data of all pairwise strain combination. This is to say that all summary statistics provided contained all four strains. As such, to see that the values are similar, even after taking into consideration the standard deviations, is comforting since this insinuates that combining the mono-strain cultures to make synthetic multi-culture data assimilates the organic multi-culture data. However, this hypothesis needs to be formally tested.

| | FSC | | SSC | | GRN.B | | YEL.B | |
|---|---|---|---|---|---|---|---|---|
| | mean | sd | mean | sd | mean | sd | mean | sd |
| mono-culture | 3.6496 | 0.9835 | 5.4343 | 0.7562 | 2.8203 | 0.5074 | 3.7881 | 2.0669 |
| duo-culture | 3.4322 | 0.8511 | 5.2863 | 0.7274 | 2.7567 | 0.5033 | 3.4686 | 2.0806 |
| tri-culture | 3.3888 | 0.8348 | 5.2746 | 0.7378 | 2.7710 | 0.4677 | 3.4391 | 2.0880 |
| tetra-culture | 3.3840 | 0.8251 | 5.2756 | 0.7386 | 2.7851 | 0.4486 | 3.5058 | 2.1005 |

Table 3: Summary statistics of the first four cytometer outputs

| | RED.B | | NIR.B | | RED.R | | NIR.R | |
|---|---|---|---|---|---|---|---|---|
| | mean | sd | mean | sd | mean | sd | mean | sd |
| mono-culture | 4.9090 | 1.1954 | 3.9657 | 1.0750 | 5.8822 | 0.8627 | 4.2902 | 0.7848 |
| duo-culture | 4.7139 | 1.2299 | 3.8061 | 1.1039 | 5.9273 | 0.9455 | 4.3187 | 0.8609 |
| tri-culture | 4.7674 | 1.2200 | 3.8546 | 1.0912 | 6.1303 | 0.8714 | 4.4950 | 0.7893 |
| tetra-culture | 4.8319 | 1.2258 | 3.9158 | 1.0904 | 6.2295 | 0.8163 | 4.5827 | 0.7354 |

Table 4: Summary statistics of the last four cytometer outputs

It is also worth noting that these cytometer outputs are correlated. In figure 2, we can clearly see how YEL.B is very correlated to NIR.B (0.88) and RED.B (0.90). While RED.B and NIR.B (0.94), and RED.R and NIR.R (0.99) are almost perfectly correlated. This correlation should be factored in somehow in any proceeding statistical modeling. In addition, the fact that some of these outputs are very closely correlated might have an impact among which cytometer outputs could be useful in predicting strain membership of the cells. Including, for example, both RED.R and NIR.R as predictors might be redundant since they are almost providing the same information to the predicting model.

In fact, after performing a principal component analysis, the first three principal components explain 85.65% of the variability in the data. And as seen in figure 3, the last three components explain very little of this variability.

### 3.2.1 Synthetic and Organic Multi-cultures

The main objective of this research is to classify the cells of the multi-culture samples to their respective strains in order to allocate the sample population proportional to the number of cell strains in each sample. Since this is going to be performed by making models trained in synthetic multi-culture data, that is, concatenating data from mono-culture sam-

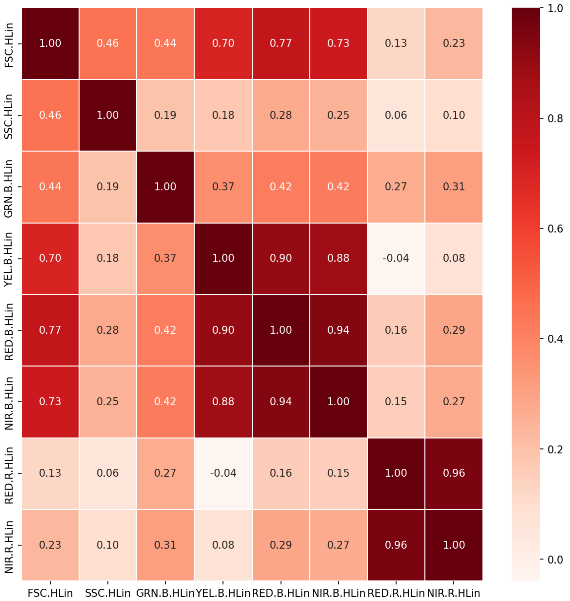

Figure 2: Pairwise correlation plot of the cytometer outputs of the mono-strain cultures

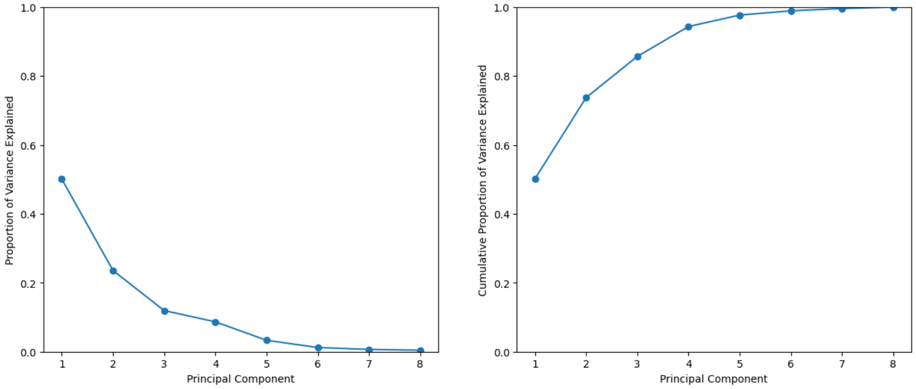

Figure 3: Explained variability of each principal component after PCA on the scaled cytometer outputs.

ples, we would want to see how different or similar these synthetic multi-cultures data are from those of the organic multi-cultures.

In figure 4, ignoring the sudden drop in the organic multi-culture data due to missing sample records, we can clearly see that the population curves of both multi-culture data, whether from organic multi-culture samples or synthetically created from mono-cultures, follow the same trend. The only visible difference between them is how the population of the synechococcus strains in the synthetic data seems to be larger than the population of the strains in the organic multi-cultures. This apparent difference in population could be attributed to the interaction among the different synechococcus strains in the organic multi-cultures, which clearly does not exist in the synthetic multi-cultures. It is only in the duo-culture of strains 2375 (V) and 2434 (VIII), and 2434 (VIII) and 2524 (V) where the population of both synthetic and organic multi-cultures seem to be coinciding.

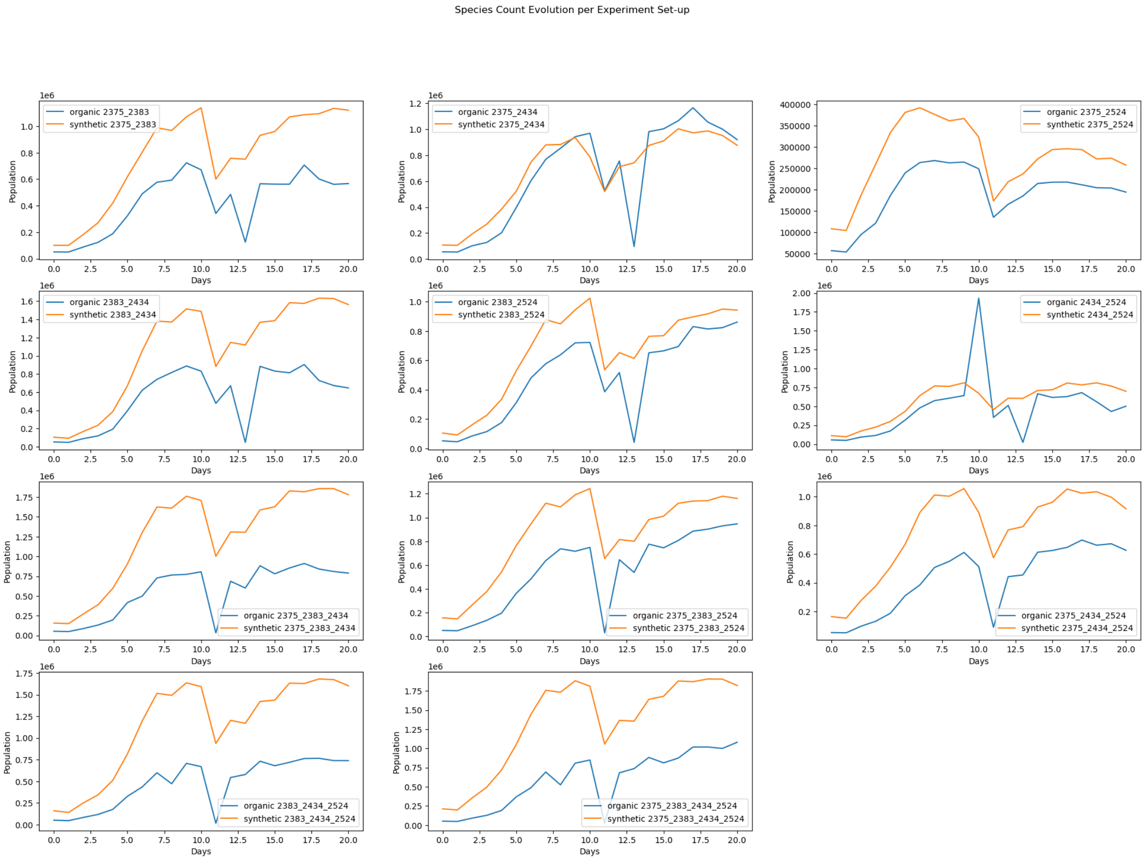

Figure 4: Population trend of synthetic multi-culture (orange) and organic multi-culture (blue) samples

In figure 5, we can see the cytometer outputs of cells from both organic and synthetic duo-cultures, where each plot represents the cytometer outputs when cells are exposed to a particular light frequency. Ideally, if there is little or no difference between the way cells from either duo-culture refract light, then in any plot, any boxplot pairs inside the rectangle should be the same. Although this seems to be true in the second (SSC) and third (GRN.B) plots for all duo-cultures, this does not seem to generalize in the remaining plots. In some cases, each pair seems relatively similar (NIR.R), in some cases, they are not (RED.R). And in some occasions, although the first pairs seem to resemble one another, the last two pairs are rather dissimilar (RED.B and NIR.B).

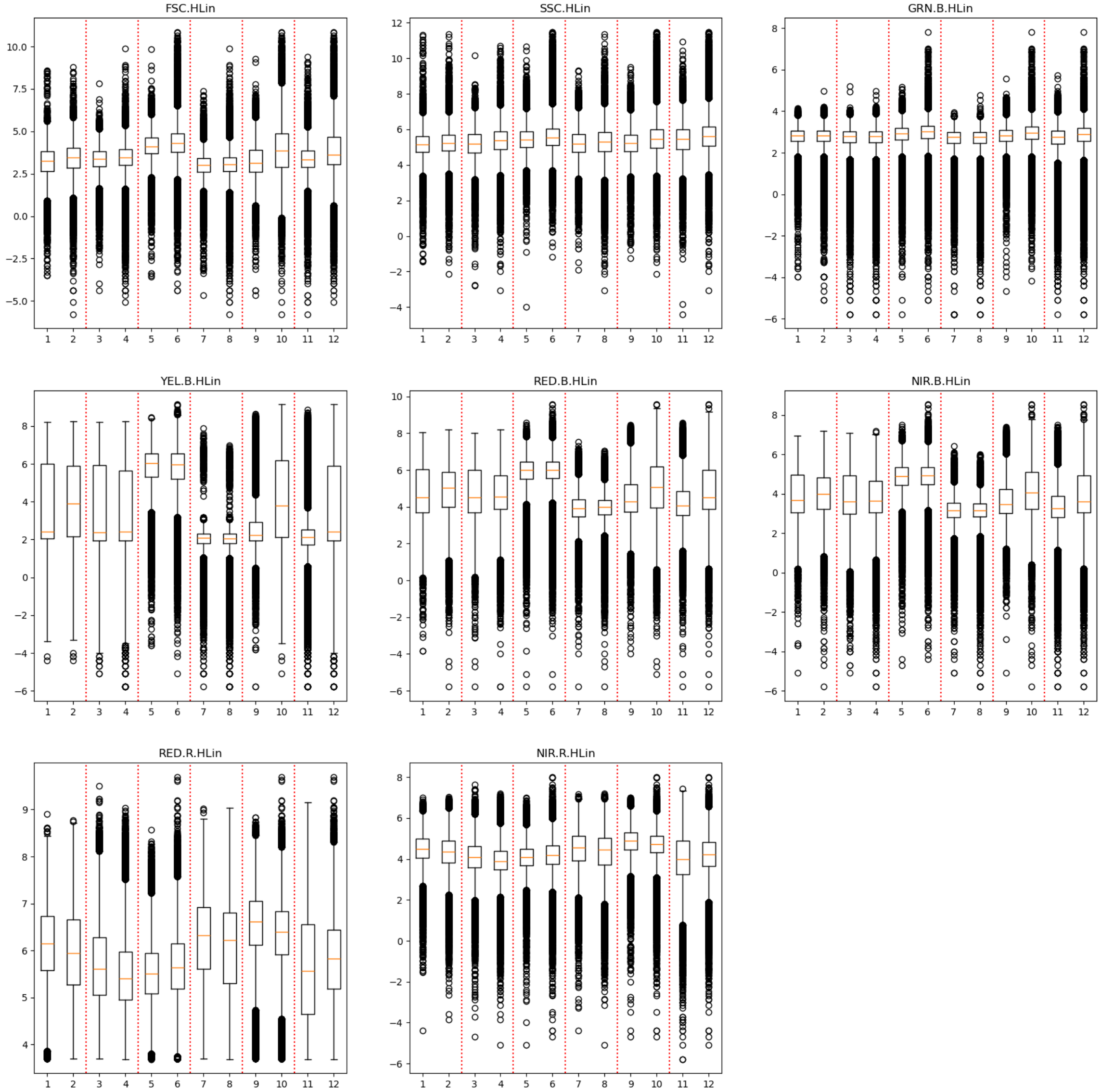

Figure 5: Cytometer outputs using different light frequencies, where each plot is for a specific frequency, of the organic duo-culture data (odd numbers on the x-axis) and of the synthetic duo-culture data (even numbers on the x-axis)

### 3.2.2 Outlier Detection

Although the data have been pre-cleaned, we still performed an outlier detection method for the possible existence of outliers in the mono-strain culture data. Given that we have a stratified experiment design, we have scanned for outliers for each strata by calculating the z-scores for each cytometer output. To remove the correlation that is present in the cytometer outputs, we first performed a pca transformation on the scaled cytometer outputs. We considered outliers those observations whose pca-transformed cytometer outputs are above the threshold. After using a threshold of 2.5, we only detected 17 observations out of the 1048906 observations, while this number drops to 3 when the threshold is 3.

## 3.3 Unsupervised Machine Learning

Aside from knowing the set of strains in cultivating the multi-strain cultures, we are in no possession of the exact strain membership of any individual cell. As such, if we focus on only the organic multi-culture data, and still try to distinguish or identify strain membership, a number of unsupervised machine learning can be explored. However, one big challenge in using such technique is interpreting what the resulting clusters mean. This means that being able to, for instance, successfully group the cells into four clusters, is no guarantee that these clusters will refer to the strain membership as this could mean other things as well. In our experiment in fact, this could refer to the condition in which the cultures have been grown, unless we apply unsupervised machine learning on data already separated by condition. However, this approach of applying unsupervised machine learning per condition, or to an extension, per strata, is inefficient since this produces various models.

To motivate our use of supervised learning on synthetic multi-cultures and using that to predict strain membership of organic multi-cultures, we will slightly look into two unsupervised machine learning techniques, namely principal component analysis and k-means clustering, and show how these might not be sufficient for our problem.

### 3.3.1 PCA

In figures 6 and 8, we can see how the cells from mono-strain cultures and cells from the tetra-culture are plotted in the first two principal dimensions after performing a pca reduction on the scaled predictors and unscaled predictors respectively. In the first set of plots in figure 6, we can notice how individual clusters of each strain seem to overlap and provide no clear separation between them. Although the first plot offers slight segregation between cells of different species, the second and third plots show that the clusters of cells of the same species are almost completely overlapping. This can also be visualized in figure 7. Furthermore, we can observe from the plot for the second principal component how strain 2434 seems to overlap with all three other strains. In terms of the multi-cultures, specifically the tetra-culture as shown in the fourth plot, we can say that it has the same shape as in the first plot, where the cells of all strains are plotted. This could suggest that if the task of separating the cells of synthetic multi-culture data into different clusters just by doing a dimension reduction seems impossible, then the same could be said for organic multi-cultures.

In the unscaled version of the pca reduction in figure 8, we can see clusters that are completely isolated. However, these clusters refer to the days of measurement and not to the four strains since there are a total of 21 clusters formed and that in each cluster, all four strains are present. Although pca could possibly be applied to a subset of the data with a specific measurement day, this would have produced 21 models in total.

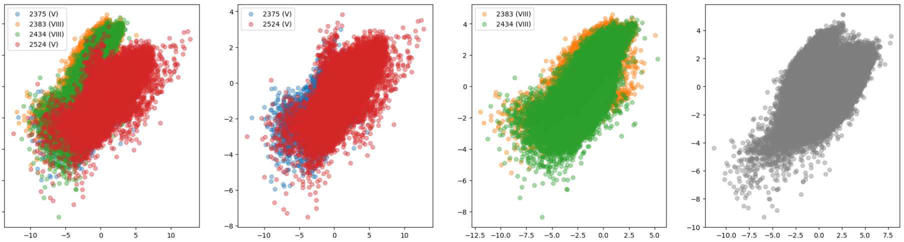

Figure 6: Separation in synthetic tetra-culture (first three images) and organic tetra-culture (last image) on the first two principal components after PCA on scaled features (cytometer outputs and dates) and encoded categorical feature (condition).

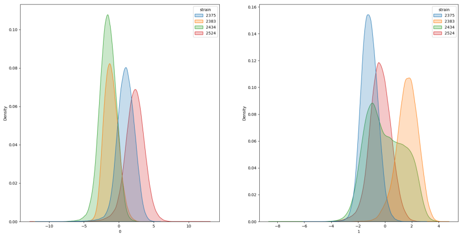

Figure 7: Density plot of the first (left) and the second (right) principal components

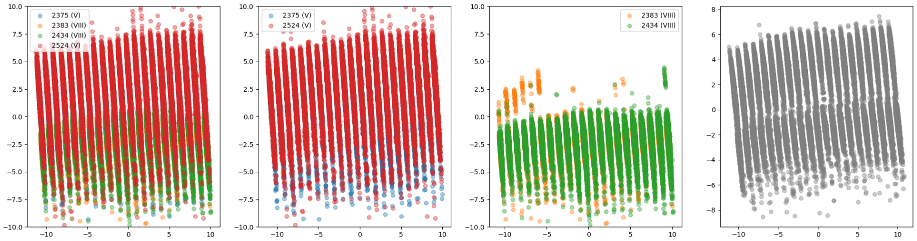

Figure 8: Separation in synthetic tetra-culture (first three images) and organic tetra-culture (last image) on the first two principal components after PCA on unscaled features (cytometer outputs and dates) and encoded categorical feature (condition).

### 3.3.2 K-Means

When using k-means, we need to indicate the number of expected clusters $k$ in the data. The k-means procedure initially assigns a random cluster to each observation. Then at each iteration, it calculates the centroid of all observations assigned to the same cluster. Each observation is then reassigned to the cluster whose centroid it is the closest. The procedure terminates once no observation is reassigned to a different cluster.

For this study, since the goal is to group in terms of strain membership, we set $k = 4$, and we group the cells in the synthetic tetra-culture data in four clusters. After that, we calculate the distances of each observation from the centroids of the first and second clusters.

This data transformation allows us to plot the observations in two dimensions.

In figures 9 and 10, we see the plots after applying k-means to scaled and unscaled predictors of the synthetic tetra-culture respectively. The left-most and middle plots are the same but with different coloring schemes. The left-most plots are colored based on their assigned groups as suggested by the k-means algorithm, while the middle plots are colored based on the strains of each observation. This clearly shows that the groups assigned using k-means do not correspond to the four strains present in the experiment. Furthermore, in the right-most plot of figure 10, we seem to observe a separation for the 21 days of measurement, and that the assigned groups correspond rather to the period of measurement, e.g., first 5 days, and not to the strains. This could be evidence of the inadequacy of unsupervised learning in identifying the strain of the cells in organic multi-cultures.

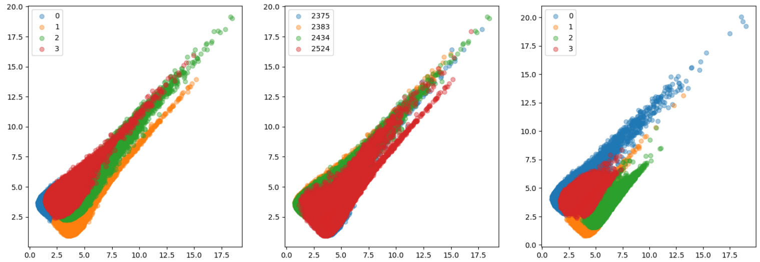

Figure 9: Clustering formed in synthetic tetra-culture as labeled by the predictions of a k-means classifier with four components (left) vs actual strain (center) and clustering formed in the organic tetra-culture (right) using scaled features (cytometer outputs and dates) and encoded categorical feature (condition).

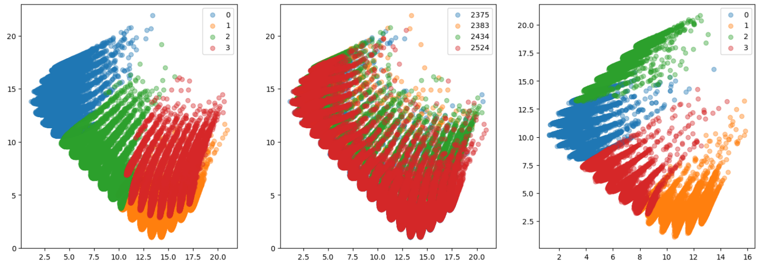

Figure 10: Clustering formed in sythetic tetra-culture as labeled by the predictions of a k-means classifier with four components (left) vs actual strain (center) and clustering formed in the organic tetra-culture (right) using unscaled features (cytometer outputs and dates) and encoded categorical feature (condition).

## 4 Multi-class classification

In contrast to binary classification problems where an observation is assigned to either of two existing classes, a multi-class classification involves assigning an observation into one of more than two classes. In this research, the classes are the different strains used in the experiments, and we would like to assign each cell in the samples to their corresponding strain.

There are two widespread approaches on these kinds of problems: the one-vs-rest and the one-vs-one.

## 4.1 One-vs-rest vs One-vs-one Models

In the one-vs-rest approach, if there are $k$ classes, we would create $k$ binary classifier models. Each model asks the question whether or not an observation belongs to class $i, 1 < i < k$. If the output is one-hot coded, that is, the output is encoded into $k$ columns, where the $i$th column is set to 1 if the observation belongs to class $i$, and all the other columns $j \neq i$ are set to 0, then each column $i$ would be the corresponding output for model $i$. During class prediction, an observation is assigned the class with the highest predicted probability.

In the one-vs-one approach, we would create a total of $\frac{k(k-1)}{2}$ binary classifier models. Each classifier corresponds to each pair of classes and models the probability of belonging to either one of the pair. Unlike in the first approach where it suffices to just encode the output to make the corresponding output columns for each model, in this approach, several datasets need to be created, each containing only the observations that belong to the pair of classes being modeled. During class prediction, the predicted probabilities for each class of all models are then averaged and an observation is assigned the class with the highest averaged predicted probability.

## 4.2 Base Classifiers and Models

### 4.2.1 Random Forests and Gradient-Boosted Trees

In theory, any existing binary classifiers can be used to build a multi-class classifier. However, we would only survey the performance of random forest classifiers and gradient-boosted trees because of their computing efficacy and their increased popularity in the past years. Among these two classifiers, we would select the one that results in higher classification accuracies.

A random forest classifier is a type of model ensembling and is composed of several decision trees. Each decision tree is built on bootstrapped training samples, wherein only $p < m$ predictors, which is a random sample of all $m$ predictors, are taken into account should a split be necessary [11].

Gradient-boosted classifiers are also an ensemble of several decision trees. However, unlike random forest classifiers, where the trees are built separately and independent of one another, the trees in gradient-boosted classifiers are built sequentially, where at each iteration, a tree models the remaining information that the previous trees were not able to capture. This way, the trees learn the patterns slowly and avoids overfitting [11].

### 4.2.2 N-way models

In the experiments performed, several multi-cultures were grown: duo-culture, tri-culture and tetra-culture. Each of this multi-strain culture could be treated in isolation and be modeled separately and independently of the other strain-combinations using the information from the mono-strain cultures.

Following this approach, we would have 6 pairwise models, 4 three-way model, and one four-way model. These models were created using the one-vs-rest approach, with the scaled cytometer outputs, day of measurement, and condition as predictors, and will be referred to as the base models. Each base model can provide predicted probabilities of strain membership to the strains that were used in training the model. For instance, a model capable of distinguishing strain 2375 from strain 2383 was trained on concatenated data from the mono-strain cultures of 2375 and of 2383. This model is capable of giving the probability of belonging to 2375 and to 2383 given a set of predictors. Although these models can already address the classification problem we are trying to solve, we will try several strategies that make use of these models to create a single model that hopefully will yield higher accuracy.

## 4.3 Classification Strategies

Although the base models could already be used on their own, we will create various models that use the predicted probabilities of belonging to a strain of these base models as new predictors in several ways. In some cases, all the predicted probabilities will be used, in some cases, only some of them will be utilized. In the event where the four-way and the two-way models are used, this could be seen as a way to combine the two different approaches in multi-class classification since it uses the one-vs-rest four-way model and the pairwise models used to build the predictions in the one-vs-one approach. Thus, instead of simply assigning an observation, in our case, a cell to a strain with the highest average predicted probability, we will let the model learn how to use each model's predicted probabilities in a more clever way. This could be considered as model ensembling, that is, we put together various models to create a final and definite model.

### 4.3.1 Strategy 0

To know how well the several strategies improve or deteriorate the classification performance, we will use the four-way model as our baseline.

### 4.3.2 Strategy I

In this strategy, all seven datasets used in building the base models will be used. This means that each strain of mono-culture data will be duplicated seven times - 3 times from training three pairwise models, another 3 times from the three-way models, and one time from the four-way model. However, each dataset has been scaled differently, that is, taking into account only the strains included in that particular dataset. Furthermore, aside from the predictors used in making the base models, four additional predictors are added which pertains to the predicted probability of belonging to each of the class. The predicted probabilities for each dataset will be provided by the base model which were trained using that dataset. And since not all strains were necessarily present in a particular dataset, the predicted probability for those strains will be set to zero. We can see how these datasets were concatenated to form the dataset used to train the classifier model for this strategy in figure 11.

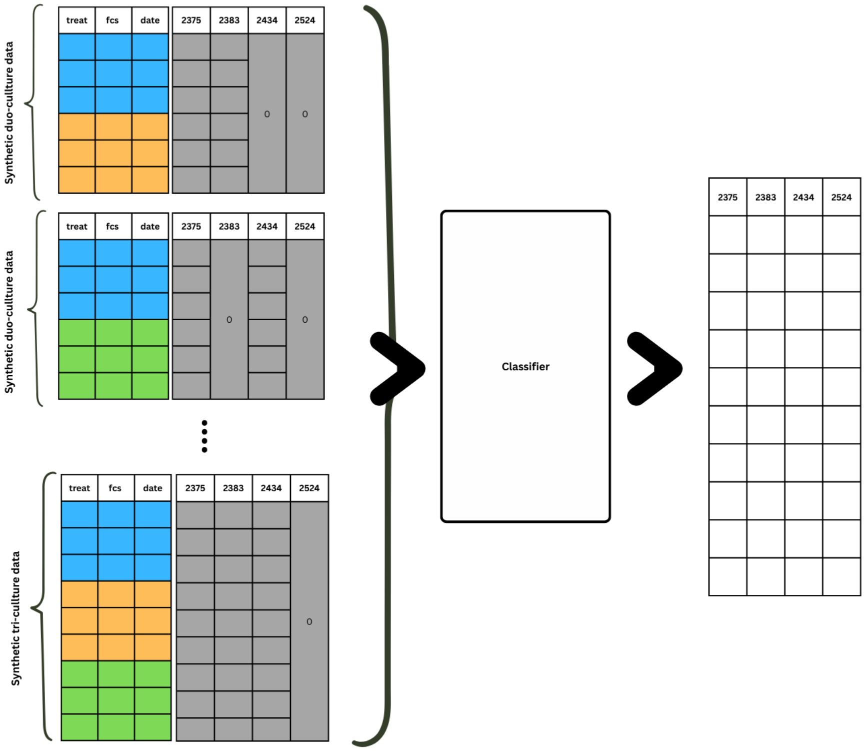

Figure 11: Strat I: Visual representation of how the data from the mono-strain cultures were concatenated and how the predicted probabilities from the base models (gray) were appended to the original predictors, which are treatment/condition, cytometer outputs, and day of measurements (colored) to form the new set of predictors.

### 4.3.3 Strategy II

Whereas the scaled datasets for building the base models were directly used to make the training data set for the first strategy, in this strategy the cytometer outputs of the concatenated mono-strain cultures were scaled taking into account all the strains. These formed the predictors for this strategy, including the day of measurement and environment conditions with the predicted probabilities from all the base models as additional predictors. Thus, there are 28 more predictors (6 pairwise models x 2, plus 4 three-way models x 3, plus 1 four-way model x 4) compared to the base models' predictors. This can be visualized in figure 12.

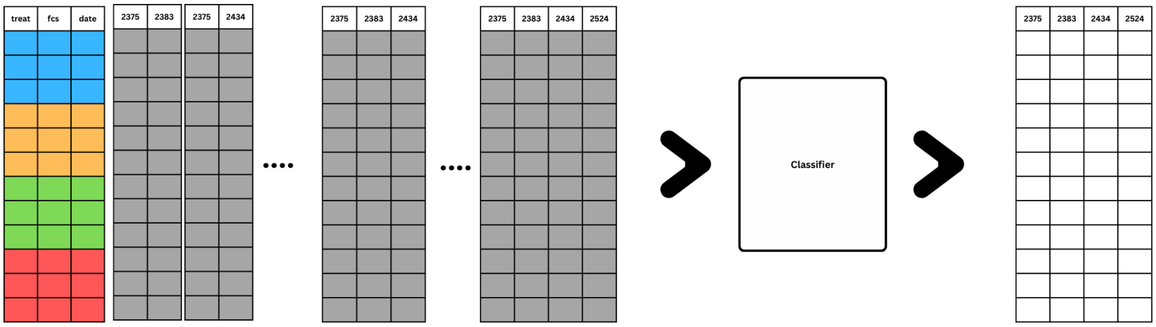

Figure 12: Strat II: Visual representation of how the data from the mono-strain cultures were concatenated and how the predicted probabilities from the base models (gray) were appended to the original predictors, which are treatment/condition, cytometer outputs, and day of measurements (colored) to form the new set of predictors.

### 4.3.4 Strategy III

In the previous strategies, we can use any kind of multi-class classifier, however, for this particular strategy we will make use of neural networks and will only make use of the predicted probabilities as predictors. In figure 13, we can see how each set of predicted probabilities produced by each base model are fed separately in an input layer, and then passed on to a block consisting of alternating normalization and dense layers. The resulting outputs of these independent blocks are then concatenated and fed to a classifier.

### 4.3.5 Strategy IV

The fourth strategy includes the cell species as part of the predictors and only uses the predicted probabilities of the two pairwise models that were trained with strains from the same species. In particular, these are the models trained with mono-strain cultures of 2375 and 2524, and 2383 and 2524. The idea is to leverage the fact that we know the specific species of a cell and that will help the model to decide which predicted probabilities it needs to give more bearing to. Now, although this is known in mono-strain cultures, this is also part of the problem in the organic multi-strain cultures. So to identify the species

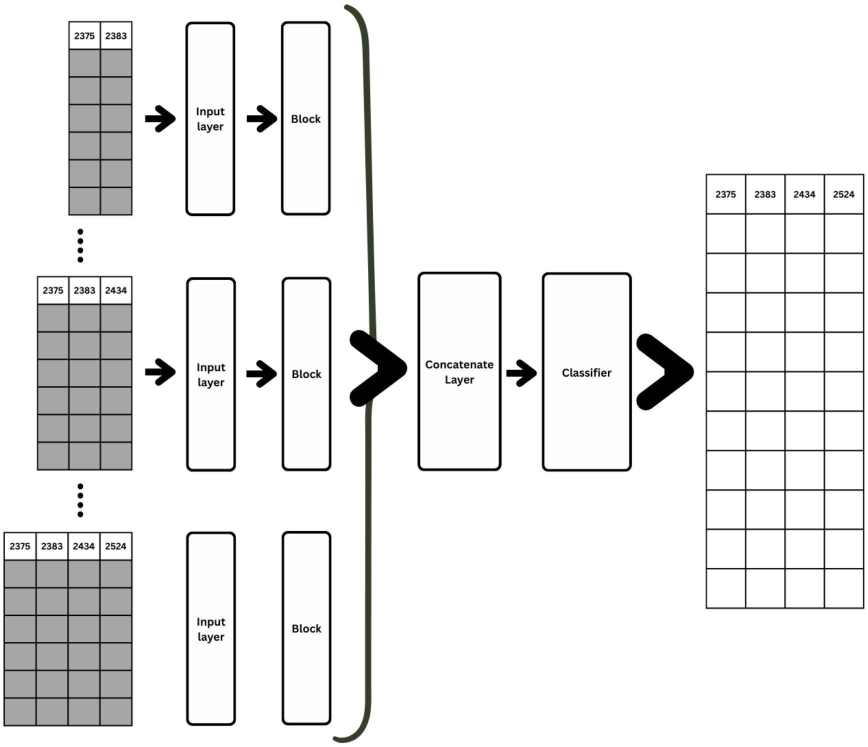

Figure 13: Strat III: Visual representation of how the predicted probabilities from the base models (gray) were concatenated to form the new set of predictors.

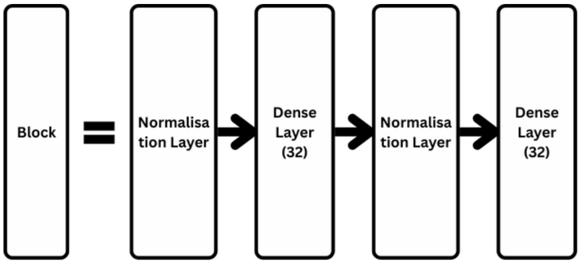

Figure 14: The block layer used in the neural network in strategy III.

to which a cell belongs in such cases, we will also make a binary classifier that determines the species of a particular cell using the concatenated mono-strain cultures, with the days of measurement, the conditions, and the scaled cytometer outputs as predictors. This can be visualized in figure 15.

### 4.3.6 Strategy V

In the last strategy, only the predicted probabilities of the pairwise models are included as new predictors, together with six semi-indicator variables, allowing values 0, 1, and 0.5. Each indicator variable represents each pairwise model, such that if a cell could make use of the predicted probabilities of a given model, then the indicator is set to 1, otherwise, if

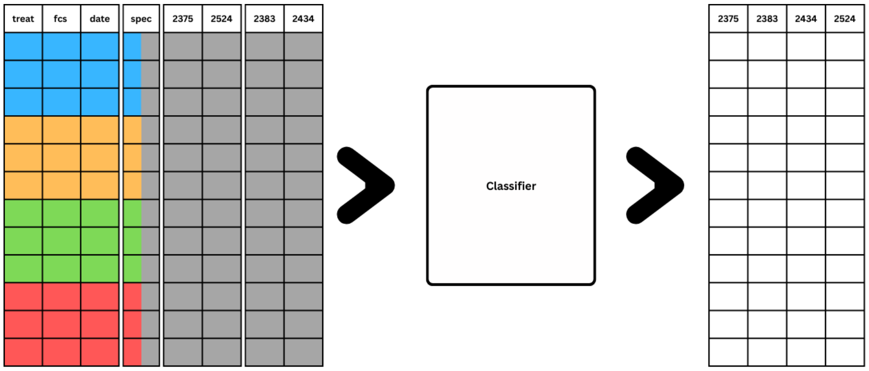

Figure 15: Strat IV: Visual representation of how the predicted probabilities from the two pairwise models (gray) were concatenated to form the new set of predictors including the species (colored, or true species for the mono-strain cultures, and gray, or predicted for the organic multi-strain cultures).

only a part of it is useful, it is 0.5, and 0 otherwise. For instance, a cell that could belong to strains 2375 and 2383 would have ones in the semi-indicator variable 2375-2383, and 0.5 for any other indicator variable that contains either 2375 or 2383, e.g., 2375-2434, 2375-2524, 2383-2434, and 2383-2524, and 0 for the remaining indicator variable 2434-2524.

## 4.4 Feature Selection

To investigate how each feature affects the base models' predictions, we will look into how important each feature is in building the model. In the case of random forest classifiers, the importance of each feature will be calculated based on what is known as permutation importance, that is, the difference between the prediction errors of trees created with the training dataset and trees created with the same dataset but with the feature randomly permuted among all observations [12]. In the case of gradient-boosted trees, the importance of each feature is calculated based on the average gain, that is, the reduction of training error, across all splits when a certain feature is used. Thus, the most important features are the most influential features that lead to more accurate predictions [13].

Such insights can help us eventually improve the base models' performance and, in turn, those of the various strategies presented earlier and gain understanding on which cytometer outputs, for instance, matter most and which do not or least affect the classification and the strain membership probability prediction.

## 4.5 Metrics and Validation Set

In order to assess the performance of the models, the accuracy of correctly classifying the strains will be calculated. Since we would be performing a supervised machine learning due to the fact that we know the strains to which the cells in the synthetic multi-culture data belong, accuracy can be easily calculated for the training, validation, and test datasets. However, this cannot be performed in the organic multi-culture datasets. Instead, we will measure the 'correctness' of the different strategies by getting the percentage of misclassified cells, that is, cells assigned to strains that were not grown in a given multi-culture. For instance, given that we are classifying the cells that were cultivated using strains 2375 and 2383, a misclassification in this case would be assigning any cell from this sample to strains other than those two, e.g., 2434 or 2524.

Now to ensure that there is no data leakage, especially since we are training new models to build our various strategies using the output from previously trained base models, we have divided each mono-strain culture into three separate sets using stratified random sampling: training set (60%), validation set (20%) and test set (20%). All trainings, be it for the base models or for the latter models, are only performed using the selected training set, and so on. The validation set is used to improve the performance of the models on unseen data by adjusting the parameters of the models accordingly. This ensures that the models do not learn patterns specific only to the training data set. Lastly, the test set will give us an idea of how the models behave on unseen data that have not been used in any sense in building or refining the model. Note that since we are only training using the mono-culture data, and not the data from the multi-culture, the latter need not be divided into different sets and are treated as test sets as a whole, to which the true strains are unavailable.

# 5 Statistical Analysis

## 5.1 Modeling the Cytometer Outputs

The premise of using cytometer outputs as the main predictors of strain membership assumes that cells from different strains differ in the way they refract various light frequencies, resulting in different readings provided by the cytometer.

In the experiment, whenever this was measured, each cell was exposed to eight different light frequencies, and the resulting cytometer readings were correlated. Thus, to capture this correlation, we shall use a multivariate approach to understand how cell strain, environmental condition or treatment, and the number of days the culture has been kept influence the cytometer readings. Furthermore, it is worth mentioning that cells from the same sample could also be correlated and should be taken into account. However, due to the large number of data and difficulty of fitting a model that incorporates sample random effects, this is unfortunately dropped.

Equation 1 gives the model formulation of the cytometer outputs that contains only the main fixed effects. However, models will be attempted to incorporate the effects of the two-way interaction of the different frequency on the day of measurement, population size, strains, and treatment.

$$
\begin{aligned}
\mathrm{fcs}_{ijk} = \beta_0 &+ \beta_1 \mathrm{days}_k + \beta_2 \mathrm{pop}_k + \beta_3 \mathbb{1}_{\mathrm{repl}_k=2} + \beta_4 \mathbb{1}_{\mathrm{repl}_k=3} \\
&+ \beta_5 \mathbb{1}_{\mathrm{treat}_k=A} + \beta_6 \mathbb{1}_{\mathrm{treat}_k=T} + \beta_7 \mathbb{1}_{\mathrm{treat}_k=AT} \\
&+ \beta_8 \mathbb{1}_{\mathrm{strain}_k=2383} + \beta_9 \mathbb{1}_{\mathrm{strain}_k=2434} + \beta_{10} \mathbb{1}_{\mathrm{strain}_k=2524} \\
&+ \beta_{11} \mathbb{1}_{\mathrm{freq}_{ijk}=SSC} + \beta_{12} \mathbb{1}_{\mathrm{freq}_{ijk}=GRN.B} + \beta_{13} \mathbb{1}_{\mathrm{freq}_{ijk}=YEL.B} + \beta_{14} \mathbb{1}_{\mathrm{freq}_{ijk}=RED.B} \\
&+ \beta_{15} \mathbb{1}_{\mathrm{freq}_{ijk}=NIR.B} + \beta_{16} \mathbb{1}_{\mathrm{freq}_{ijk}=RED.R} + \beta_{17} \mathbb{1}_{\mathrm{freq}_{ijk}=NIR.R} + \epsilon_{ijk}
\end{aligned}
\tag{1}
$$

where:

- $\mathbf{fcs}_{jk}$ is an 8-element vector containing the cytometer outputs of cell $j$ of sample $k$
- $\mathrm{days}_k$ is the day when sample k is sampled and measured
- $\mathrm{pop}_k$ is the estimated number of cells in the culture
- $\mathbb{1}_{\mathrm{repl}_k=\{2,3\}}$ indicates whether the sample is obtained in from the second or third replication
- $\mathbb{1}_{\mathrm{treat}_k=\{A,T,AT\}}$ indicates whether the environment contains atrazine (A), maintained on a temperature of 24°C (T), or both (AT)
- $\mathbb{1}_{\mathrm{strain}_k=\{2383,2434,2524\}}$ indicates the strain cultivated where sample $j$ is acquired
- $\mathbb{1}_{\mathrm{freq}_{ijk}=\{SSC,GRN.B,YEL.B,RED.B,NIR.B,RED.R,NIR.R\}}$ indicates with which frequency the cell $j$ of sample $k$ was exposed
- $\boldsymbol{\epsilon_{jk}} \sim MVN(0, \Sigma)$

The $\beta$'s are the parameter estimates of the fixed effects and the vector of error terms follows a multivariate normal distribution, with mean 0, and an unstructured variance-covariance matrix. Lastly, we control the the false discovery rate (fdr) for multiple hypotheses testing using the Benjamini-Hochberg method of adjusting of p-values.

## 5.2 Synthetic Multi-culture vs Organic Multi-culture

Since we will be creating models trained on concatenated mono-culture data, and we will be using these models to predict strain membership in actual multi-culture data, it is important to know whether the cytometer outputs of these two different cultures are comparable. Should the analysis show that there is no significant difference between the cytometer outputs of the synthetic and the organic multi-culture, or should this difference be relatively small, then we will gain confidence in predictions to be made by the model. Otherwise, these predictions should be considered with caution.

The model formulation given in equation 2 is similar to the model formulation in the first statistical analysis, except for a small change. Instead of strains, we will use an indicator variable that indicates whether a cell is from a synthetic dataset or not.

$$\begin{aligned}\text{fcs}_{ijk} &= \beta_0 + \beta_1 \text{days}_k + \beta_2 \text{pop}_k + \beta_3 \mathbb{1}_{\text{repl}_k=2} + \beta_4 \mathbb{1}_{\text{repl}_k=3} \\ &+ \beta_5 \mathbb{1}_{\text{treat}_k=A} + \beta_6 \mathbb{1}_{\text{treat}_k=T} + \beta_7 \mathbb{1}_{\text{treat}_k=AT} + \beta_8 \mathbb{1}_{\text{culture}_k=synth} \\ &+ \beta_9 \mathbb{1}_{\text{freq}_{ijk}=SSC} + \beta_{10} \mathbb{1}_{\text{freq}_{ijk}=GRN.B} + \beta_{11} \mathbb{1}_{\text{freq}_{ijk}=YEL.B} + \beta_{12} \mathbb{1}_{\text{freq}_{ijk}=RED.B} \\ &+ \beta_{13} \mathbb{1}_{\text{freq}_{ijk}=NIR.B} + \beta_{14} \mathbb{1}_{\text{freq}_{ijk}=RED.R} + \beta_{15} \mathbb{1}_{\text{freq}_{ijk}=NIR.R} + \epsilon_{ijk}\end{aligned} \tag{2}$$

where:

- $\mathbb{1}_{\text{culture}_k=synth}$ indicates whether sample $k$ is from a synthetic multi-culture

We also use the Benjamini-Hochberg procedure to adjust the p-values of the aforementioned indicator variable.

### 5.3 True Strains vs Predicted Strains

Although we can assess the accuracy of the models trained using synthetic multi-culture data, we would not be able to know how the models perform in the organic multi-culture data. This is because the true strain memberships of the cells are only known up to the strains cultivated together in a particular culture, but could never be reduced to any single strain. For this reason, to emulate the assessment whether the model correctly classifies a cell into its true strain, the cytometer outputs of all cells in the organic multi-culture data predicted to belong to a particular strain are compared with the cytometer outputs of cells belonging to the mono-culture of that strain. If indeed a model can discriminate cells and correctly predict their strain membership, then we should not see any significant difference from these cytometer outputs, or that such difference is relatively small.

We can see how the model is formulated in equation 3. This is similar to the first two model formulations, but instead includes a variable to indicate whether the sample is from a mono-strain culture or a multi-strain culture.

$$\begin{aligned}\text{fcs}_{ijk} &= \beta_0 + \beta_1 \text{days}_k + \beta_2 \text{pop}_k + \beta_3 \mathbb{1}_{\text{repl}_k=2} + \beta_4 \mathbb{1}_{\text{repl}_k=3} \\ &+ \beta_5 \mathbb{1}_{\text{treat}_k=A} + \beta_6 \mathbb{1}_{\text{treat}_k=T} + \beta_7 \mathbb{1}_{\text{treat}_k=AT} + \beta_8 \mathbb{1}_{\text{cult}_i=multi\_culture} \\ &+ \beta_9 \mathbb{1}_{\text{freq}_{ijk}=SSC} + \beta_{10} \mathbb{1}_{\text{freq}_{ijk}=GRN.B} + \beta_{11} \mathbb{1}_{\text{freq}_{ijk}=YEL.B} + \beta_{12} \mathbb{1}_{\text{freq}_{ijk}=RED.B} \\ &+ \beta_{13} \mathbb{1}_{\text{freq}_{ijk}=NIR.B} + \beta_{14} \mathbb{1}_{\text{freq}_{ijk}=RED.R} + \beta_{15} \mathbb{1}_{\text{freq}_{ijk}=NIR.R} + \epsilon_{ijk}\end{aligned} \tag{3}$$

where:

- $\mathbb{1}_{\text{cult}_k = multi_culture}$ indicates whether sample is from a multi-strain culture

Similarly to the previous analyses, fdr will be controlled using the Benjamini-Hochberg procedure.

### 5.4 Probability Prediction of One-vs-one Models

In some strategies, in particular the second and the third, the predicted probabilities of all the models were used as new features naively. Although these models are especially trained to predict strain membership of only a subset of all the strains, except for the four-way model, which clearly predicts all four, these models are used even for cells which we are certain to not belong to the strains with which these models were trained and are capable of giving probability predictions. For instance, we used the model for strains 2375 (V) and 2383 (VIII) on strains 2434 (VIII) and 2524 (V), which would then probably predict the 2434 strains to be 2383 and 2524 to be 2375 just because they are of the same species.

This analysis aims to know whether the predicted probabilities of a strain using a model that was trained with it will have the same predicted probabilities when a model that was trained with another strain but of the same species is used instead. This will allow us to gain insight on why certain strategies might work, and some might not. This analysis, which will be of the form of a paired t-test, will only be performed with the four pairwise models that were trained with strains of different species. A paired-t test is sufficient and ideal in this scenario, since the predicted probabilities for any two models will have the same covariates.

## 6 Results and Discussion

### 6.1 Base Models

We have summarized the accuracies of the three sets of base models trained using synthetic multi-culture data in table 5. All base models used the cytometer outputs, as well as the stratification variables, condition as defined by temperature and the presence or absence of atrazine, and the measurement days, as predictors. In the third set of models however, the culture population was also included as predictor.

A striking observation is how the training accuracies of the base models using random forests yield almost perfect scores of 100%. Nonetheless this does not translate well to test accuracies where, although still high, some dropped to around 91-92% as in the case of models used for the duo-culture (2383, 2434) and the tetra-culture. This suggests overfitting, that is, the random forests tried so hard to fit the training data set that they do not generalize well in other datasets.

Comparing random forests' training accuracies with those of the models built with gradient boosting, we clearly see how only the models for the duo-cultures (2375, 2383), (2375, 2434), (2383, 2524) and (2434, 2524) obtained accuracies near 100%. These models are trained on duo-culture data with strains belonging to different species. However, in cases where the training data set contained two strains of the same species, the gradient boosting trees seemed to have a little bit of difficulty. For instance, we see an accuracy of only 93% for the duo-culture (2383, 2434), strains belonging to species VIII. In addition, in contrast to the random forests, the models built with gradient boosting trees seem to generalize way better, as evidenced by the test accuracies being only slightly inferior to the training accuracies.

Lastly, we obtain models whose performance in both the training set and the test set is almost 100% after adding the culture population as a predictor. This suggests that we can be sure of the predictions made by this set of models. However, after further investigation and as illustrated in figure 18, the trend of the resulting population allocation to various strains of three selected organic multi-culture does not follow the trend of the strain population in the synthetic multi-cultures. Furthermore, the curves are rather zigzagged and overlapping, which can be considered indications of having wrongly allocated the multi-culture's total population. Although these models have very promising test accuracies, they did not seem to have performed well in the organic multi-culture data. This could be partly explained by the fact that a strain's population in a mono-culture is quite different from its population in a multi-culture set-up, where competition for resources among other factors could affect population growth. We have observed this earlier in section 3.2.1, figure 4.

| Strain | Random Forest | | XGBoost | | XGBoost with population | |
|---|---|---|---|---|---|---|
| Combination | Train Acc. | Test Acc. | Train Acc. | Test Acc. | Train Acc. | Test Acc. |
| 2375, 2383 | 1.0000 | 0.9985 | 0.9998 | 0.9984 | 1.0000 | 0.9996 |
| 2375, 2434 | 1.0000 | 0.9968 | 0.9990 | 0.9972 | 1.0000 | 0.9998 |
| 2375, 2524 | 1.0000 | 0.9337 | 0.9469 | 0.9398 | 0.9999 | 0.9995 |
| 2383, 2434 | 1.0000 | 0.9185 | 0.9302 | 0.9215 | 0.9983 | 0.9980 |
| 2383, 2524 | 1.0000 | 0.9983 | 0.9998 | 0.9985 | 1.0000 | 0.9995 |
| 2434, 2524 | 1.0000 | 0.9967 | 0.9986 | 0.9970 | 1.0000 | 0.9999 |
| 2375, 2383, 2434 | 1.0000 | 0.9411 | 0.9506 | 0.9448 | 0.9974 | 0.9964 |
| 2375, 2383, 2524 | 1.0000 | 0.9532 | 0.9631 | 0.9578 | 0.9999 | 0.9996 |
| 2375, 2434, 2524 | 1.0000 | 0.9570 | 0.9658 | 0.9612 | 1.0000 | 0.9998 |
| 2383, 2434, 2524 | 1.0000 | 0.9400 | 0.9494 | 0.9437 | 0.9982 | 0.9975 |
| 2375, 2383, 2434, 2524 | 1.0000 | 0.9220 | 0.9344 | 0.9281 | 0.9975 | 0.9967 |

Table 5: Training and test accuracy of the base models trained and tested using synthetic multi-strain cultures.

We also investigated which strains are predicted more accurately for the sets of base models using random forests and gradient boosting. The results are summarized in Tables 6 - 9, where the rows indicate the true strains and the columns indicate the strains to which a cell is predicted. For example, in the first confusion matrix in table 6, 50943 cells of strain 2375 were correctly identified as 2375, while 66 cells were incorrectly identified as 2383. The results agree with previous observations that the models were able to discriminate cells very accurately, provided they belong to different species. However, once cells belong to the same species, discrimination becomes more difficult. For instance, in the case of the random forest model for duo-strain cultures containing 2375 (V), the model only misclassified 66 and 326 cells when it was trained alongside 2383 (VIII) and 2434 (VIII) respectively. This is in contrast to 2676 cells when discriminated against cells of strain 2524 (V). It is also worth noting that it seems that there is a higher chance of misclassifying a cell of strain 2375 or 2524 to strain 2434 than to strain 2383. For example, in the first two confusion matrices in table 8 for gradient boosting, 244 cells of strain 2375 have been mislabeled as strain 2434 in contrast to 68 cells as strain 2383. This suggests that the models seem to identify the features of strain 2434 as more similar to those of strain 2375 and 2383. This is not surprising given that we have observed how the density plot of strain 2343 intersects more with the density plots of 2375 and 2524 in subsection 3.3.1, figure 7.

This is also observed when cells of strain 2383 or 2434 are misclassified to strains of the other species. However, unlike in the former case, the disparity between having misclassified to strain 2375 or to 2524 is smaller. This is portrayed better in tables 7 and 9. In particular, in the bottom left confusion matrix of both tables, in the case of random forests, 40 cells of strain 2434 were misclassified to 2375 and 39 cells to 2524, and in the case of gradient boosting, these numbers were 53 and 77 respectively. This suggests that when the models fail to correctly classify a cell belonging to species VIII, they almost indiscriminately assign them to either strain of species V.

| | 2375 | 2383 |
|---|---|---|
| 2375 | 50943 | 66 |
| 2383 | 82 | 44601 |

| | 2375 | 2434 |
|---|---|---|
| 2375 | 50683 | 326 |
| 2434 | 53 | 66302 |

| | 2375 | 2524 |
|---|---|---|
| 2375 | 48333 | 2676 |
| 2524 | 3896 | 44247 |

| | 2383 | 2434 |
|---|---|---|
| 2383 | 40938 | 3745 |
| 2434 | 5301 | 61054 |

| | 2383 | 2524 |
|---|---|---|
| 2383 | 44604 | 79 |
| 2524 | 76 | 48067 |

| | 2434 | 2524 |
|---|---|---|
| 2434 | 66302 | 53 |
| 2524 | 320 | 47823 |

Table 6: Confusion matrices of random forest models trained using synthetic duo-cultures on the test data.

| | 2375 | 2383 | 2434 |
|---|---|---|---|
| 2375 | 50647 | 8 | 354 |
| 2383 | 54 | 40867 | 3762 |
| 2434 | 47 | 5324 | 60984 |

| | 2375 | 2383 | 2524 |
|---|---|---|---|
| 2375 | 48325 | 57 | 2627 |
| 2383 | 70 | 44575 | 38 |
| 2524 | 3886 | 56 | 44201 |

| | 2375 | 2434 | 2524 |
|---|---|---|---|
| 2375 | 48142 | 332 | 2535 |
| 2434 | 40 | 66276 | 39 |
| 2524 | 3869 | 304 | 43970 |

| | 2383 | 2434 | 2524 |
|---|---|---|---|
| 2383 | 40841 | 3792 | 50 |
| 2434 | 5333 | 60983 | 39 |
| 2524 | 17 | 314 | 47812 |

Table 7: Confusion matrices of random forest models trained using synthetic tri-cultures on the test data.

| | 2375 | 2383 |
|---|---|---|
| 2375 | 50941 | 68 |
| 2383 | 88 | 44595 |

| | 2375 | 2434 |
|---|---|---|
| 2375 | 50765 | 244 |
| 2434 | 79 | 66276 |

| | 2375 | 2524 |
|---|---|---|
| 2375 | 48553 | 2456 |
| 2524 | 3517 | 44626 |

| | 2383 | 2434 |
|---|---|---|
| 2383 | 40849 | 3834 |
| 2434 | 4877 | 61478 |

| | 2383 | 2524 |
|---|---|---|
| 2383 | 44611 | 72 |
| 2524 | 68 | 48075 |

| | 2434 | 2524 |
|---|---|---|
| 2434 | 66254 | 101 |
| 2524 | 246 | 47897 |

Table 8: Confusion matrices of xgboost models trained using synthetic duo-cultures on the test data.

| | 2375 | 2383 | 2434 |
|---|---|---|---|
| 2375 | 50742 | 21 | 246 |
| 2383 | 62 | 40885 | 3736 |
| 2434 | 64 | 4821 | 61470 |

| | 2375 | 2383 | 2524 |
|---|---|---|---|
| 2375 | 48553 | 58 | 2398 |
| 2383 | 72 | 44576 | 35 |
| 2524 | 3457 | 46 | 44640 |

| | 2375 | 2434 | 2524 |
|---|---|---|---|
| 2375 | 48380 | 240 | 2389 |
| 2434 | 53 | 66225 | 77 |
| 2524 | 3444 | 218 | 44481 |

| | 2383 | 2434 | 2524 |
|---|---|---|---|
| 2383 | 40909 | 3722 | 52 |
| 2434 | 4842 | 61428 | 85 |
| 2524 | 20 | 243 | 47880 |

Table 9: Confusion matrices of xgboost models trained using synthetic tri-cultures on the test data.

In figures 16 and 17 we see a comparison of the population trends of various strains in selected synthetic multi-cultures, and the trend of the allocated populations in the corresponding organic multi-cultures. It is quite evident that the trends observed in the synthetic data have been propagated to the organic data, where they mostly only differ in the magnitude of the population, as there are relatively fewer cells in the organic multi-cultures. The almost absence of the zigzag pattern and curves crossing each other observed when the population was included as a predictor is reassuring, as this shows inconsistencies with allocating the culture population to various strains, and by extension, inconsistencies in the strain membership prediction. In addition, we also gain confidence in our base models, both with random forests and gradient-boosted trees, as the trends of the allocated populations are generally similar and seemingly identical.

We also looked at the importance of the different predictors in each base model. This is illustrated in the rightmost plots of the given figures. It can be observed that in situations where we only consider cells of different species, for instance, strains 2375 (V) and 2434 (VIII), the most important and very dominating predictor is the YEL.B cytometer measurements. However, once we start training models on data that contain strains of the same species, the other cytometer measurements gain importance. It is worth noting that in the random forest models for strains 2375 and 2524, and strains 2383 and 2434, the permutation importance scores of FSC and RED.R, and RED.R respectively, were the highest. This is in contrast with the xgboost models whose accuracy is greatly influenced by FSC and RED.R respectively. Nevertheless, in both cases the YEL.B cytometer measurements are deemed less important.

Looking deeper at the importance of the cytometer outputs as predictors for the random forest models in tri-cultures and tetra-cultures, we notice that GRN.B, NIR.B and NIR.R seem to be relatively irrelevant. Although this is not surprising for NIR.B, which is highly correlated with RED.B (0.94) and for NIR.R, which is highly correlated to RED.R (0.96), GRN.B does not have any other frequency with which it is highly correlated. However, since the models considered RED.B and RED.R, both of which GRN.B has a correlation of 0.42, very important, GRN.B bears very little importance in the models. However, these observations do not hold for the xgboost models where YEL.B still dominated and is the most important predictor that lead to more accurate predictions.

Lastly, while we have included the measurement days and the growth environment conditions of the cultures, these seems to have little importance in both sets of base models, except for the models for duo-culture strains with the same species.

## 6.2 Strategies' Performance

From the previous section, we have seen how the performance of the base models constructed using xgboost classifiers are superior than those constructed using random forests. For this reason, we have created the models for our five strategies using the xgboost base models.

After generating the predicted probabilities of the training, validation, and as well as the organic multi-culture datasets using the base models, we have created several datasets in order to build the models for the various strategies. The resulting accuracies of these new models are listed in table 10. In this table, we notice how the validation and the test accuracies are not far off from the training accuracy, which indicates that the model generalize well, that is, they did not learn patterns only specific to the training datasets. This is in addition to the fact that the models perform quite well, with training accuracies ranging from 93.44% to 96.70% and test accuracies ranging from 92.81% to 95.78%. However, these values are inferior to the average accuracies of the set of base models, which are 96.71% and 96.25%, respectively.

It can also be stated that the base strategy, strategy 0, has accuracies comparable to those of strategies II, III and IV. These strategies, however, are a little less accurate than strategies I and V, with a difference ranging from 1.38% to 3.26%. The reason for this slightly better performance is how these two strategies used information on which possible strains the cells could belong. In strategy I, for instance, we put a predicted probability of 0.00 to strains that were clearly not part of a given multi-culture. A cell coming from a duo-culture of strains 2375 and 2343 would have predicted probabilities for 2375 and 2343 generated by the base model for those two strains, while the predicted probabilities of 2383 and 2524 will be 0.00. In so doing, we leak useful information that helps the model predict strain membership more accurately, but not totally leaking the exact strain to which such cell belongs. And in strategy V, where only the generated predicted probabilities of the six duo-culture models are use, this information on possible strain membership is leaked by the additional six semi-indicator predictors, indicating whether a base model's predicted probabilities should be used fully (1), partially (0.5) or not at all (0). These pieces of information are not available to strategies II, III, and IV, where the predicted probabilities from all base models are used naively, albeit with equal bearing.

| Strategy | Train Accuracy | Validation Accuracy | Test Accuracy |
|---|---|---|---|
| Strategy 0 | 0.9344 | 0.9272 | 0.9281 |
| Strategy I | 0.9605 | 0.9548 | 0.9551 |
| Strategy II | 0.9467 | 0.9277 | 0.9287 |
| Strategy III | 0.9374 | 0.9270 | 0.9279 |
| Strategy IV | 0.9437 | 0.9294 | 0.9305 |
| Strategy V | 0.9670 | 0.9573 | 0.9578 |

Table 10: Accuracy of different data sets during the different phases of building the models of various proposed strategies.

Aside from looking at the accuracies obtained on the synthetic multi-cultures, we also looked into how to assess the strategies' performance on the actual multi-strain cultures. Since these models actually classify a cell to any of the four strains, as opposed to the base models, we can quantify how these models missclassify an individual cell's strain membership, that is, the models predicting that the cell is of a strain not present in the multi-culture from which it is sampled.

In tables 11 and 12, we list the proportion of misclassified population in each organic multi-culture data. It should be noted that strategies 0, II, III and IV, and strategies I and V, still yielded comparable results, with the latter group having much less misclassified cells. This is expected since these last two strategies have an advantage over the strategies in the former group. Of the 10 multi-strain cultures, strategy I outperformed strategy V in 7 of them, with an average misclassification of 0.2710% versus 0.3720% of strategy V. This contrasts with the misclassification errors in strategies 0, II, III and IV, which are 7.34%, 7.40%, 7.43%, and 7.24% respectively. Note that for strategy IV, there is no misclassification in the duo-culture data when the strains involved are of different species, which suggests that the model was able to discriminate cells into their corresponding species.

| Strategy | Strain Combination | | | | | |
|---|---|---|---|---|---|---|
| | 2375, 2383 | 2375, 2434 | 2375, 2524 | 2383, 2434 | 2383, 2524 | 2434, 2524 |
| Strategy 0 | 0.1172 | 0.1509 | 0.0047 | 0.0133 | 0.1122 | 0.1756 |
| Strategy I | 0.0028 | 0.0094 | 0.0015 | 0.0010 | 0.0030 | 0.0018 |
| Strategy II | 0.1190 | 0.1506 | 0.0045 | 0.0148 | 0.1137 | 0.1776 |
| Strategy III | 0.1077 | 0.1582 | 0.0044 | 0.0179 | 0.0995 | 0.1961 |
| Strategy IV | 0.1207 | 0.1510 | 0.0000 | 0.0000 | 0.1143 | 0.1776 |
| Strategy V | 0.0059 | 0.0037 | 0.0027 | 0.0029 | 0.0018 | 0.0019 |

Table 11: Proportion of organic duo-culture populations allocated into strains not part of the given duo-culture after predicting strain membership of cells.

| Strategy | Strain Combination | | | |
|---|---|---|---|---|
| | 2375, 2383, 2434 | 2375, 2383, 2524 | 2375, 2434, 2524 | 2383, 2434, 2524 |
| Strategy 0 | 0.0351 | 0.0523 | 0.0565 | 0.0160 |
| Strategy I | 0.0032 | 0.0016 | 0.0008 | 0.0020 |
| Strategy II | 0.0352 | 0.0538 | 0.0552 | 0.0157 |
| Strategy III | 0.0349 | 0.0445 | 0.0630 | 0.0163 |
| Strategy IV | 0.0353 | 0.0538 | 0.0554 | 0.0158 |
| Strategy V | 0.0113 | 0.0021 | 0.0034 | 0.0015 |

Table 12: Proportion of organic tri-culture populations allocated into strains not part of the given tri-culture after predicting strain membership of cells.

The misclassification rates can be visualized in the rightmost plots of figure 19. We can also see in this figure how the population trends of the allocated population to each strain are similar to those of the base models and that all strategies follow the same pattern. This is anticipated since the strategies made use of the predicted probabilities of the base models and that the slight variations in the strategies' population allocation are due to how these new predictors are incorporated in the models.

Similar to what was previously done in the base models, we also looked at the strains that are often misclassified using the proposed strategies. The confusion matrix for each strategy was combined in table 13. Like in the base models, we notice that when the models do not correctly identify a cell of strain 2375 (V) or 2524 (V), they assign them more to strain 2434 (VIII) than to 2383 (VIII). However, unlike in the base models where a cell of strain 2383 or 2434 could be almost indiscriminately misclassified to either 2375 or 2524, in all strategies except the third, if the actual strain is 2383, a cell is more likely to be identified as 2375 rather than 2524, while it is the exact opposite for a cell whose actual strain is 2434. For instance, in strategy 0, there are 58 cells assigned to strain 2375 and 14 cells to strain 2524 when the actual strain is 2383. This is 37 cells against 74 cells when the actual strain is 2434. Note that this tendency was actually also slightly observed in some of the base models. Lastly, while there is almost a concensus as to which strain is more accurately predicted than the rest, which is strain 2375, there is no strain that was singled out to have been more frequently misidentified as other strains. In fact, in most cases, their accuracy is comparable.

| Strategy | True | Predicted Strains | | | | Accuracy |
|---|---|---|---|---|---|---|
| | Strains | 2375 | 2383 | 2434 | 2524 | |
| Strategy 0 | 2375 | 48357 | 14 | 257 | 2381 | 0.9480 |
| | 2383 | 58 | 40932 | 3679 | 14 | 0.9161 |
| | 2434 | 37 | 4879 | 61365 | 74 | 0.9248 |
| | 2524 | 3470 | 15 | 236 | 44422 | 0.9227 |
| Strategy I | 2375 | 339369 | 112 | 878 | 16704 | 0.9504 |
| | 2383 | 374 | 299254 | 12918 | 235 | 0.9568 |
| | 2434 | 334 | 22831 | 440052 | 1268 | 0.9474 |
| | 2524 | 9541 | 121 | 732 | 326607 | 0.9692 |
| Strategy II | 2375 | 48373 | 25 | 209 | 2402 | 0.9483 |
| | 2383 | 55 | 40768 | 3839 | 21 | 0.9124 |
| | 2434 | 67 | 4806 | 61389 | 93 | 0.9252 |
| | 2524 | 3252 | 18 | 199 | 44674 | 0.9279 |
| Strategy III | 2375 | 48645 | 16 | 182 | 2166 | 0.9537 |
| | 2383 | 52 | 41579 | 3027 | 25 | 0.9305 |
| | 2434 | 89 | 5633 | 60545 | 88 | 0.9124 |
| | 2524 | 3665 | 21 | 192 | 44265 | 0.9194 |
| Strategy IV | 2375 | 48539 | 0 | 0 | 2470 | 0.9516 |
| | 2383 | 0 | 40835 | 3848 | 0 | 0.9139 |
| | 2434 | 0 | 4881 | 61474 | 0 | 0.9264 |
| | 2524 | 3404 | 0 | 0 | 44739 | 0.9293 |
| Strategy V | 2375 | 345904 | 176 | 991 | 9992 | 0.9687 |
| | 2383 | 337 | 296810 | 15425 | 209 | 0.9489 |
| | 2434 | 313 | 19699 | 444039 | 434 | 0.9560 |
| | 2524 | 13403 | 173 | 940 | 322485 | 0.9569 |

Table 13: Combined confusion matrices of the proposed strategies

## 6.3 Statistical Analysis Results

We wanted to check whether the cytometer outputs of a cell are partially determined by its strain. If there are significant differences in the cytometer outputs depending on the strain of a particular cell, then using these outputs as predictors to determine the strain membership of a cell is sensible.

The final model that we were able to fit included the interaction effects of the different frequencies with the strain, population, day of measurement and the environmental condition or treatment where the culture was grown or exposed to. Thus, in order to assess whether there is enough evidence to believe that there is a difference between the cytometer output of a certain frequency between two strain, we needed to consider not only the parameter estimates for strain but also those of frequency and those of their interactions. Thus, we have used several contrast statements to estimate the differences and have performed a Wald-test to check if they are significant. The p-values are also corrected for the 48 hypotheses (8 frequencies x 6 pairs) simultaneously tested using Benjamini-Hocher procedure.

The results in 14 suggest that for almost all scenarios, we can reasonably believe that there is a difference between the cytometer outputs of the different strains. The only scenarios where this does not hold is when comparing the NIR.R cytometer output of strains 2375 and 2434 and the RED.B cytometer output of strains 2383 and 2434. Furthermore, we notice that the absolute values of the parameter estimates for YEL.B when comparing strains of different species (2375 vs 2383, 2375 vs 2434, 2383 vs 2524, 2434 vs 2424) were larger than the other frequency differences. This explains why the most important feature or predictor when creating the base models was YEL.B for those cases. Similarly, RED.R, which was the most important predictor for the base model for 2383 and 2434, has the highest average difference when comparing the cytometer outputs for these strains. Although the estimate for NIR.R is also high, and almost of the same magnitude with RED.R, this was no longer considered an important feature for the model because of its high correlation with RED.R. This correlation could also explain the close estimated difference. However, the estimated average difference for FSC when comparing 2375 and 2524, which was supposedly the most important predictor, was not the largest.

However, although the results seem to be ideal, we must be careful with only taking into account the calculated p-values. Since we have hundreds of thousands of observations, it is not surprising to see very significant results. However, comparing the resulting differences, for instance, to the parameter estimate of the intercept of the model, which is 3.8312, we can vaguely say that these differences matter.

We also fit a similar model on concatenated data from organic multi-cultures and synthetic

|  | 2375 vs 2383 |  |  | 2375 vs 2434 |  |  | 2375 vs 2524 |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* |
| FSC | 0.9089 | 0.0196 | < 0.0001 | 0.6995 | 0.0177 | < 0.0001 | -0.8609 | 0.0178 | < 0.0001 |
| SSC | 1.5306 | 0.0337 | < 0.0001 | 1.2578 | 0.0321 | < 0.0001 | 0.9664 | 0.0335 | < 0.0001 |
| GRN.B | -0.9247 | 0.0264 | < 0.0001 | -0.8268 | 0.0256 | < 0.0001 | -1.3303 | 0.0262 | < 0.0001 |
| YEL.B | 5.5222 | 0.0381 | < 0.0001 | 5.6078 | 0.0361 | < 0.0001 | 1.4463 | 0.0377 | < 0.0001 |
| RED.B | 3.8975 | 0.0266 | < 0.0001 | 3.8744 | 0.0252 | < 0.0001 | 1.8175 | 0.0259 | < 0.0001 |
| NIR.B | 2.5346 | 0.0278 | < 0.0001 | 2.6144 | 0.0270 | < 0.0001 | 0.6839 | 0.0275 | < 0.0001 |
| RED.R | 0.3485 | 0.0320 | < 0.0001 | 1.3206 | 0.0305 | < 0.0001 | 1.0565 | 0.0313 | < 0.0001 |
| NIR.R | -0.7635 | 0.0306 | < 0.0001 | 0.0671 | 0.0291 | 0.0634 | -0.3433 | 0.0300 | < 0.0001 |
|  | 2383 vs 2434 |  |  | 2383 vs 2524 |  |  | 2434 vs 2524 |  |  |
|  | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* |
| FSC | -0.2094 | 0.0173 | < 0.0001 | -1.7698 | 0.0203 | < 0.0001 | -1.5604 | 0.0184 | < 0.0001 |
| SSC | -0.2728 | 0.0186 | < 0.0001 | -0.5642 | 0.0223 | < 0.0001 | -0.2914 | 0.0200 | < 0.0001 |
| GRN.B | 0.0979 | 0.0122 | < 0.0001 | -0.4056 | 0.0144 | < 0.0001 | -0.5035 | 0.0132 | < 0.0001 |
| YEL.B | 0.0856 | 0.0206 | < 0.0001 | -4.0759 | 0.0240 | < 0.0001 | -4.1615 | 0.0213 | < 0.0001 |
| RED.B | -0.0231 | 0.0160 | 0.4465 | -2.0800 | 0.0187 | < 0.0001 | -2.0569 | 0.0164 | < 0.0001 |
| NIR.B | 0.0798 | 0.0150 | < 0.0001 | -1.8507 | 0.0177 | < 0.0001 | -1.9305 | 0.0163 | < 0.0001 |
| RED.R | 0.9721 | 0.0184 | < 0.0001 | 0.7080 | 0.0212 | < 0.0001 | -0.2641 | 0.0192 | < 0.0001 |
| NIR.R | 0.8306 | 0.0177 | < 0.0001 | 0.4202 | 0.0203 | < 0.0001 | -0.4104 | 0.0184 | < 0.0001 |

Table 14: Pairwise difference of the average cytometer outputs among the four strains. The p-values were adjusted using Benjamini-Hochberg to correct for multiple-hypotheses (48) testing.

multi-cultures to investigate whether these cultures significantly differ in their cytometer outputs. Recall that although we know that there are factors present in organic multi-cultures that are not present in the synthetic multi-cultures that could affect the behavior of the cells, and in turn, affect the cytometer readings, we are banking on the assumption that such difference is minimal and can be ignored.

Similarly to the first statistical analysis, we were able to include the interaction effects of the frequencies with date, population, condition, and the culture type (organic vs synthetic). As such, we have also calculated the differences in the cytometer outputs per frequency between organic and synthetic multi-cultures in all strain combinations. The results are summarized in table 15 and show that there is enough evidence to reject the null hypothesis, that is, the cytometer outputs of cells from a synthetic multi-culture are significantly different from those of cells from an organic multi-culture. Among the 88 comparisons, there were only 6 cases in which the analysis did not show evidence of such a difference. These are for FSC in the multi-cultures (2383, 2434), (2375, 2383, 2434), (2375, 2434, 2524), and (2375, 2383, 2434, 2524), YEL.B in (2383, 2524) and NIR.B in (2434, 2524). This is quite alarming because looking at the corrected p-values and even at the calculated differences themselves, the synthetic multi-culture is quite different from the organic multi-culture. This difference could be attributed to the interactions among the strains present in the organic multi-cultures. Some strains interact more than others, and some interact less. However, given the circumstances, we cannot do any better. Thus, although we have trained the classifiers using data that is arguably different from the data for which it is intended, we can still use the classifier models but with great caution.

| | 2375, 2383 | | | 2375, 2434 | | | 2375, 2524 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* |
| FSC | 0.1318 | 0.0182 | < 0.0001 | -0.1221 | 0.0166 | < 0.0001 | 0.1253 | 0.0185 | < 0.0001 |
| SSC | -1.7786 | 0.0359 | < 0.0001 | -1.4367 | 0.0411 | < 0.0001 | -1.2866 | 0.0388 | < 0.0001 |
| GRN.B | 0.5890 | 0.0314 | < 0.0001 | 0.7348 | 0.0329 | < 0.0001 | 1.2732 | 0.0307 | < 0.0001 |
| YEL.B | -0.8470 | 0.0880 | < 0.0001 | -2.1266 | 0.0959 | < 0.0001 | -1.4429 | 0.0503 | < 0.0001 |
| RED.B | -1.7725 | 0.0431 | < 0.0001 | -2.5789 | 0.0448 | < 0.0001 | -1.7205 | 0.0325 | < 0.0001 |
| NIR.B | -0.7871 | 0.0395 | < 0.0001 | -1.3902 | 0.0444 | < 0.0001 | -0.6216 | 0.0318 | < 0.0001 |
| RED.R | -2.5367 | 0.0526 | < 0.0001 | -2.6158 | 0.0455 | < 0.0001 | -1.3211 | 0.0294 | < 0.0001 |
| NIR.R | -0.9783 | 0.0462 | < 0.0001 | -1.0708 | 0.0410 | < 0.0001 | 0.0983 | 0.0296 | 0.0049 |
| | 2383, 2434 | | | 2383, 2524 | | | 2434, 2524 | | |
| | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* |
| FSC | 0.0034 | 0.0139 | 1.0000 | 0.3083 | 0.0246 | < 0.0001 | 0.3095 | 0.0209 | < 0.0001 |
| SSC | -2.0160 | 0.0375 | < 0.0001 | -1.1336 | 0.0409 | < 0.0001 | -1.3170 | 0.0397 | < 0.0001 |
| GRN.B | 0.3455 | 0.0328 | < 0.0001 | 1.1664 | 0.0342 | < 0.0001 | 0.9927 | 0.0319 | < 0.0001 |
| YEL.B | 0.9903 | 0.0348 | < 0.0001 | -0.0080 | 0.0831 | 1.0000 | 0.4057 | 0.0759 | < 0.0001 |
| RED.B | -1.2462 | 0.0266 | < 0.0001 | -1.1824 | 0.0391 | < 0.0001 | -1.0065 | 0.0351 | < 0.0001 |
| NIR.B | -0.3254 | 0.0322 | < 0.0001 | -0.1547 | 0.0371 | 0.0002 | -0.0482 | 0.0355 | 0.9601 |
| RED.R | -4.0544 | 0.0456 | < 0.0001 | -2.5020 | 0.0530 | < 0.0001 | -2.6625 | 0.0428 | < 0.0001 |
| NIR.R | -2.2491 | 0.0423 | < 0.0001 | -0.8917 | 0.0477 | < 0.0001 | -0.9816 | 0.0392 | < 0.0001 |
| | 2375, 2383,2434 | | | 2375, 2383, 2524 | | | 2375, 2434, 2524 | | |
| | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* |
| FSC | -0.0415 | 0.0162 | 0.0573 | 0.0801 | 0.0223 | 0.0018 | 0.0411 | 0.0189 | 0.16313 |
| SSC | -1.5416 | 0.0362 | < 0.0001 | -1.2624 | 0.0379 | < 0.0001 | -1.1963 | 0.0376 | < 0.0001 |
| GRN.B | 0.7074 | 0.0303 | < 0.0001 | 1.1133 | 0.0308 | < 0.0001 | 1.2004 | 0.0293 | < 0.0001 |
| YEL.B | -1.0667 | 0.0818 | < 0.0001 | -1.3175 | 0.0846 | < 0.0001 | -1.7354 | 0.0808 | < 0.0001 |
| RED.B | -2.1042 | 0.0395 | < 0.0001 | -2.0221 | 0.0413 | < 0.0001 | -1.9813 | 0.0383 | < 0.0001 |
| NIR.B | -1.0202 | 0.0389 | < 0.0001 | -0.8549 | 0.0379 | < 0.0001 | -0.8410 | 0.0372 | < 0.0001 |
| RED.R | -2.8973 | 0.0491 | < 0.0001 | -2.0642 | 0.0503 | < 0.0001 | -1.9659 | 0.0383 | < 0.0001 |
| NIR.R | -1.3019 | 0.0435 | < 0.0001 | -0.5869 | 0.0446 | < 0.0001 | -0.4517 | 0.0347 | < 0.0001 |
| | 2383, 2434, 2524 | | | 2375, 2383, 2434, 2524 | | | | | |
| | Est. | Std. Err. | p-value* | Est. | Std. Err. | p-value* | | | |
| FSC | 0.2162 | 0.0200 | < 0.0001 | -0.0121 | 0.0200 | 1.0000 | | | |
| SSC | -1.3182 | 0.0372 | < 0.0001 | -1.3389 | 0.0364 | < 0.0001 | | | |
| GRN.B | 1.0137 | 0.0308 | < 0.0001 | 1.0355 | 0.0291 | < 0.0001 | | | |
| YEL.B | 0.2852 | 0.0702 | 0.0003 | -1.3301 | 0.0821 | < 0.0001 | | | |
| RED.B | -1.2428 | 0.0332 | < 0.0001 | -2.0595 | 0.0389 | < 0.0001 | | | |
| NIR.B | -0.2117 | 0.0338 | < 0.0001 | -0.9090 | 0.0373 | < 0.0001 | | | |
| RED.R | -3.2160 | 0.0467 | < 0.0001 | -2.5893 | 0.0461 | < 0.0001 | | | |
| NIR.R | -1.4853 | 0.0422 | < 0.0001 | -1.0133 | 0.0412 | < 0.0001 | | | |

Table 15: Difference of the average cytometer outputs between synthetic and organic multi-cultures in each strain combination. The p-values were adjusted using Benjamini-Hochberg to correct for multiple-hypotheses (88) testing

In section 6.2, tables 11 and 12, we redefined the notion of misclassification in the context of using the models in organic multi-cultures to assess the validity of the models. For similar reason, we have compared the cytometer outputs of the mono-strain cultures and those of the cells from the organic multi-culture predicted to belong to the same strain using strategy V since along with the first strategy, it resulted in a better model. The results of the model for each strain are given in table 16, where the parameters are indicator variables, indicating whether or not the cell is from the multi-culture sample of a particular strain combination. For each strain, there are a total of seven organic multi-strain cultures where that strain is present; thus, we have a total of seven groups of cells predicted to belong to that strain. Furthermore, the fitted model did not contain the interaction between the frequencies and the indicator variable for the culture type.

In almost all scenarios, the analysis showed that there is enough evidence to reject the hypotheses that the cytometer outputs of the predicted cells are indistinguishable from those of the mono-strain cultures. Only for cells predicted to belong to strain 2383 in the duo-culture (2383, 2434), 2434 in the multi-cultures (2375, 2434), (2434, 2524), and (2375, 2383, 2434), and 2524 in the duo-culture (2375, 2524) was there insufficient evidence to state that the cytometer outputs are different. These results suggest that if we firmly believe that cells of, for instance, strain 2375 behave the same way whether cultivated in a mono-strain environment or not, then clearly the model has failed. However, similar to the initial warning, focusing mainly on p-values when there is a sufficiently large number of observations in the dataset could be problematic. In fact, it could be argued that the parameter estimates of the indicator variables in all strains are relatively small to constitute a difference. In addition, the model is not accurate 100%, and so we expect that some cells labeled as a particular strain are not really of that strain. This could also cause the difference between the cytometer outputs.

For the last analysis, we tried to understand why strategies II and III did not perform better than the base strategy. Recall that both of these strategies ensemble the predicted probabilities of the base models, including those of the base strategy, albeit not in a clever manner. It seems that the predicted probabilities obtained from the base models for the duo-cultures are not helpful in getting better results and are just confusing the predicted probabilities of the tetra-culture model. This could explain why the accuracies obtained are not higher than those of strategy 0. Note that for strategies II and III, although we know, for instance, that a culture only contains strains of 2375 (V) and 2383 (VIII), because of the way the model is built, we will include predicted probabilities from the base model for 2434 (VIII) and 2524 (V). Pragmatically, what these predicted probabilities are trying to do is assign a cell that is actually 2375 to 2524, and similarly a cell that is 2383 to 2434. This is what might have caused the models for these strategies to perform poorly.

In table 17 are the results of the paired t-test performed. The first observation we make is

| Strain | Parameter | | Estimate | Std. Error | p-value* |
|---|---|---|---|---|---|
| 2375 | 2375_2383 | $\beta_8$ | 0.0627 | 0.0044 | $< 0.0001$ |
| | 2375_2434 | $\beta_8$ | 0.0556 | 0.0045 | $< 0.0001$ |
| | 2375_2524 | $\beta_8$ | -0.0100 | 0.0033 | 0.0025 |
| | 2375_2383_2434 | $\beta_8$ | 0.0708 | 0.0052 | $< 0.0001$ |
| | 2375_2383_2524 | $\beta_8$ | 0.0571 | 0.0050 | $< 0.0001$ |
| | 2375_2434_2524 | $\beta_8$ | 0.0744 | 0.0044 | $< 0.0001$ |
| | 2375_2383_2434_2524 | $\beta_8$ | 0.0824 | 0.0054 | $< 0.0001$ |
| 2383 | 2375_2383 | $\beta_8$ | -0.0216 | 0.0030 | $< 0.0001$ |
| | 2383_2434 | $\beta_8$ | -0.0039 | 0.0028 | 0.1573 |
| | 2383_2524 | $\beta_8$ | 0.0070 | 0.0026 | 0.0091 |
| | 2375_2383_2434 | $\beta_8$ | 0.0092 | 0.0033 | 0.0074 |
| | 2375_2383_2524 | $\beta_8$ | 0.0184 | 0.0030 | $< 0.0001$ |
| | 2383_2434_2524 | $\beta_8$ | 0.0204 | 0.0029 | $< 0.0001$ |
| | 2375_2383_2434_2524 | $\beta_8$ | 0.0232 | 0.0034 | $< 0.0001$ |
| 2434 | 2375_2434 | $\beta_8$ | 0.0021 | 0.0033 | 0.5286 |
| | 2383_2434 | $\beta_8$ | -0.0227 | 0.0034 | $< 0.0001$ |
| | 2434_2524 | $\beta_8$ | -0.0045 | 0.0027 | 0.1201 |
| | 2375_2383_2434 | $\beta_8$ | 0.0081 | 0.0041 | 0.0643 |
| | 2375_2434_2524 | $\beta_8$ | -0.0186 | 0.0033 | $< 0.0001$ |
| | 2383_2434_2524 | $\beta_8$ | 0.0114 | 0.0037 | 0.0037 |
| | 2375_2383_2434_2524 | $\beta_8$ | 0.0468 | 0.0044 | $< 0.0001$ |
| 2524 | 2375_2524 | $\beta_8$ | 0.0065 | 0.0060 | 0.2821 |
| | 2383_2524 | $\beta_8$ | 0.0273 | 0.0066 | $< 0.0001$ |
| | 2434_2524 | $\beta_8$ | -0.0282 | 0.0056 | $< 0.0001$ |
| | 2375_2383_2524 | $\beta_8$ | 0.0467 | 0.0082 | $< 0.0001$ |
| | 2375_2434_2524 | $\beta_8$ | 0.0449 | 0.0076 | $< 0.0001$ |
| | 2383_2434_2524 | $\beta_8$ | 0.0464 | 0.0078 | $< 0.0001$ |
| | 2375_2383_2434_2524 | $\beta_8$ | 0.0724 | 0.0087 | $< 0.0001$ |

Table 16: Results of comparing the cytometer outputs of the mono-strain cultures to those of cells predicted to belong to a given strain that were cultivated in a multi-strain set-up. The p-values were adjusted using Benjamini-Hochberg to correct for multiple-hypotheses (7) testing

how similar the average predicted probabilities are whether we have used the appropriate models or not, and they are almost equal to 1.00. This is generally true except for 2434, where the inappropriate models yielded a much inferior value. This only means that instead of classifying some cells of strain 2434 to its co-species 2383, they are being classified to the strains of other species. The second observation is that for all strains, the paired t-tests resulted in having rejected the null hypothesis that the average predicted probabilities are the same. Although this is not what was expected, perhaps the models in strategies II and III do not discriminate these predicted probabilities, thus resulting in inferior accuracies.

| Strain | Average of predicted probabilities | | | p-value |
|---|---|---|---|---|
| | Appropriate models | Inappropriate models | Mean Difference | |
| 2375 | 0.9965 | 0.9802 | 0.0163 | $< 0.0001$ |
| 2383 | 0.9977 | 0.9964 | 0.0013 | $< 0.0001$ |
| 2434 | 0.9966 | 0.6469 | 0.3497 | $< 0.0001$ |
| 2524 | 0.9965 | 0.9919 | 0.0045 | $< 0.0001$ |

Table 17: Paired t-test results of comparing the predicted probabilities of appropriate models and inappropriate models on duo-culture data.

## 7 Conclusion and Future Studies

In order to discriminate the cells in samples of multi-strain cultures, we have created various base models and implemented various strategies to combine the results of these base models. The base models are quite accurate, and the accuracies increase by up to almost 100% when classifying cells belonging to different species, in which case the most important predictor is YEL.B. However, these accuracies drop to around 92-94%, which are still high, when classifying strains from the same species. In these cases, the indicator variables for the culture environment conditions and the days of measurement seem to have an impact on the model, suggesting that it is harder to discriminate cells belonging to the same species. In addition, when classifying cells of species V (2375, 2524), FSC is the most important predictor, while this is RED.R for species VIII (2383, 2434). Lastly, when building base models for tri-strain and tetra-strain cultures using random forests, for pairs of almost perfectly correlated cytometer outputs like RED.R and NIR.R, and RED.B and NIR.B, only one cytometer output of each pair is important. This is not the case for the xgboost base models, where the most important features are YEL.B, FSC, and RED.R.

We ensemble the xgboost base models to combine the patterns learned in each multi-culture model. Ensembling naively leads to inferior models, as in the case with strategies II and III, while ensembling with hints of which strain a cell could belong to is better, as in the case with strategies I and V. Nevertheless, these remain inferior to just using the base models individually, and even assign cells to strains not present in the multi-culture. This kind of misclassification is non-existent in the base models. However, this misclassification could be seen as a positive feature of ensembling. It could be argued that removing the misclassified cells could be seen as removing uncertainties since the ensembled model does not confidently classify them to any strains present in the multi-culture from which the cell is sampled.

All these models have been built under the principles of supervised machine learning and were trained on synthetic multi-cultures, working on the assumption that the cells' characteristics remain unchanged, or at least are insignificant, whether they are cultivated separately, that is, one strain per culture, or are cultivated in the presence of other strains.

The statistical analysis performed showed that in most cases, we have found significant evidence to reject our assumption. Thus, the models to predict strain memberships of cells in organic multi-cultures must be used cautiously.

Given this, we do not have any mechanisms to assess whether the models correctly discriminate cells and assign them to their correct strains. In order to gauge the correctness of the models, we have compared the cytometer outputs of cells from mono-strain cultures and cells from organic multi-strain cultures predicted to belong to a given strain. Our statistical analysis was performed on the assumption that they are similar, and thus we expect no significance difference. However, as our results showed, this is not mainly the case. However, the parameter estimates for the observed differences are relatively small.

The difficulty encountered in classification and the uncertainties with the statistical analysis could be both attributed to the inter-strain interaction in organic multi-culture setups that were not captured in the synthetic multi-cultures with which the models were trained.

For future studies, instead of using all predictors, we might just want to use the most important features identified above. Accordingly, instead of modeling all cytometer outputs to determine whether synthetic multi-cultures are significantly different from organic multi-cultures, it should be enough to only model the cytometer outputs which were retained as predictors. Furthermore, it might be wise to train the models only on synthetic multi-cultures that are shown to have no significant difference to the organic multi-cultures. In addition, we have seen how base models for multi-cultures containing strains of the same species have lower accuracies. Adjusting the model parameters could help to improve model performance. Thus, hyper-tuning the model parameters is recommended. Alternative strategies in combining the base models could also be explored, for instance, combining 4 one-vs-rest models and 6 pairwise models.

Lastly, to incorporate inter-strain interactions, we could set up a smaller experiment of multi-strain cultures, where we employ various techniques to know the actual strains of cells of a small sample, for example, using molecular techniques only on around a hundred or so cells. This would allow us to use supervised machine learning directly on organic multi-strain cultures.

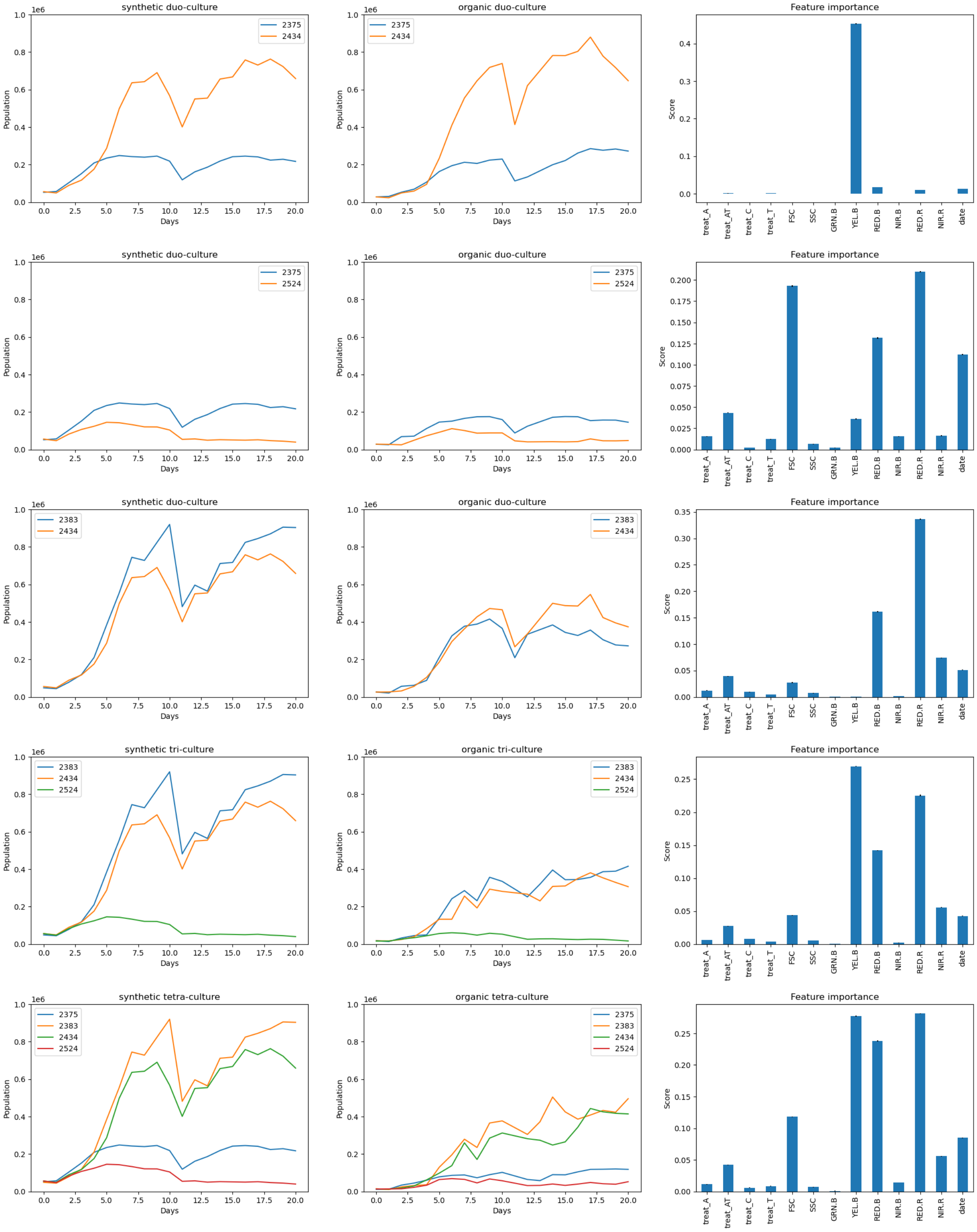

Figure 16: Population evolution strains in synthetic multi-cultures (left) and in organic multi-cultures (middle) after using the random forest classifiers, and the feature scores of the predictors used in each classifier (right).

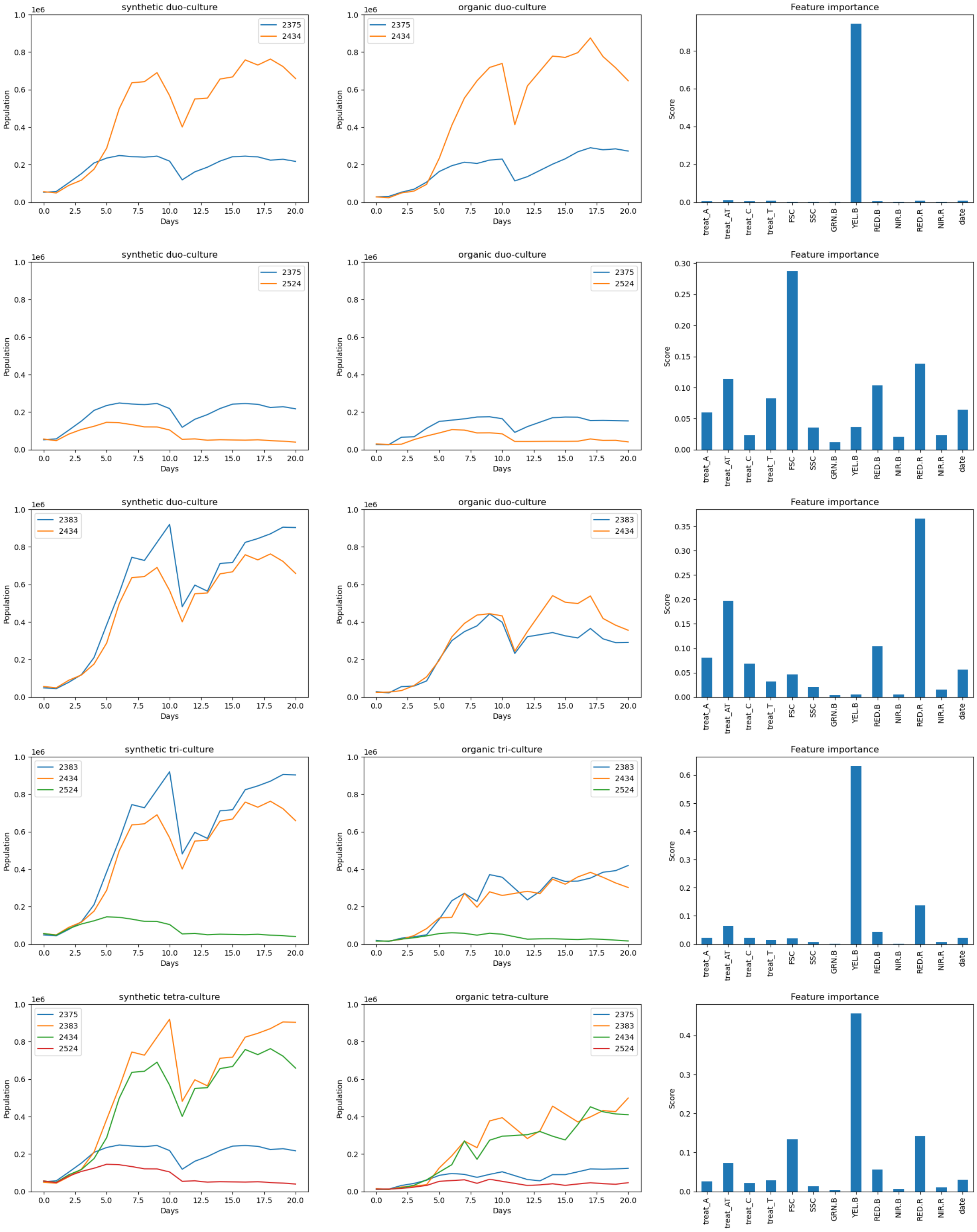

Figure 17: Population evolution strains in synthetic multi-cultures (left) and in organic multi-cultures (middle) after using the xgboost classifiers, and the feature scores of the predictors used in each classifier (right).

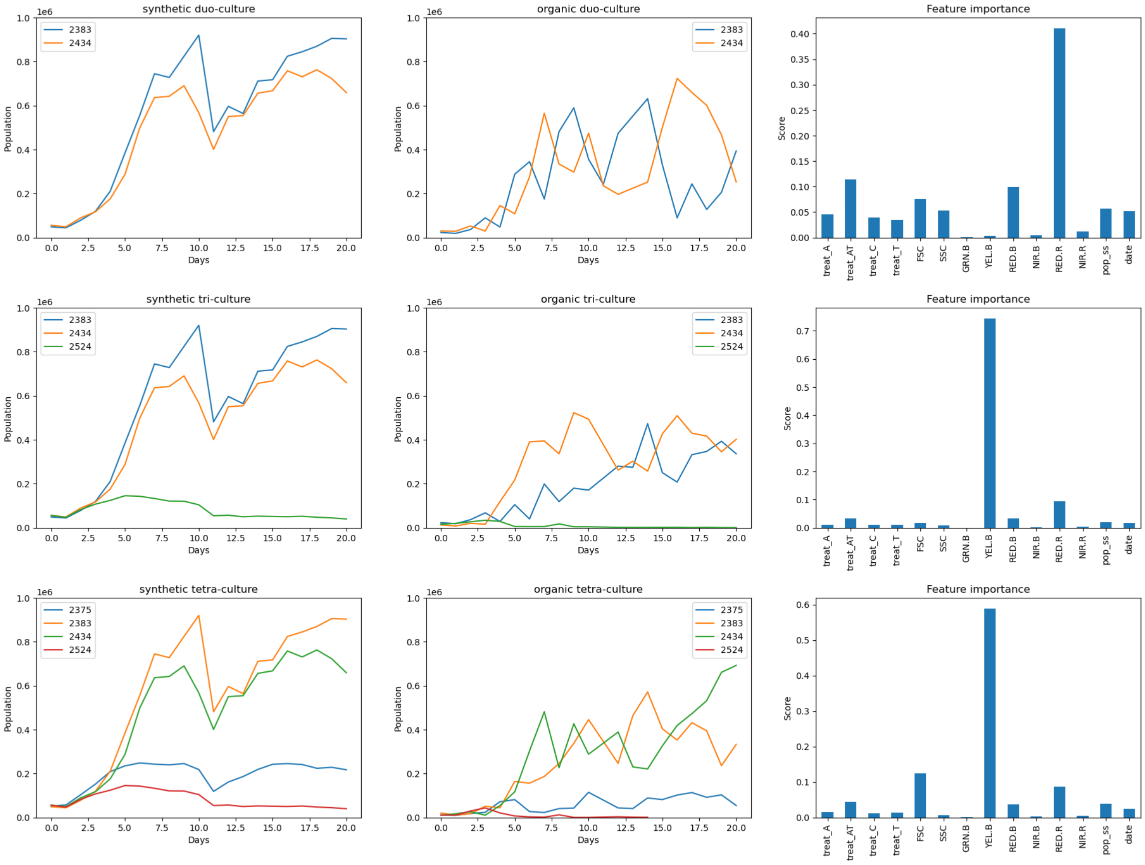

Figure 18: Population evolution strains in synthetic multi-cultures (left) and in organic multi-cultures (middle) after using xgboost classifiers, and the feature scores of the predictors, which includes the population, used in each classifier (right).

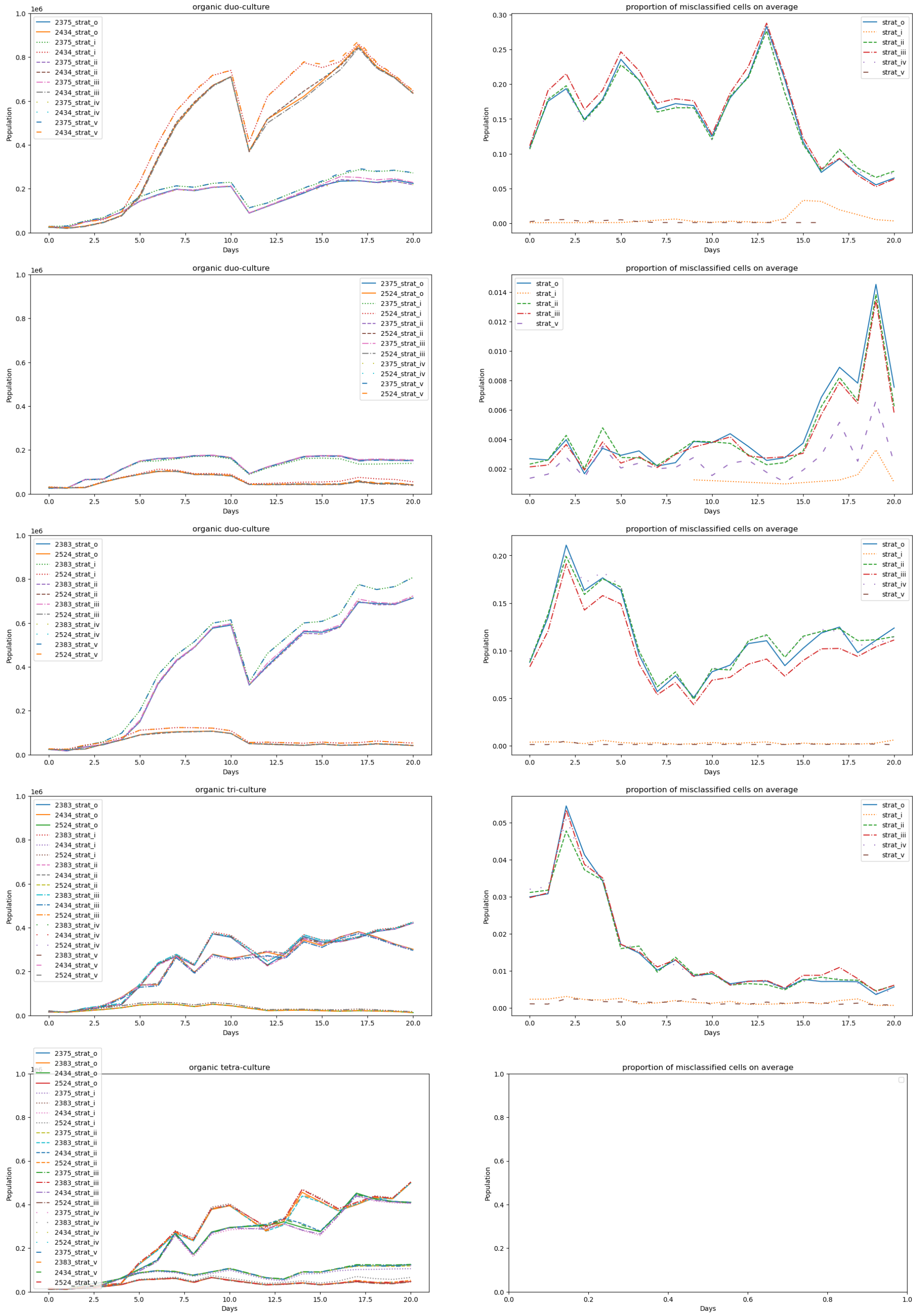

Figure 19: Population evolution of strains in organic multi-cultures (left) after using the various proposed strategies, and the proportion of population assigned to strains not part of the organic multi-culture (right).

# References

[1] Mackay, E.B., Jones, I.D., & Gray, E. (2022). Biophysical Interactions in Phytoplankton. In: Tockner, K. & Mehner, T. (Eds.). Encyclopedia of Inland Waters (2nd ed.). Amsterdam: Elsevier. 154-162

[2] Pal, R. & Choudhury, A. K. (2014). An Introduction to Phytoplanktons: Diversity and Ecology. Springer.

[3] Rousseaux, C. S., & Gregg, W. W. (2014). Interannual Variation in Phytoplankton Primary Production at A Global Scale. Remote Sensing, 6(1), 1-19. https://doi.org/10.3390/rs6010001

[4] Clementson, L. (2021). Monitoring and sensing systems. In M. C. Smith. (pp. 155-158). https://doi.org/10.1016/B978-0-12-822861-6.00014-5

[5] Smith, M. C., & Bodrossy, L. (2020). Advances in in situ molecular systems for phytoplankton research and monitoring. In M. C. Smith. (pp. 191-215). https://doi.org/10.1016/B978-0-12-822861-6.00014-5

[6] Kruk, C., Huszar, V. L. M., Peeters, E. T. H. M., Bonilla, S., Costa, L., Lürling, M., Reynolds, C. S., & Scheffer, M. (2010). A morphological classification capturing functional variation in phytoplankton. Freshwater Biology, 55(3), 614-627.

[7] Why is biodiversity important?: Royal Society (no date) The Royal Society. Available at: https://royalsociety.org/news-resources/projects/biodiversity/why-is-biodiversity-important/ (Accessed: 31 May 2025).

[8] Aquatic Food Webs — National Oceanic and Atmospheric Administration (no date). Available at: https://www.noaa.gov/education/resource-collections/marine-life/aquatic-food-webs (Accessed: 31 May 2025).

[9] Phytoplankton (no date) MIT Climate Portal. Available at: https://climate.mit.edu/explainers/phytoplankton (Accessed: 16 June 2025).

[10] Agawin NSR, Duarte CM, Agustí S (1998) Growth and abundance of Synechococcus sp. in a Mediterranean Bay: seasonality and relationship with temperature. Mar Ecol Prog Ser 170: 4553

[11] James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). An Introduction to Statistical Learning with Applications in Python. Springer. pp. 343-354

[12] Nembrini, S., König, I. R., & Wright, M. N. (2018). The revival of the Gini importance? *Bioinformatics, 34*(21), 3711–3718. https://doi.org/10.1093/bioinformatics/bty373

[13] XGBOOST Best Feature Importance Score (no date) XGBoosting. Available at: https://xgboosting.com/xgboost-best-feature-importance-score/ (Accessed: 31 May 2025).

# A Appendices

## A.1 Complete Base Model Population Plots

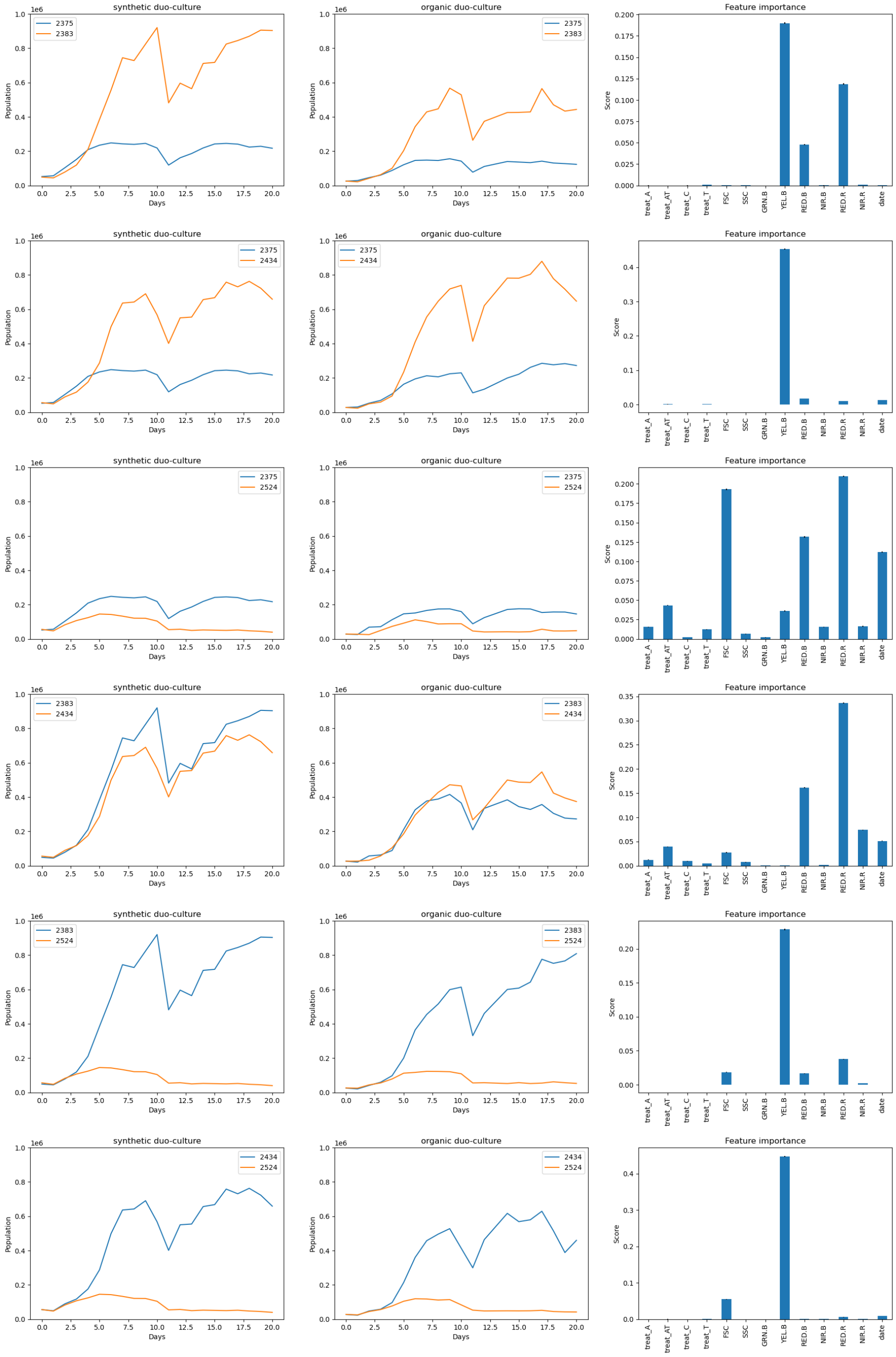

Figure 20: Population evolution of synthetic duo-cultures (left) and organic duo-cultures (middle) resulting from allocating the actual duo-culture population using strain membership predictions using random forest classifiers, and the feature scores of the predictors used (right).

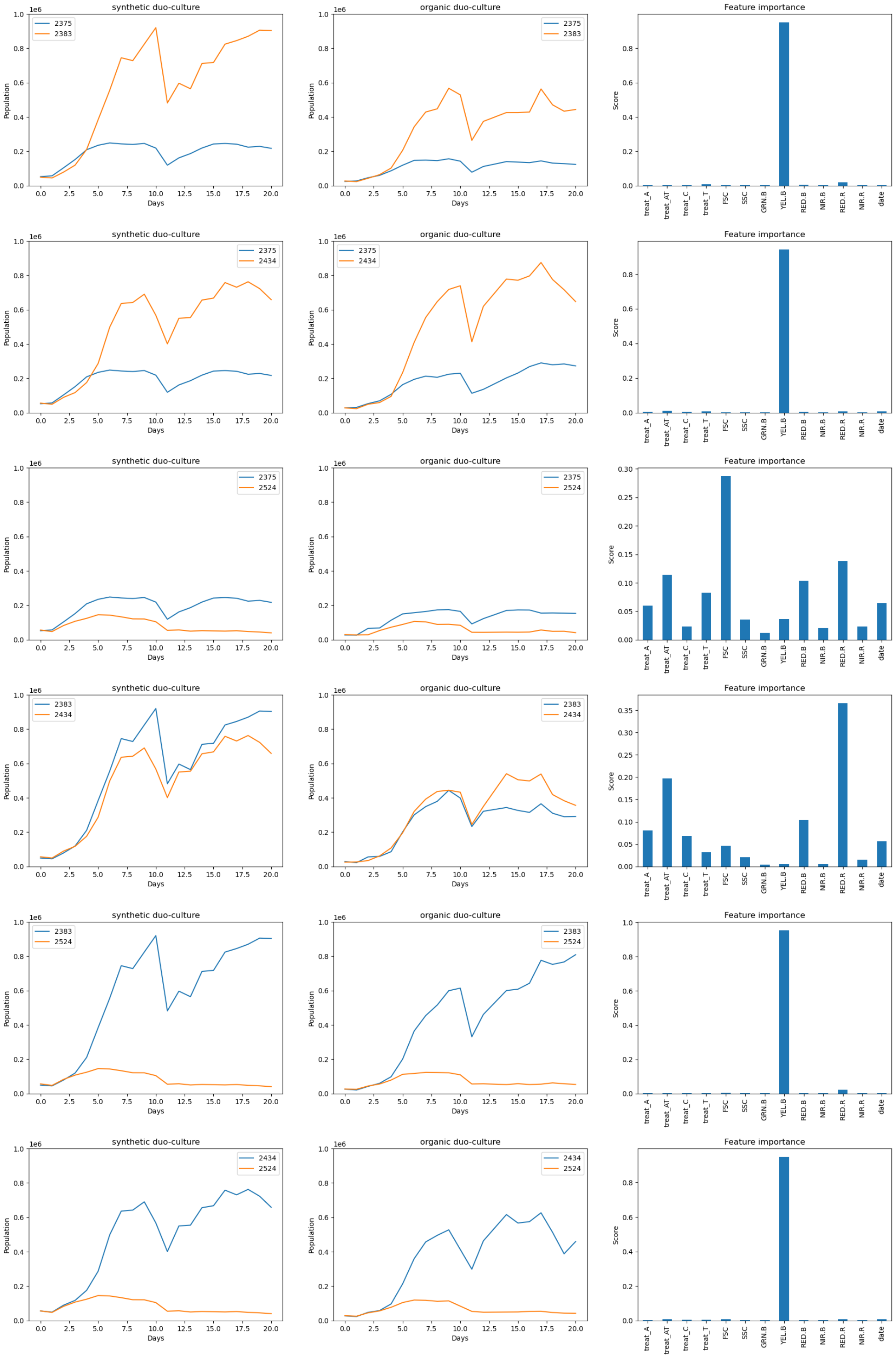

Figure 21: Population evolution of synthetic duo-cultures (left) and organic duo-cultures (middle) resulting from allocating the actual duo-culture population using strain membership predictions of cells using xgboost classifiers, and the feature scores of the predictors used in each classifier (right).

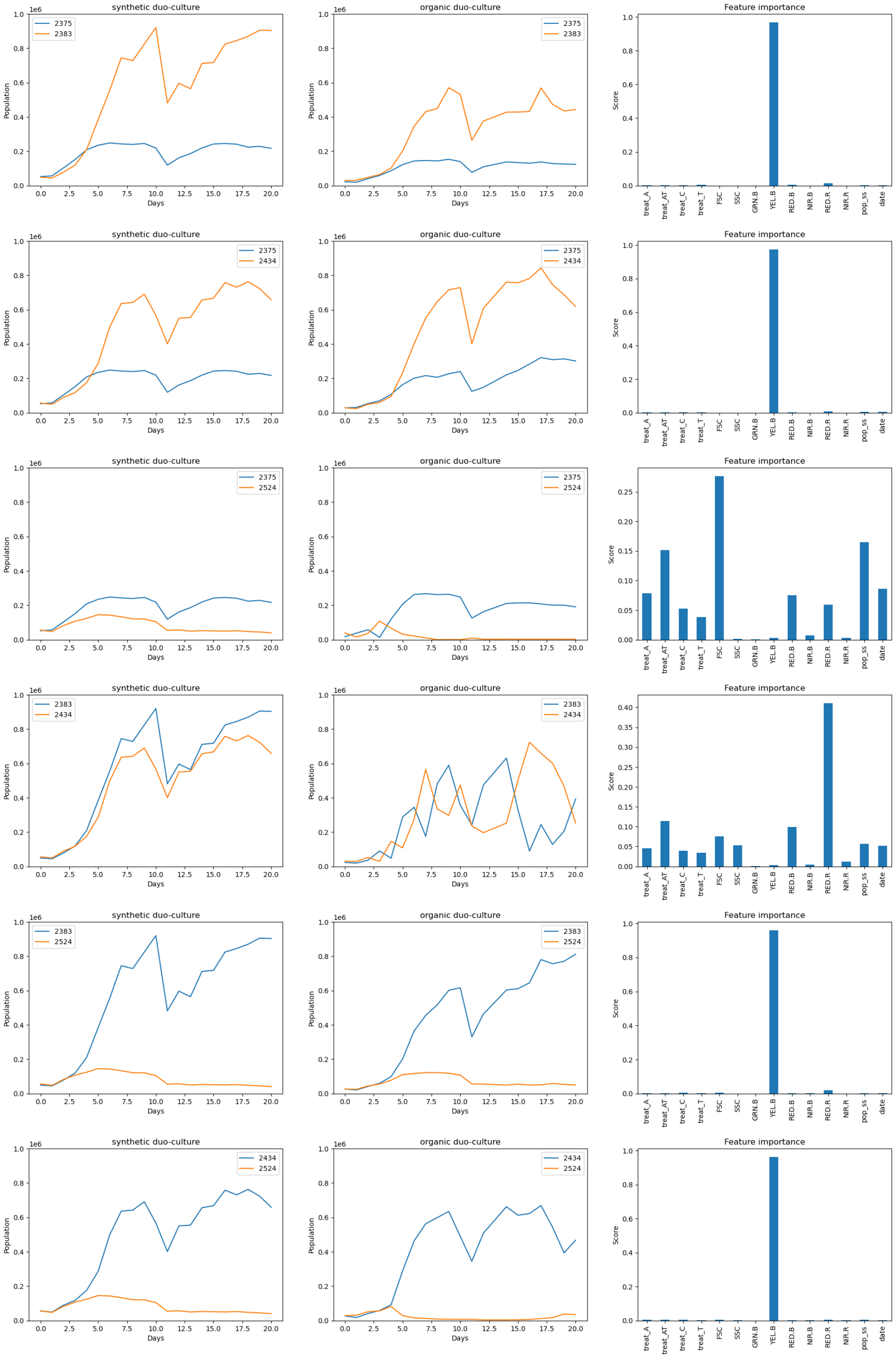

Figure 22: Population evolution of synthetic duo-cultures (left) and organic duo-cultures (middle) resulting from allocating the actual duo-culture population using strain membership predictions of cells using xgboost classifiers, and the feature scores of the predictors, including population, used in each classifier (right).

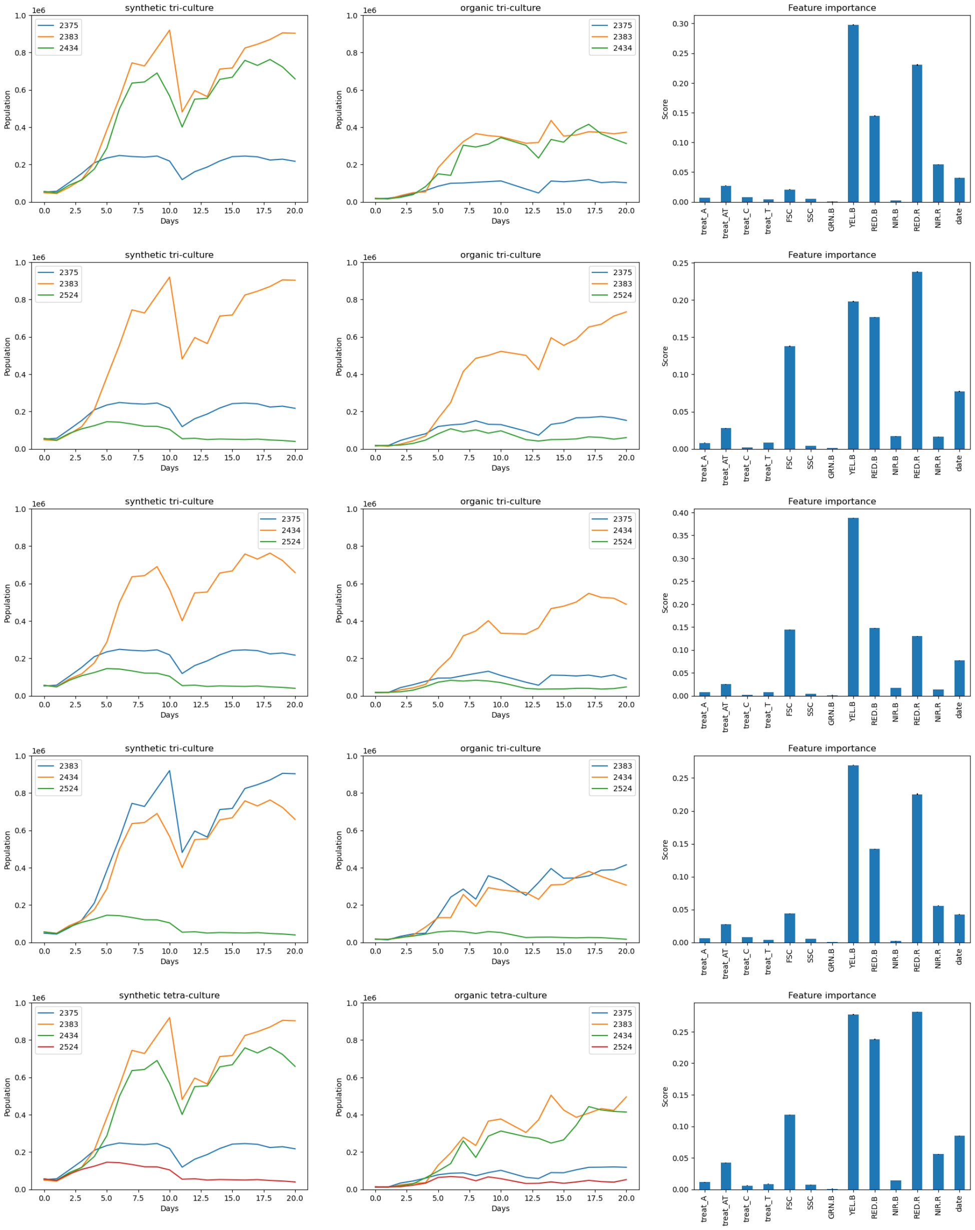

Figure 23: Population evolution of synthetic tri-cultures and tetra-culture (left) and organic tri-cultures and tetra-culture (middle) resulting from allocating the actual tri-culture and tetra-culture population using strain membership predictions of cells using random forest classifiers, and the feature scores of the predictors used in each classifier (right).

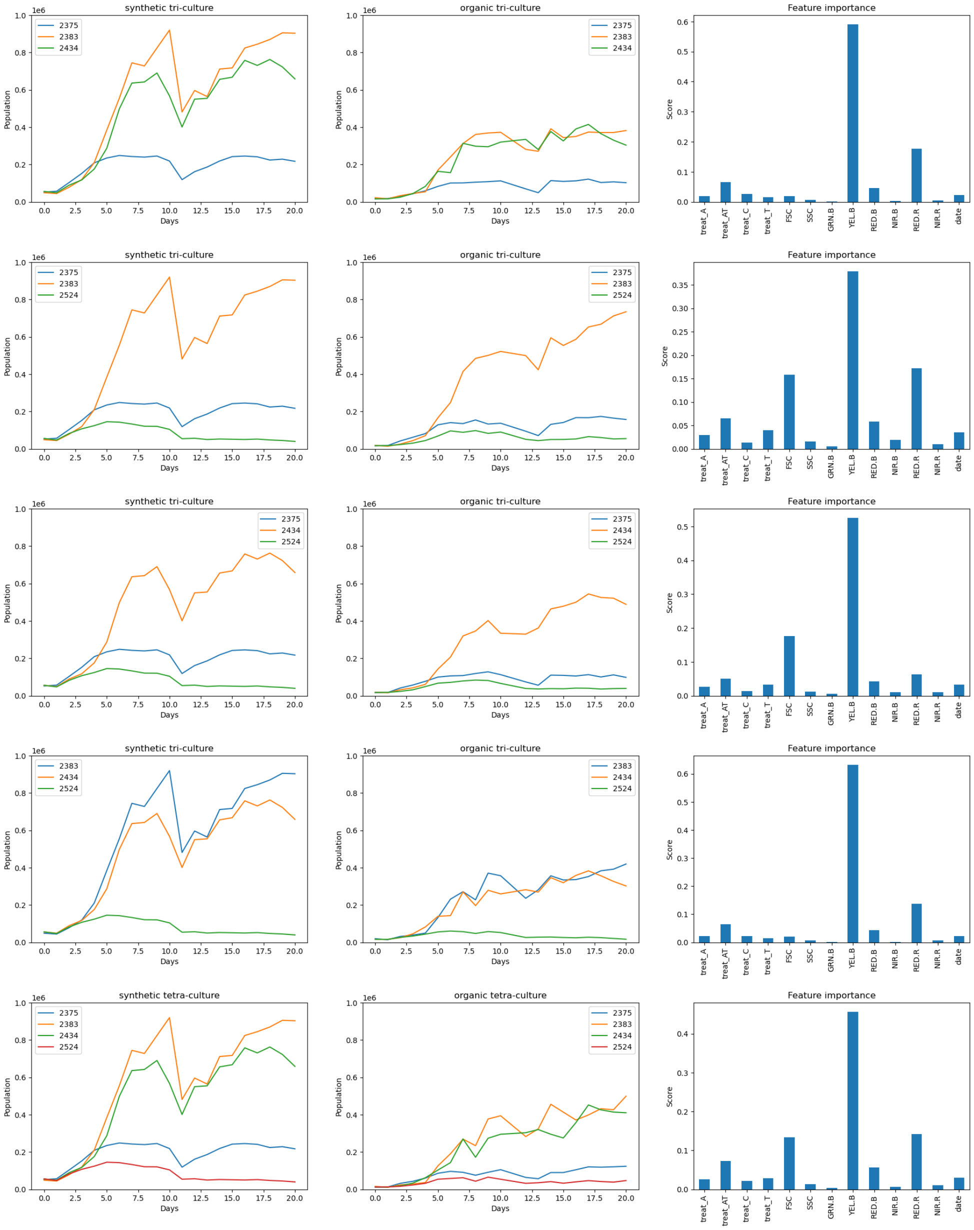

Figure 24: Population evolution of synthetic tri-cultures and tetra-culture (left) and organic tri-cultures and tetra-culture (middle) resulting from allocating the actual tri-culture and tetra-culture population using strain membership predictions of cells using xgboost classifiers, and the feature scores of the predictors used in each classifier (right).

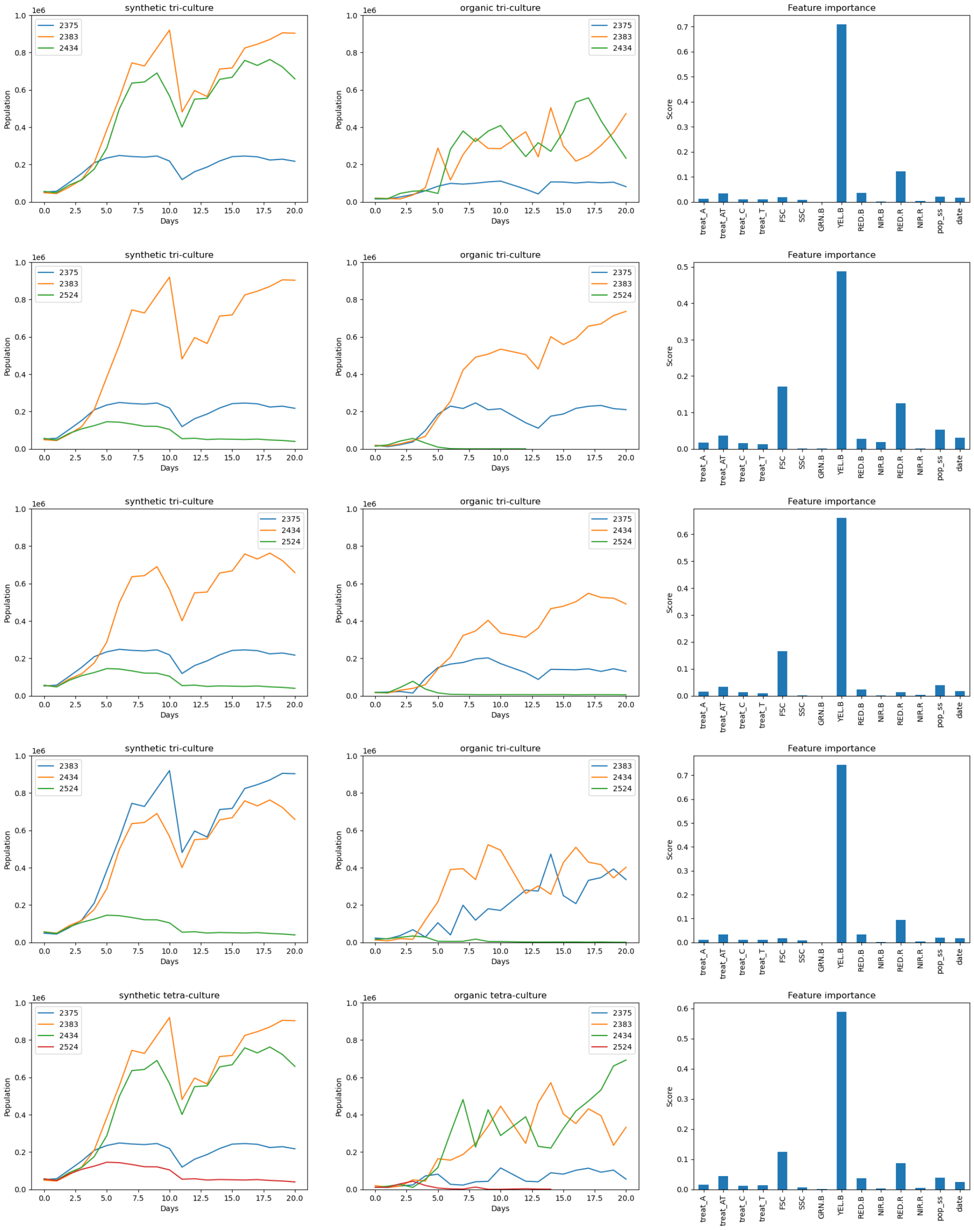

Figure 25: Population evolution of synthetic tri-cultures and tetra-culture (left) and organic tri-cultures and tetra-culture (middle) resulting from allocating the actual tri-culture and tetra-culture population using strain membership predictions of cells using xgboost classifiers, and the feature scores of the predictors, including population, used in each classifier (right).

## A.2 Selected Population Plots of Proposed Strategies

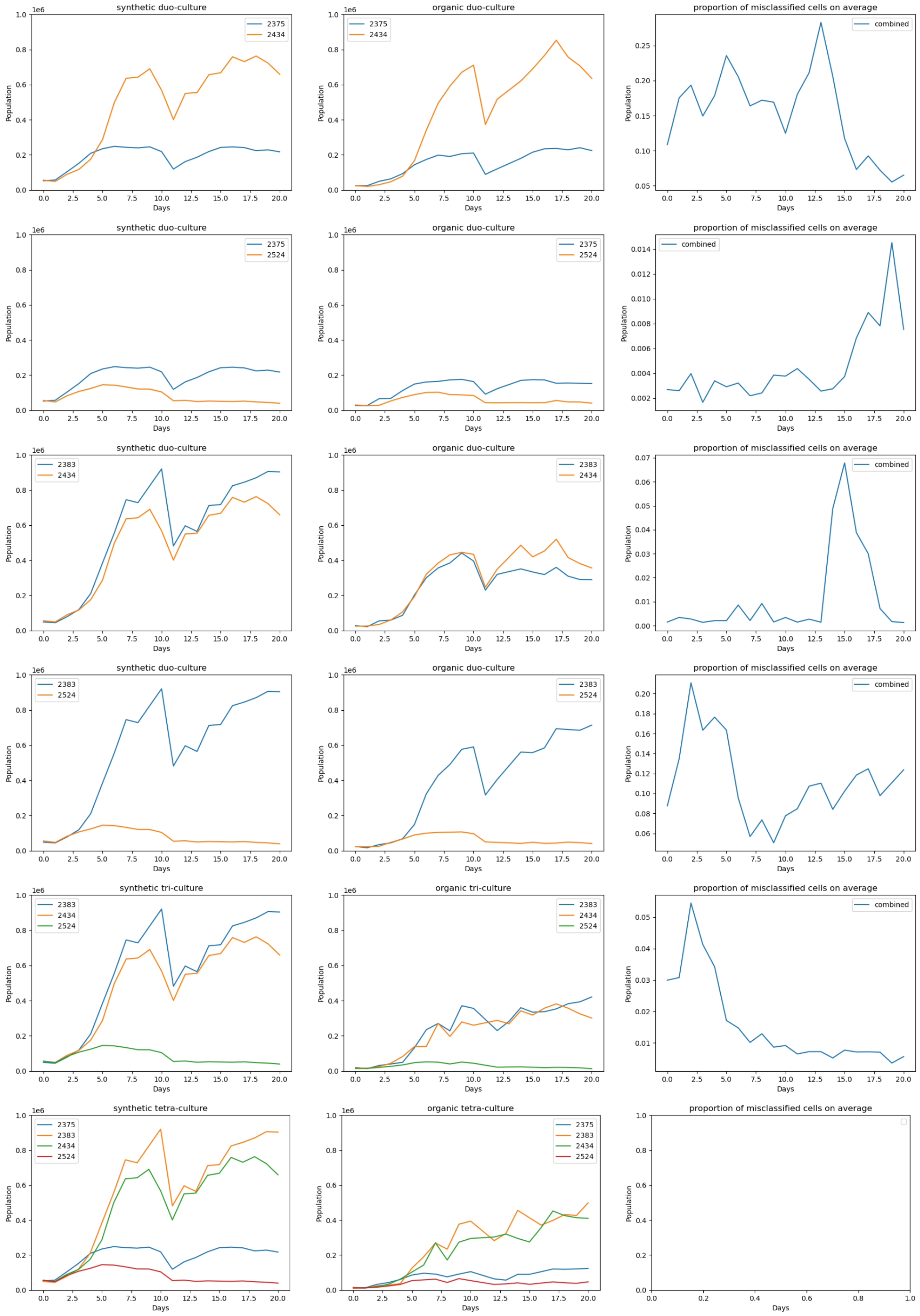

Figure 26: Population evolution of synthetic multi-cultures (left) and organic multi-cultures (middle) resulting from allocating the actual multi-culture population using strain membership predictions of cells using strategy o, and the proportion of population assigned to strains not part of the organic multi-culture (right).

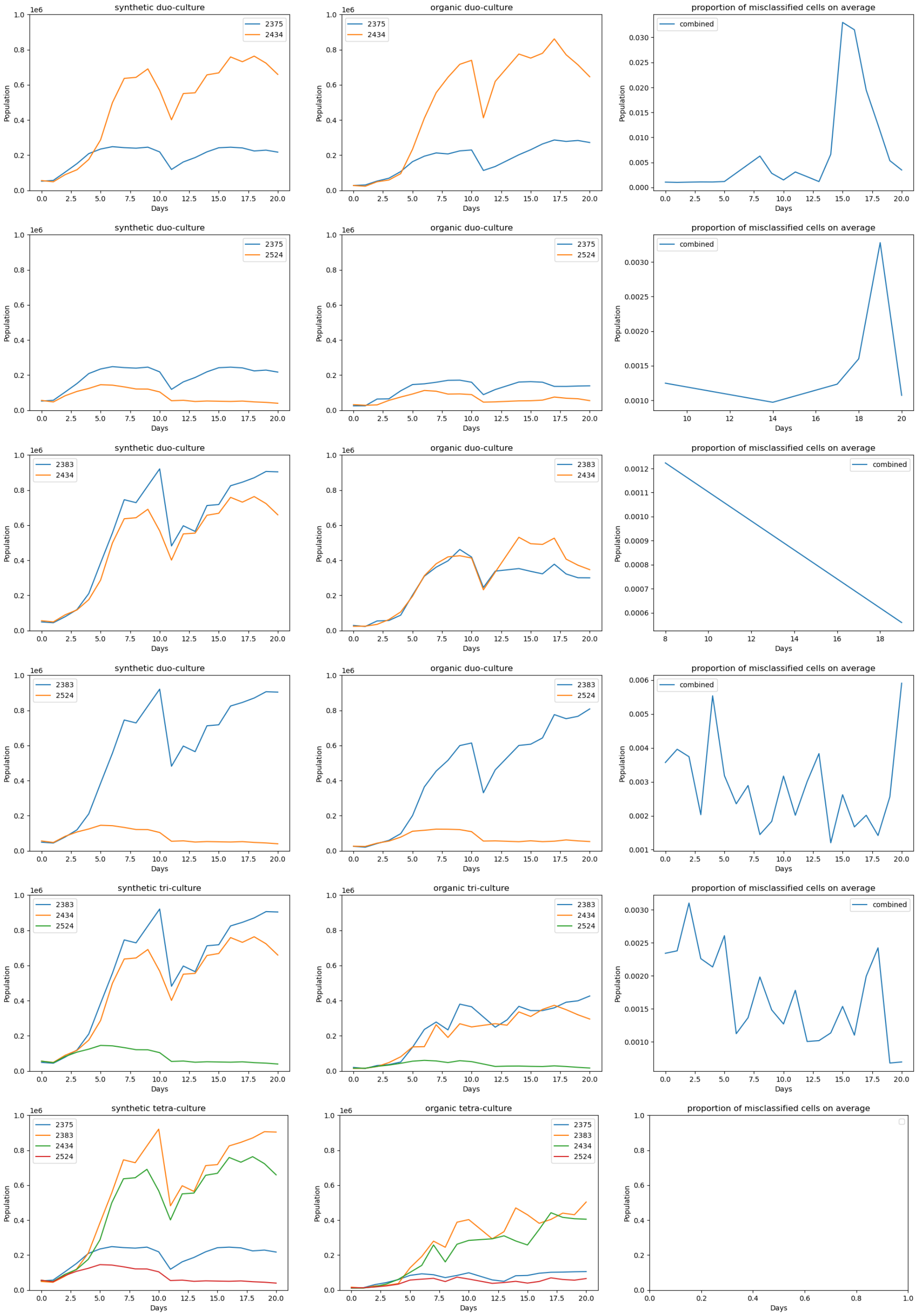

Figure 27: Population evolution of synthetic multi-cultures (left) and organic multi-cultures (middle) resulting from allocating the actual multi-culture population using strain membership predictions of cells using strategy I, and the proportion of population assigned to strains not part of the organic multi-culture (right).

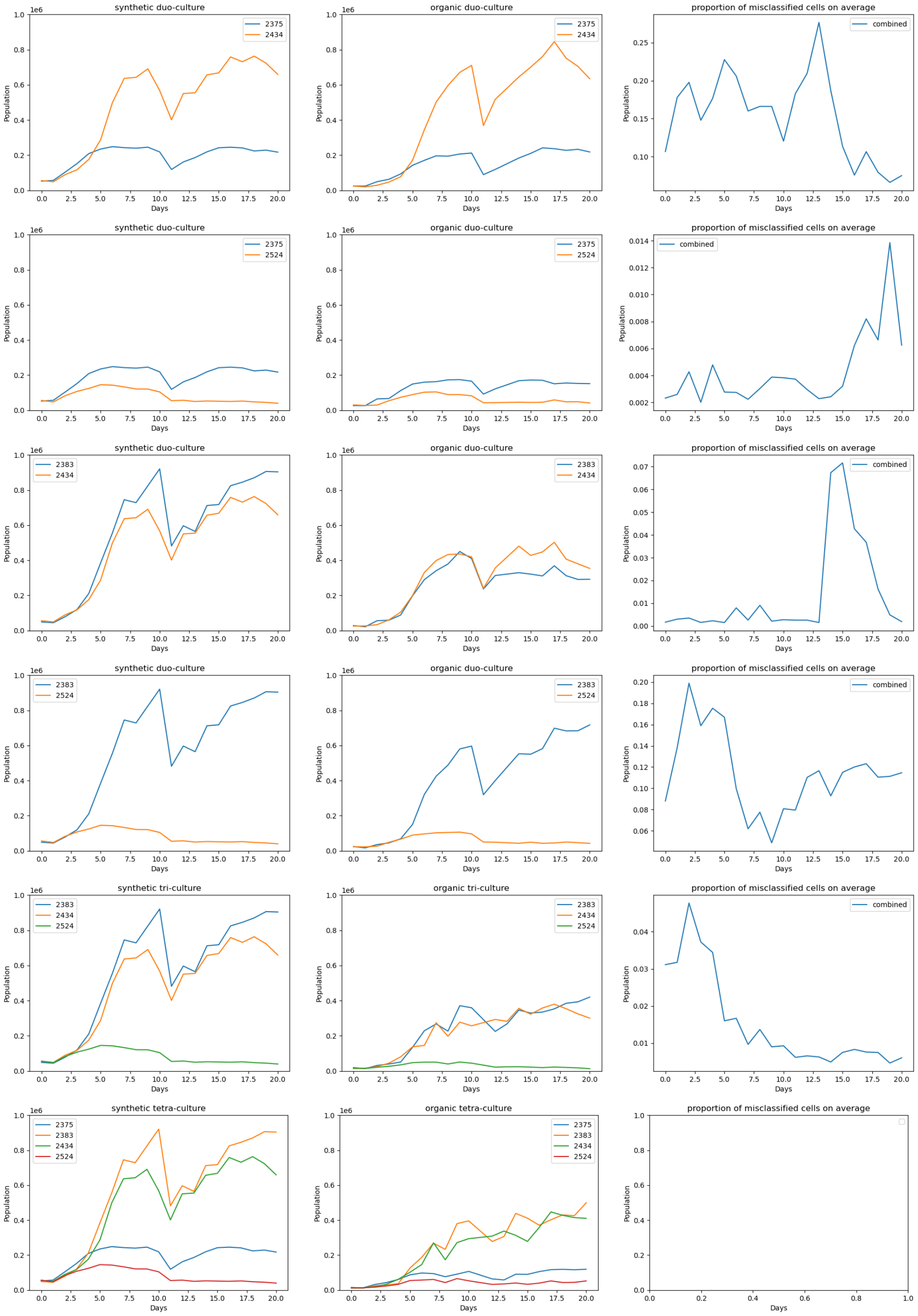

Figure 28: Population evolution of synthetic multi-cultures (left) and organic multi-cultures (middle) resulting from allocating the actual multi-culture population using strain membership predictions of cells using strategy II, and the proportion of population assigned to strains not part of the organic multi-culture (right).

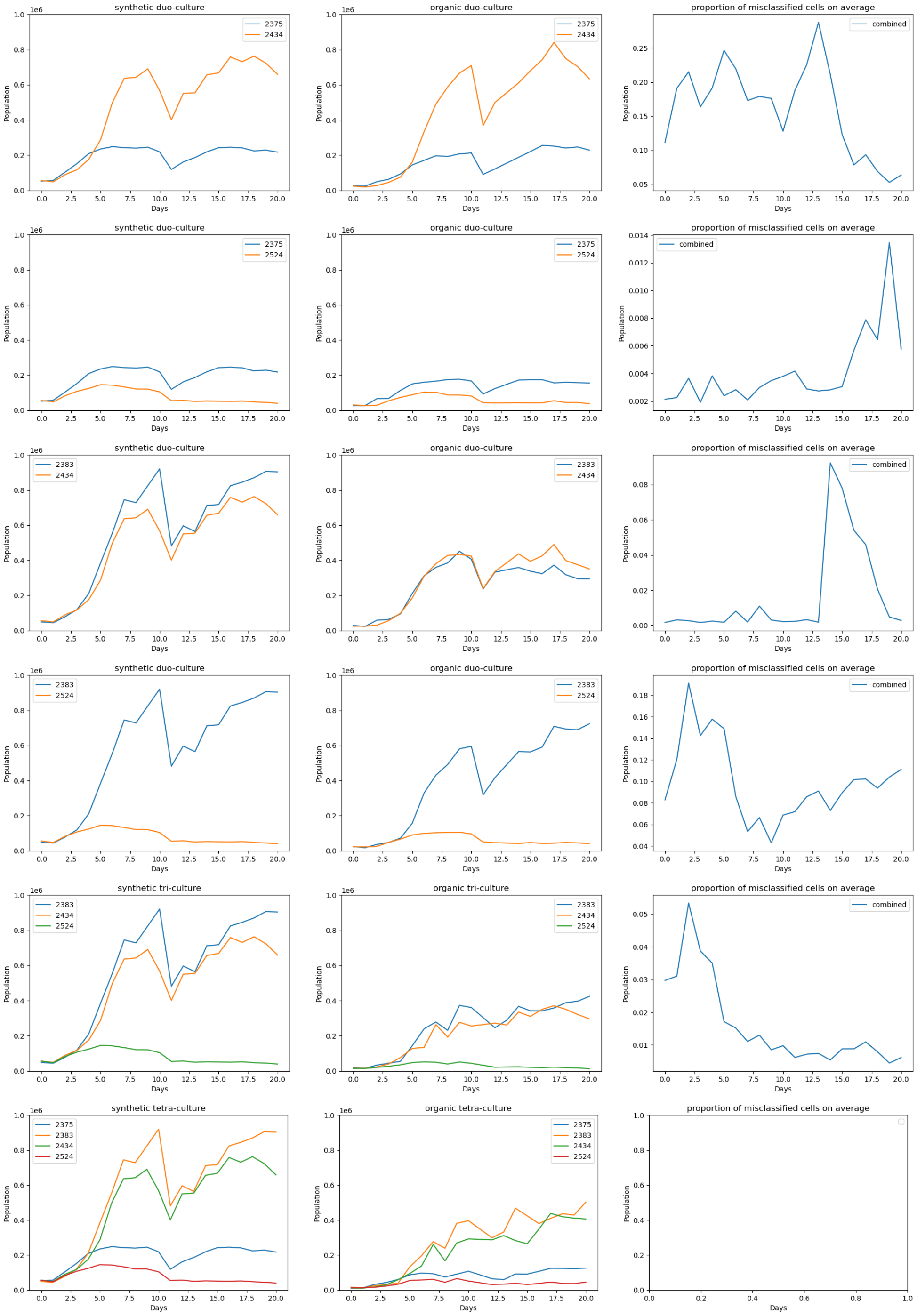

Figure 29: Population evolution of synthetic multi-cultures (left) and organic multi-cultures (middle) resulting from allocating the actual multi-culture population using strain membership predictions of cells using strategy III, and the proportion of population assigned to strains not part of the organic multi-culture (right).

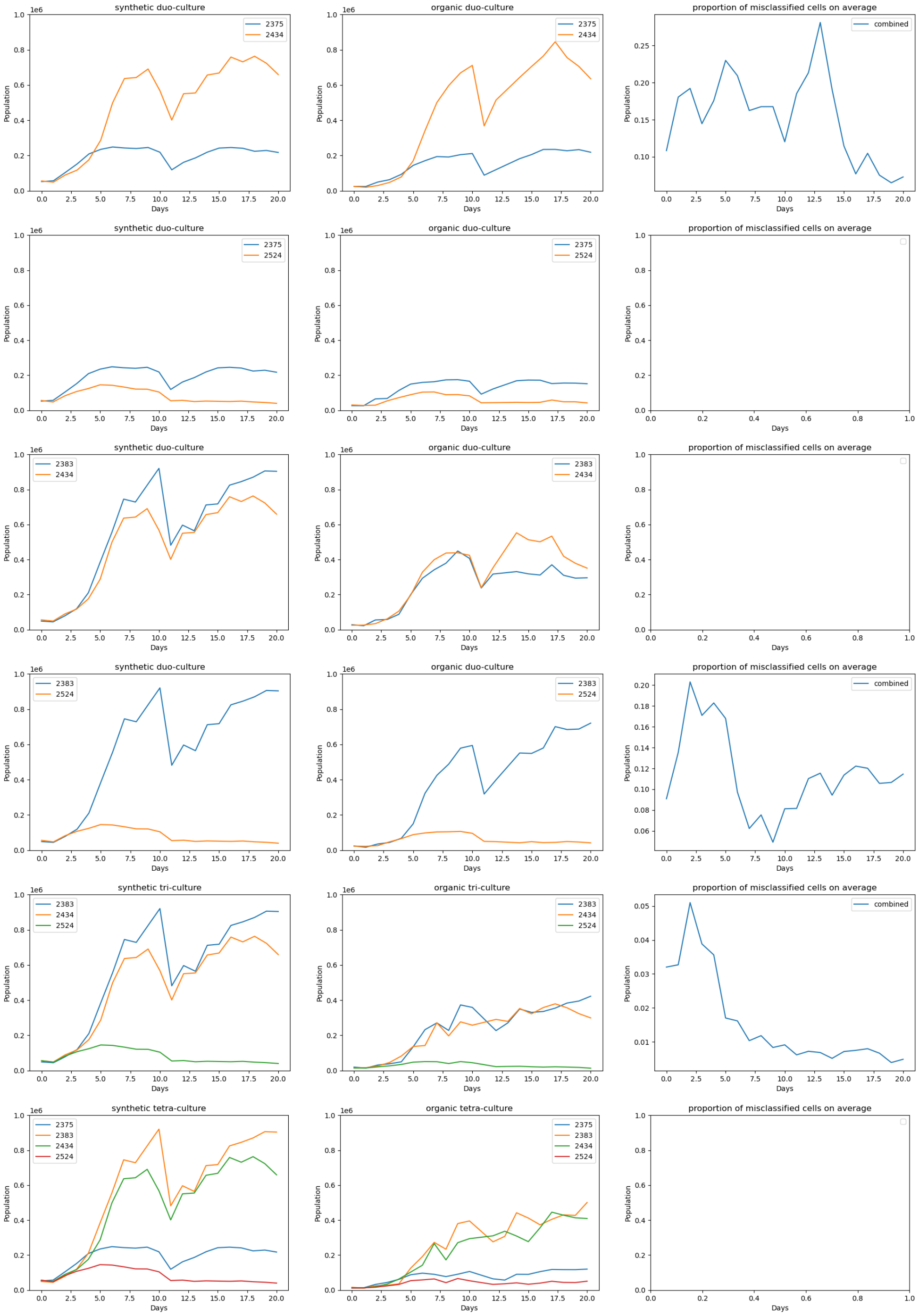

Figure 30: Population evolution of synthetic multi-cultures (left) and organic multi-cultures (middle) resulting from allocating the actual multi-culture population using strain membership predictions of cells using strategy IV, and the proportion of population assigned to strains not part of the organic multi-culture (right).

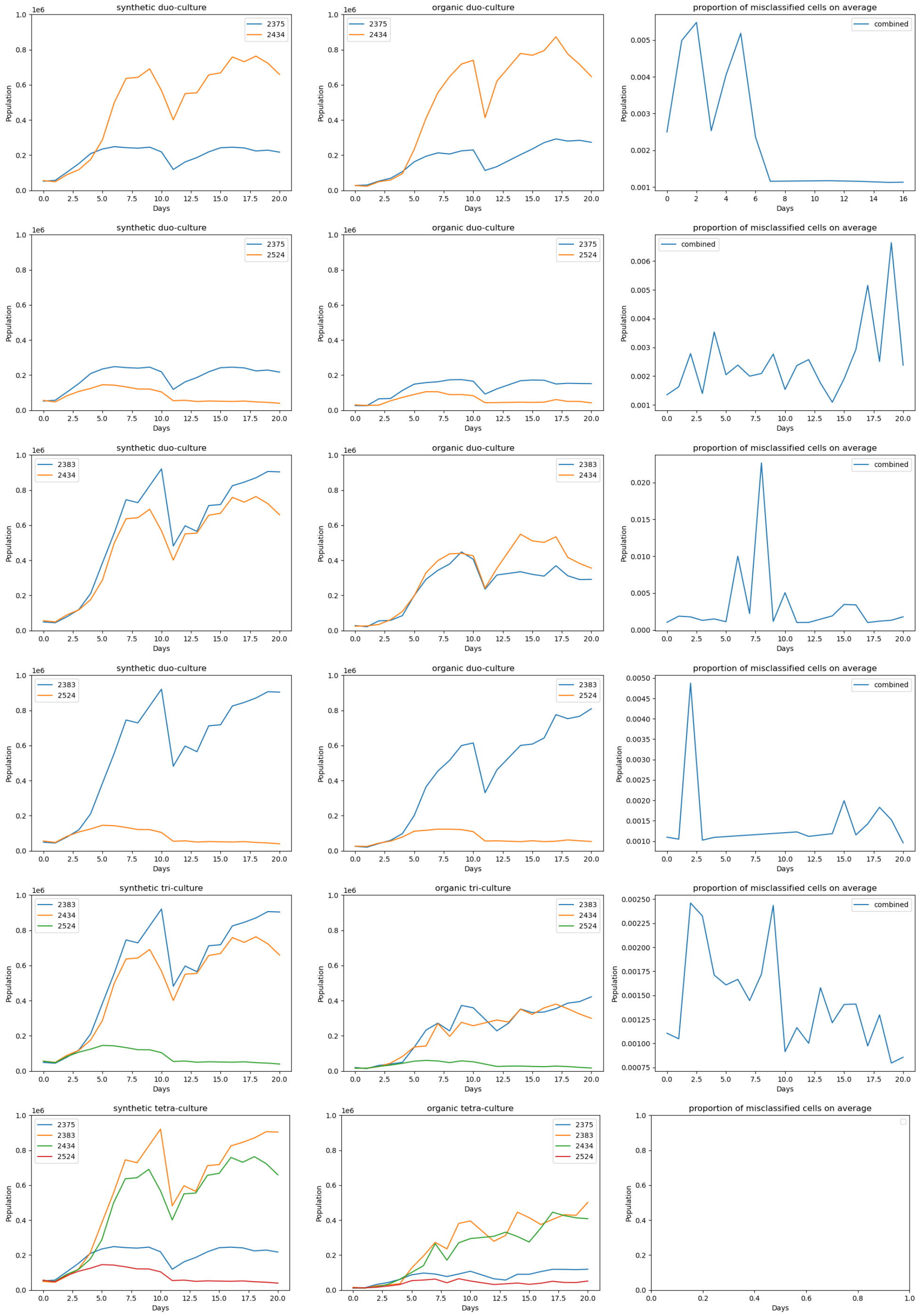

Figure 31: Population evolution of synthetic multi-cultures (left) and organic multi-cultures (middle) resulting from allocating the actual multi-culture population using strain membership predictions of cells using strategy V, and the proportion of population assigned to strains not part of the organic multi-culture (right).

### A.3 Relevant Python Codes

```
def run_trial(df_dict, mono_strains, combi_strains, combi_type,
              clf, inc_pop=False, show_bplots=False,
              encode_output=False, input_columns=None):

    if inc_pop:
        scale_cols = laser_columns + ['pop']
    else:
        scale_cols = laser_columns

    strain_map = {strain:i for i, strain in
        enumerate(mono_strains)}

    encoder, scaler = (None, None)
    new_encoded_cols, new_scaled_cols = ([], [])
    for type_ in df_dict.keys():
        strain_filters = ('strains', combi_strains) if type_ ==
            'combi' else ('strain', mono_strains)
        df_dict[type_], encoder, scaler, new_encoded_cols,
            new_scaled_cols = aux.preprocess_data(
            df_dict[type_], encoder=encoder, scaler=scaler,
            encode_cols=categorical_columns,
            scale_cols=scale_cols,
            strain_filters=strain_filters)
        df_dict[type_] = pd.concat([df_dict[type_],
            pd.DataFrame(np.zeros((df_dict[type_].shape[0],
            len(all_strains))), columns=all_strains)], axis=1)
        if encode_output and type_ in ['train', 'val', 'test']:
            df_dict[type_]['strain_alt'] = df_dict[type_]
            ['strain'].map(strain_map)


    input_columns = new_encoded_cols + new_scaled_cols +
        ['date'] if input_columns is None else input_columns
    output_column = 'strain_alt' if encode_output else 'strain'

    acc_dict = {}
    clf, acc_dict['train'] = aux.train(clf, df_dict['train']
        [input_columns], df_dict['train']
        [[output_column]].iloc[:, 0], v=False)
    for type_ in ['train', 'val', 'test']:
        acc_dict[type_] = aux.evaluate(clf, df_dict[type_]
            [input_columns], df_dict[type_]
            [[output_column]].iloc[:, 0], t=type_)
        print(type_)
        for strain in mono_strains:
            cur_acc = aux.evaluate(clf,
            df_dict[type_].loc[df_dict[type_]
            ["strain"].isin([strain])][input_columns],
                            df_dict[type_].loc[df_dict[type_]["strain"]
                            .isin([strain])][[output_column]].iloc[:, 0],
                            t=type_)
            print(f'{strain}: {cur_acc}')
            preds_ = pd.Series(clf.predict(df_dict[type_]
                .loc[df_dict[type_]["strain"].isin([strain])][input_columns]))
            if encode_output:
                preds_ = preds_.apply(lambda x: mono_strains[x])
            print(preds_.value_counts().sort_index())


    df_dict['combi'][output_column] = clf.predict(df_dict['combi'][input_columns])

    if encode_output:
        df_dict['combi']['strain'] = df_dict['combi'][output_column]
            .map(lambda x: mono_strains[x])

    for type_ in df_dict.keys():
        temp_pred_proba = np.transpose(clf.predict_proba
            (df_dict[type_][input_columns]))
        for i, strain in enumerate(mono_strains):
            df_dict[type_][strain] = temp_pred_proba[i]

    if show_bplots:
        df_dict['train']['exp_type'] = np.repeat
            ('mono', df_dict['train'].shape[0])
        df_dict['combi']['exp_type'] = np.repeat
            (df_dict['combi'], df_dict['combi'].shape[0])
        aux.show_boxplots(3, 3, laser_columns, pd.concat
            ([df_dict['train'], df_dict['combie']], axis=0),
            [’strain’, ’exp_type’], [(strain, exp_type) for
            strain in mono_strains for exp_type in [’mono’,
            combi_type]])


    feature_importances = pd.DataFrame()
    input_columns_alt = [input_column.replace(’.HLin_ss’, ’’)
        for input_column in input_columns]
    if isinstance(clf, XGBClassifier):
        feature_importances[’mean’] =
            pd.Series(clf.feature_importances_, index=input_columns_alt)

    elif isinstance(clf, RandomForestClassifier):
        result = permutation_importance(
            clf, df_dict[’test’][input_columns], df_dict[’test’]
            [[output_column]].iloc[:, 0],
            n_repeats=10, random_state=1010, n_jobs=4
        )
        feature_importances[’mean’] =
            pd.Series(result.importances_mean, index=input_columns_alt)
        feature_importances[’std’] =
            pd.Series(result.importances_std, index=input_columns_alt)
    return clf, encoder, scaler, df_dict, acc_dict, feature_importances


def get_clf(clf_opt=’knn’):
    if clf_opt == ’xgbc’:
        return XGBClassifier(n_jobs=4, random_state=1010)
    elif clf_opt == ’rfc’:
        return RandomForestClassifier(n_jobs=4, random_state=1010)
    else:
        return KNeighborsClassifier(n_jobs=4)

def run_trial_for_exp(exp_type, clf_opt=’knn’, inc_pop=False,
                       save_dfs=False, save_models=False, encode_output=False):

    acc_dict_list = {’strains’:strain_combi[exp_type], ’train’:[],
    ’val’:[], ’test’:[]}
    for strains in strain_combi[exp_type]:
        strain_list = aux.strain_to_list(strains)
```

```
clf = get_clf(clf_opt)
model, encoder, scaler, df_dict, acc_dict, feature_importances
    = run_trial({'train':df_mono_train,
    'val':df_mono_val, 'test':df_mono_test,
    'combi':df_exp[exp_type]}, strain_list, [strains],
    exp_type, clf, show_bplots=False, inc_pop=inc_pop,
    encode_output=encode_output)

fig, ax = plt.subplots(1, 3, figsize=(15*1.5, 3*1.5))

df_temp_1, pop_col_1, _ = aux.calculate_pop(df_dict['train'],
    mono_data=True, strain_col='strain')
df_temp_2, pop_col_2, _ = aux.calculate_pop(df_dict['combi'],
    mono_data=False, strain_col='strain')

aux.time_series_plot(df_temp_1, ax=ax[0], pop_col=pop_col_1)
aux.time_series_plot(df_temp_2, ax=ax[1], pop_col=pop_col_2,
dates_to_exclude=aux.dates_to_exclude[strains])

ax[0].set_title(f'synthetic {exp_type}-culture')
ax[1].set_title(f'organic {exp_type}-culture')
ax[0].set_ylim(0, 1e6)
ax[1].set_ylim(0, 1e6)

#print(feature_importances)
feature_importances['mean'].plot.bar(yerr=feature_importances['std']
    if clf_opt == 'rfc' else None, ax=ax[2])
ax[2].set_title("Feature importance")
ax[2].set_ylabel("Score")
#fig.tight_layout()
#plt.show()

for key in acc_dict.keys():
    acc_dict_list[key].append(acc_dict[key])

if save_dfs:
    dirname = 'data/base_model/'
    for key, df_ in df_dict.items():
        df_.to_csv(f'{dirname}{key}_{strains}
                {"_with_pop" if inc_pop else ""}.csv’, index=False)

        if save_models:

            aux.save_model(model,
                f’base_models/{type(clf)}_{exp_type}_{strains}.pkl’)
            aux.save_model(scaler,
                f’scalers/scaler_{exp_type}_{strains}.pkl’)
            aux.save_model(encoder,
                f’encoders/encoder_{exp_type}_{strains}.pkl’)

    print(pd.DataFrame(acc_dict_list))

#Base models
run_trial_for_exp(’duo’, clf_opt=’rfc’)
run_trial_for_exp(’duo’, clf_opt=’xgbc’, encode_output=True)
run_trial_for_exp(’tri’, clf_opt=’rfc’)
run_trial_for_exp(’tri’, clf_opt=’xgbc’, encode_output=True)
run_trial_for_exp(’tetra’, clf_opt=’rfc’)
run_trial_for_exp(’tetra’, clf_opt=’xgbc’, encode_output=True)

#Strat 0
exp_type = ’tetra’
strat_o_model, encoder, scaler, df_dict, acc_dict, feature_importances =
    run_trial({’train’:df_mono_train, ’val’:df_mono_val, ’test’:df_mono_test,
    ’combi’:df_exp[exp_type]}, all_strains, strain_combi[exp_type], exp_type,
    clf=XGBClassifier(n_jobs=4, random_state=1010), show_bplots=False,
    encode_output=True, inc_pop=False)

for type_ in [’train’, ’val’, ’test’]:
    for strain in all_strains:
        df_temp = df_dict[type_].loc[df_dict[type_][’strain’].isin([strain])]
        aux.evaluate(strat_o_model, df_temp[inp])

df_combi = pd.concat([df_exp[exp_type] for exp_type in strain_combi.keys()],
    axis=0).reset_index().iloc[:,1:]
df_combi, encoder, scaler, new_encoded_cols, new_scaled_cols =
    aux.preprocess_data(df_combi, encoder=encoder, scaler=scaler,
    encode_cols=categorical_columns, scale_cols=laser_columns)
input_columns = new_encoded_cols + new_scaled_cols + [’date’]
```

```
output_columns = ['strain']
predictions = pd.Series(strat_o_model.predict(
    df_combi[input_columns]), name='strain')
predictions = predictions.map(lambda x: all_strains[x])
strat_o_plot_data = aux.assess_discrimination(
    predictions, df_mono_train, df_combi, strain_combi)

#Strat I
inputs = keras.Input(shape=(len(input_columns),))
x = layers.Dense(units=32, activation='relu')(inputs)
x = layers.Dense(units=64, activation='relu')(x)
outputs = layers.Dense(units=len(all_strains), activation='softmax')(x)

model = keras.Model(inputs, outputs)

output_encoder = layers.IntegerLookup(vocabulary=all_strains,
    num_oov_indices=0)

model.compile(optimizer='rmsprop',
    loss='sparse_categorical_crossentropy', metrics=['accuracy'])

callbacks = [keras.callbacks.ModelCheckpoint(
        filepath='strat_1_model.keras',
        save_best_only=True,
        monitor='val_loss'
    )]

model.fit(
        df_dict['train'][input_columns],
        output_encoder(df_dict['train'][output_columns]),
        epochs=20,
        validation_data=(df_dict['val'][input_columns],
        output_encoder(df_dict['val'][output_columns])),
        callbacks=callbacks,
        verbose=True)

callback_model = keras.models.load_model('strat_1_model.keras')

for type_ in ['train', 'val', 'test']:
    print(type_)
    print(callback_model.evaluate(df_dict[type_][input_columns],
    output_encoder(df_dict[type_][output_columns])))

combi_predictions = callback_model.predict(df_dict[’combi’][input_columns])
combi_predictions = pd.Series([all_strains[combi_prediction.argmax()]
    for combi_prediction in combi_predictions], name=’strain’)
strat_1_plot_data = aux.assess_discrimination(combi_predictions,
    df_mono_train, df_dict[’combi’], strain_combi)

#Strat II
def extract_features(df):
    new_features = []
    for exp_type, strains_list in strain_combi.items():
        for strains in strains_list:
            e_ = aux.load_model(f’encoders/encoder_{exp_type}_{strains}.pkl’)
            s_ = aux.load_model(f’scalers/scaler_{exp_type}_{strains}.pkl’)
            model_ = aux.load_model(f"base_models/<class
                ’xgboost.sklearn.XGBClassifier’>_{exp_type}_{strains}.pkl")

            df_, _, _, new_encoded_cols, new_scaled_cols =
                aux.preprocess_data(df, encoder=e_, scaler=s_,
                encode_cols=categorical_columns, scale_cols=laser_columns)
            new_features.append(pd.DataFrame(model_.
                predict_proba(df_[new_encoded_cols + new_scaled_cols
                + [’date’]]), columns=[f’{strains}.{strain}’
                for strain in aux.strain_to_list(strains)]))
    return pd.concat(new_features, axis=1)


strain_dict = {strain:i for i, strain in enumerate(all_strains)}

encoder, scaler = (None, None)
new_encoded_cols, new_scaled_cols = ([], [])
for type_ in df_dict.keys():
    df_temp = pd.DataFrame()
    df_temp[laser_columns + categorical_columns +
        [’date’, ’repl’, ’pop’]] = df_dict[type_][laser_columns
        + categorical_columns + [’date’, ’repl’, ’pop’]]
    if type_ in [’train’, ’val’, ’test’]:
        df_temp[’strain’] = df_dict[type_][’strain’]
        df_temp['strain_sparse'] = df_dict[type_]['strain'].
            map(lambda x: strain_dict[x])
    else:
        df_temp['strains'] = df_dict[type_]['strains']

    df_temp, encoder, scaler, new_encoded_cols, new_scaled_cols =
        aux.preprocess_data(
        df_temp, encoder=encoder, scaler=scaler,
        encode_cols=categorical_columns, scale_cols=laser_columns)

    df_new_features = extract_features(df_dict[type_])
    df_dict[type_] = pd.concat([df_temp, df_new_features], axis=1)

aux.save_model_(df_dict, 'data/strat_ii/strat_ii_df_dict.pkl')

acc_dict = {}
output_columns = ['strain_sparse']

clf_xgbc, acc_dict['train'] = aux.train(XGBClassifier(n_jobs=4,
    random_state=1010), df_dict['train'][input_columns],
    df_dict['train'][output_columns].iloc[:, 0], v=False)
for type_ in ['val', 'test']:
    acc_dict[type_] = aux.evaluate(clf_xgbc, df_dict[type_]
        [input_columns], df_dict[type_][output_columns].
        iloc[:, 0], v=False, t=type_)

xgbc_predictions = pd.Series(clf_xgbc.
    predict(df_dict['combi'][input_columns]), name='strain')
xgbc_predictions = xgbc_predictions.map(lambda x: all_strains[x])
strat_ii_xgbc_plot_dat = aux.assess_discrimination(xgbc_predictions,
    df_mono_train, df_dict['combi'], strain_combi)

#Strat III
all_new_features = []
for exp_type, strains_list in strain_combi.items():
    for strains in strains_list:
        new_features = []
        for strain in aux.strain_to_list(strains):
            new_features.append(f'{strains}.{strain}')
        all_new_features.append(new_features)
```

```
all_new_features

inputs = []
xs = []
for new_features in all_new_features:
    input = Input(shape=(len(new_features),))
    x = input
    for size in [32, 32]:
        x = keras.layers.BatchNormalization()(x)
        x = keras.layers.Dense(units=size, activation='relu')(x)
    inputs.append(input)
    xs.append(x)

x = keras.layers.concatenate(xs)
outputs = keras.layers.Dense(units=4, activation='softmax')(x)

strat_3_model = keras.Model(inputs, outputs)

strat_3_model.compile(optimizer=keras.optimizers.
    RMSprop(learning_rate=0.0005), loss='sparse_categorical_crossentropy', metrics=['accurac

callbacks = [keras.callbacks.ModelCheckpoint(
        filepath='strat_3_model.keras',
        save_best_only=True,
        monitor='val_loss'
    )]

strat_3_model.fit(
        [df_dict['train'][new_features] for new_features in all_new_features],
        df_dict['train']['strain_sparse'],
        epochs=10,
        validation_data=([df_dict['val'][new_features]
        for new_features in all_new_features], df_dict['val']['strain_sparse']),
        callbacks=callbacks,
        verbose=True)

callback_model = load_model('strat_3_model.keras')

for type_ in ['train', 'val', 'test']:
    print(type_)
    print(callback_model.evaluate([df_dict[type_][new_features]
        for new_features in all_new_features], df_dict[type_]['strain_sparse']))

predictions = callback_model.predict([df_dict['combi'][new_features]
    for new_features in all_new_features])
predictions = pd.Series([all_strains[prediction.argmax()]
    for prediction in predictions], name='strain')

strat_iii_nn_plot_data = aux.assess_discrimination(predictions,
    df_mono_train, df_dict['combi'], strain_combi)

#Strat IV
acc_dict = {}
output_columns = 'species'

clf_xgbc_spec, acc_dict['train'] = aux.train(XGBClassifier(n_jobs=4,
    random_state=4), df_dict['train'][input_columns],
    df_dict['train'][output_columns], v=False)

for type_ in ['val', 'test']:
    acc_dict[type_] = aux.evaluate(clf_xgbc_spec,
        df_dict[type_][input_columns],
        df_dict[type_][output_columns], v=False, t=type_)
print(f'{acc_dict}')

acc_dict = {}
output_columns = 'strain'

output_encoder = IntegerLookup(vocabulary=all_strains, num_oov_indices=0)

clf_xgbc_strain, acc_dict['train'] = aux.train(
    XGBClassifier(n_jobs=4, random_state=1010),
    df_dict['train'][input_columns + input_columns_extra
    + ['species']], output_encoder(df_dict['train'][output_columns]), v=False)

for type_ in ['val', 'test']:
    acc_dict[type_] = aux.evaluate(clf_xgbc_strain,
    df_dict[type_][input_columns + input_columns_extra + ['species']],
    output_encoder(df_dict[type_][output_columns]), v=False, t=type_)
print(f'{acc_dict}')
```

```
def get_species(row, strains, species_dict, special_strains, clf_spec):
    if strains in special_strains:
        return species_dict[aux.strain_to_list(strains)[0]]
    return clf_spec.predict(row)

df_dict['combi']['species'] = clf_xgbc_spec.
    predict(df_dict['combi'][input_columns])

special_strains = {'2375_2524', '2383_2434'}
list_of_df = []

for exp_type, strain_list in strain_combi.items():
    for strains in strain_list:
        if strains in special_strains:
            df_dict['combi'].loc[df_dict['combi']['strains']
                .isin([strains]), 'species'] =
                np.repeat(species_dict[aux.strain_to_list(strains)[0]],
                df_dict['combi'].loc[df_dict['combi']['strains']
                .isin([strains])].shape[0])
        else:
            df_dict['combi'].loc[df_dict['combi']['strains']
                .isin([strains]), 'species'] = clf_xgbc_spec
                .predict(df_dict['combi'].loc[df_dict['combi']['strains']
                .isin([strains])][input_columns])

    predictions =  pd.Series(clf_xgbc_strain
        .predict(df_dict['combi'][input_columns + input_columns_extra
        + ['species']]), name='strain')

predictions = predictions.map(lambda x: all_strains[x])
strat_iv_plot_data = aux.assess_discrimination(predictions,
    df_mono_train, df_dict['combi'], strain_combi)

#Strat V
new_columns = ['2375_2383', '2375_2434', '2375_2524',
    '2383_2434', '2383_2524', '2434_2524']
for type_ in ['train', 'val', 'test']:
    list_new_df = []
    for strain in all_strains:
        for exp_type, strain_list in strain_combi.items():
            for strains in strain_list:
                if str(strain) not in strains:
                    continue
                temp_df = df_dict[type_].loc[df_dict[type_]['strain']
                    .isin([strain])].copy()
                temp_df.loc[:, ['strains']] = strains
                strain_set = set(aux.strain_to_list(strains))
                for new_column in new_columns:
                    new_col_set = set(aux.strain_to_list(new_column))
                    temp_df.loc[:, [new_column]] = len(strain_set
                        .intersection(new_col_set)) / 2
                list_new_df.append(temp_df)
    df_dict[type_] = pd.concat(list_new_df).reset_index().iloc[:,1:]

for exp_type, strain_list in strain_combi.items():
    for strains in strain_list:
        strain_set = set(aux.strain_to_list(strains))
        for new_column in new_columns:
            new_col_set = set(aux.strain_to_list(new_column))
            df_dict['combi'].loc[df_dict['combi']['strains']
                .isin([strains]), [new_column]]
                = len(strain_set.intersection(new_col_set)) / 2

acc_dict = {}
output_columns = ['strain_sparse']
new_input_columns = input_columns + pred_columns + new_columns

clf_xgbc, acc_dict['train'] = aux.train(XGBClassifier(n_jobs=4,
    random_state=1010),
    df_dict['train'][new_input_columns], df_dict['train'][output_columns]
    .iloc[:, 0], v=False)
for type_ in ['val', 'test']:
    acc_dict[type_] = aux.evaluate(clf_xgbc,
        df_dict[type_][new_input_columns], df_dict[type_][output_columns]
        .iloc[:, 0], v=False, t=type_)

acc_dict

xgbc_predictions = pd.Series(clf_xgbc.predict
    (df_dict['combi'][new_input_columns]), name='strain')
xgbc_predictions = xgbc_predictions.map(lambda x: all_strains[x])
strat_v_xgbc_plot_dat = aux.assess_discrimination(xgbc_predictions,
    df_mono_train, df_dict['combi'], strain_combi)
aux.save_model_(strat_v_xgbc_plot_dat, 'data/plot/strat_v_xgbc_plot_data.pkl')
```

### A.4 Relevant SAS Codes

```
PROC MIXED DATA=WORK.IMPORT METHOD=ML;
 CLASS id sample_id repl(ref=’1’) strain(ref=’2375’)
    treat_alt freq(ref=’FSC.HLin’);
 MODEL cytout = date pop repl treat_alt strain freq
    freq*date freq*pop freq*treat_alt freq*strain  / SOLUTION COVB;
 REPEATED freq / SUBJECT=id type=un rcorr;
 ODS OUTPUT COVPARMS=covparms;
RUN;

PROC MIXED DATA=WORK.IMPORT METHOD=ML;
 CLASS id sample_id repl(ref=’1’) cult_type(ref=’organic’)
    treat_alt freq(ref=’FSC.HLin’);
 MODEL cytout = date pop repl treat_alt cult_type freq
    freq*date freq*cult_type freq*pop freq*treat_alt / SOLUTION COVB;
 PARMS / PARMSDATA=work.covparms;
 REPEATED freq / SUBJECT=id type=un rcorr;
 ODS OUTPUT SolutionF=stat_anal_2;
RUN;

PROC MIXED DATA=WORK.IMPORT METHOD=ML;
 CLASS id sample_id repl(ref=’1’) membership(ref=’actual’)
    treat_alt repl freq(ref=’FSC.HLin’);
 MODEL cytout = date pop repl treat_alt membership freq
    freq*date freq*pop  / SOLUTION;
 REPEATED freq / SUBJECT=id type=un rcorr;
 PARMS / PARMSDATA=work.cov_parms_in;
 ODS OUTPUT SolutionF=stat_anal_3_date_mem_pop;
 ODS OUTPUT COVPARMS=cov_parms_out;
RUN;
```

### A.5 Relevant R Codes

```
check_contrast <- function(beta_, var_cov_mat_, cont_, p_, n_){
  idx <- c(3, 6, 8, 5, 7, 2, 4, 1)
  beta_ <- matrix(beta_, p_, 1)
  var_cov_mat_ <- matrix(var_cov_mat_, p_, p_)
  cont_ <- t(matrix(cont_, p_, 8))
  estimate <- cont_ %*% beta_
  stderr <- diag(sqrt(cont_ %*% var_cov_mat_ %*% t(cont_)))
  df <- round(data.frame(idx, estimate, stderr), 4)
  df$tval <- round(df$estimate/df$stderr, 4)
  df$pval <- 1-pnorm(abs(df$tval))
  df$pval_adj <- round(p.adjust(df$pval, method='BH', n=n_), 5)
  df$pval <- round(df$pval, 5)
  df <- df[order(df$idx),]
  df
}
```

# B Acknowledgements

I express my gratitude to my advisers, Prof. dr. Frédérik de Laender and Prof. dr. Olivier Thas, for choosing me to do this study and for their guidance and supervision that allowed me to work seamlessly on this project. I also would like to extend my thanks to my Filipino community in Belgium, both in Leuven and in Hasselt, for their unwavering support. To Roel and Riri, who joined me for lunch during the last semester, thank you. To Grace and Neil, with whom I was the closest in this program, thank you. To Rico, Thea, Ate P, and Yam, my Leuven family, who knew about my endeavors and who always willingly lend me a hand, thank you. To my family in the Philippines, who never gets tired every time I announce my plans to study again, I love you. And lastly to myself, for holding on and hanging in there, I am proud of you.