Article

A Comparison of Clustering and Prediction Methods for Identifying Key Chemical-Biological Features Affecting Bioreactor Performance

Yiting Tsai [1,†,‡], Sue Baldwin [2,‡], Lim C. Siang [3,], and Bhushan Gopaluni [4,*]

[1] Affiliation 1; yttsai@chbe.ubc.ca
[2] Affiliation 2; sue.baldwin@ubc.ca
[3] Affiliation 3; siang@alumni.ubc.ca
[4] Affiliation 4; bhushan.gopaluni@chbe.ubc.ca
[*] Correspondence: yttsai@chbe.ubc.ca; Tel.: +1 604 822 3238
[†] Current address: Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC V6T 1Z3, Canada

Version September 2, 2019 submitted to Journal Not Specified

Abstract: Chemical-biological systems, such as bioreactors, contain stochastic and non-linear interactions which are difficult to characterize. The highly complex interactions between microbial species and communities may not be sufficiently captured using first-principles, stationary, or low-dimensional models. This paper explores a data analysis strategy, which combines three predictive models (Random Forests, Support Vector Machines, and Neural Networks), three clustering models (hierarchical, Gaussian mixtures, and Dirichlet mixtures), and two feature selection approaches (Mean Decrease in Accuracy and its conditional variant). By doing so, the outcome of a bioreactor is not only predicted with high accuracy, but the important features correlated with said outcome are also identified. The novelty of this work lies in the extensive compare-and-contrast of a wide arsenal of methods, as opposed to single methods which are often observed in papers in similar fields. The results of this work show that Random Forest models predict test set outcomes with the highest accuracy. Moreover, although the clustering methods successfully identified groups of microbial species and their leaders, the groups are inconsistent when compared across the three clustering methods. Finally, the two feature selection methods identified key variables features which agree with a domain-knowledge understanding of the bioreactor system. Overall, the results indicate that although it is possible to perform simultaneous analysis with chemical and biological data, the clustering and feature analysis methods must be further refined for consistency and robustness.

Keywords: Machine Learning; bioinformatics; statistics

1 Introduction and Literature Review

Process control in the chemical and biological industries is undergoing a data revolution, as the ability to extract knowledge from large volumes of data is becoming a reality. Between the 1980s and 2010, the total volume of historical data expanded from megabytes to terabytes. This sparked a big-data revolution, which resulted in the study of *Machine Learning (ML)* algorithms being developed in the field of computer science. On one extreme, where data are abundant in samples but relatively scarce in features, neural nets and Deep Learning by Hinton's group [1] allows predictive models to be constructed with unprecedented accuracy. On the other extreme, if data are scarce in samples but abundant in features, models such as Bayesian Networks [2] and Markov Random Fields [3] enable tasks such as inference and sampling. This results in a deeper understanding of the underlying probabilistic distributions behind the data, and enables “artificial" samples to be generated via methods such as Gibbs [4] and importance sampling. ML tools are so accessible today, such that anybody can train a deep neural net containing hundreds of layers and neurons within seconds using Tensorflow [5].

ML in computer science typically focuses on predictive modelling of datasets containing large numbers of samples, also known as *big-N* problems. Social-media data such as emails, images, or videos come in the billions, and therefore models like Deep Learning have an abundance of data to train and validate on. By contrast, the chemical and biological communities often see *small-N* problems, in which having even hundreds of samples is considered exorbitant. For example, direct concentration measurements of uncommon chemical or biological specimens species are costly and time-consuming. In most cases, these are obtained using *soft sensors* or *inferentials*, which suffer from large time-lags in-between measurements. Therefore, the abundance of *small-N*, *big-d* (high dimensionality) problems warrant a different modelling and analysis paradigm in the field of engineering.

When process engineering data is combined with biological data, the difficulty of meaningful analysis increases multiple-fold. The task of finding interpretable patterns and correlations in a combined chemical-biological dataset is an enormous challenge. This is partially due to the potential differences in time-scales, sampling rates, and dimensions in the two different types of data. Moreover, if the microbial data contains species-relative abundance data, the modelling task becomes extremely confounded. The species composition is stochastic, due to only some participating in the main reactions, and many bystander species which exert diminished, indirect effects. Many species also perform the same metabolic functions in a community, thus rendering them functionally redundant. Finally, member species of a community may interact with others in protagonistic or antagonistic manners. All of the aforementioned phenomena are present, but difficult to capture clearly in terms of their direct effects on bioreactor performance changes. Therefore, the analysis of datasets of such complexity require a strict workflow, to address as many anomalies as possible.

A general analysis framework can be suggested as follows. Given historical data, the process outcome is identified and preferably separated into *good* and *poor* groups. This is common in processes where the outcome pertains to a fractional or percentage value, corresponding to chemical yield or removal. The data are “compressed" using dimensionality-reduction techniques, which results in a smaller subset of representative features. Predictive models are built using these high-impact features, instead of the original feature-set (which may contain irrelevant or redundant data). Finally, the representative features are ranked in terms of importance (in contributing to the final process outcomes), using univariate feature selection techniques. The results from this approach serve as an informative pre-cursor to decision-making and control, especially in processes where little to no prior domain knowledge is available.

A visualization of the aforementioned framework is provided in Fig. 1 below, which can be realized as a typical closed-loop feedback control block diagram [6]:

In the proposed workflow, the choice of MVs is dynamic - it is re-identified given each influx of new data. On the other hand, traditional feedback control uses a *static* set of pre-specified MVs, which may not always be impactful variables if the process and/or noise dynamics vary with time. Specifically, the use of machine learning achieves a three-fold goal:

1. During each operating stage, operators would only need to monitor a small set of variables, instead of hundreds or thousands. This simplifies the controller tuning and maintenance drastically, and undesirable multivariable effects (such as input coupling) are reduced.
2. If the process model is time-varying and non-linear, first-principles models need to be re-identified at every operating stage. These models are also known as *white-box*, as they are purely mechanistic (ex. from mass, energy, or force balances) and based on physically-intuitive parameters. By using *black-box* or purely empirical (i.e. data-based) machine learning models instead, the process outcomes can be predicted ahead of time, such that unsatisfactory outcomes

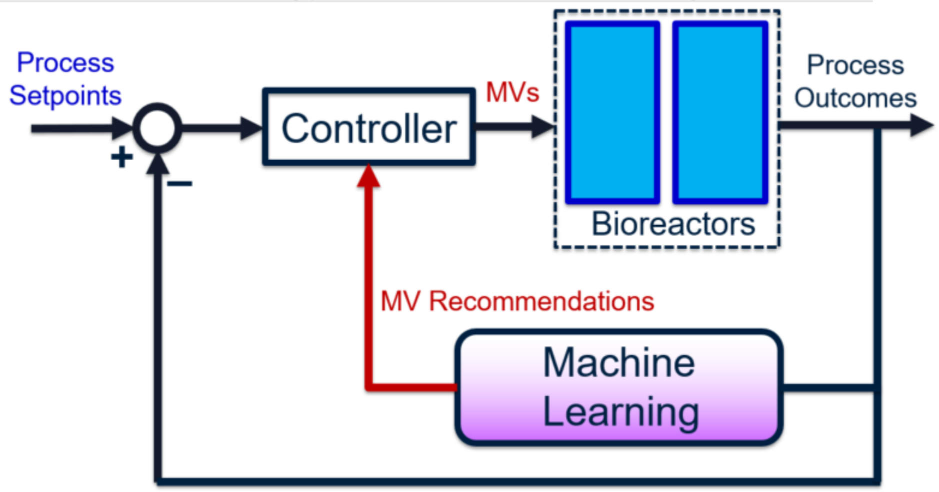


Figure 1. ML-guided process control and decision-making. Manipulated Variables (MVs) selected from the original process features may be high-dimensional and full of confounding effects. Instead, the small subset of MVs most responsible for causing observed process changes is identified using ML algorithms. The key MVs may change from one operating stage to another, but they can be re-identified given the corresponding new data.

are prevented. Moreover, the machine learning models can be updated using new data collected from each new operating stage, therefore eliminating the need of complete re-identification.

3. The ranking of feature impacts can be performed using *grey-box* models. These are data-based machine learning models *guided by* a modest amount of first-principles or domain knowledge about the system. This combination is exceptionally powerful if the domain knowledge is accurate, since it allows the model structure to be well-defined. Not only does this improve prediction accuracy dramatically, it also allows an analysis of the *relative importance* of each system variable (or feature) compared against one another. From a control engineer's perspective, monitoring and adjusting the entire set of system variables may be impractical, especially if the dimensionality is too high (i.e. hundreds or thousands of features). This is due to the well-known phenomenon of *the curse of dimensionality*, as well as other issues such as loop-coupling interactions [6]. On the other hand, if a small subset of that entire feature-space is identified as the key, high-impact variables to monitor, then the control problem becomes feasible and much more focused.

This paper will demonstrate the use of data analytics on a wastewater treatment process aimed at removing selenium. The first part of this paper outlines a systematic data pre-processing workflow, which combines both chemical and biological data on an equal scale. Then, a review of the state-of-art ML techniques in bioinformatics is provided. Three *unsupervised learning* techniques - *hierarchical clustering*, *Gaussian mixtures*, and *Dirichlet mixtures* - are explored as methods for dimensionality reduction. Three *supervised learning* techniques - *Random Forests (RFs)*, *Support Vector Machines (SVMs)*, and *Artificial Neural Networks (ANNs)* - are used to construct predictive models. Finally, important process features are correlated with selenium removal rate using two techniques - *Mean Decrease in Accuracy (MDA)*, and its *conditionally*-permutated variant, *C-MDA*. The quality of modelling and feature selection results are compared and contrasted across all explored methods.

One key difference between this work and others in the literature is the broad range of exploration, as well as extensive compare-and-contrast, of numerous methodologies for data analysis. Most papers focus on the proof-of-concept and results of a single technique, with focus on either the prediction task or feature analysis task. When reading this paper, The reader should focus more on the strengths and limitations of each method, given the results obtained, rather than the numerical values of the results themselves. The main goal of this work is to bring clarity to the appropriate use of analytics, given the various characteristics and circumstances of the available raw process data.

1.1 Nomenclature and commonly-used terms

The nomenclature in this paper will follow machine learning literature by [7] and [8]. Historical data can be divided into input data which contains time measurements of all process variables, and output data which contains desired process outcomes. Input data are compactly expressed using the matrix $\boldsymbol{X} \in \mathbb{R}^{N \times d_x}$, where N denotes the total number of samples and d_x the total number of variables. Examples of these process variables or *features* include temperature, pH, valve actuator positions, pump speeds, etc. Output data are denoted by $\boldsymbol{y} \in \mathbb{R}^N$, assuming only one outcome is considered in any model. Furthermore, it is assumed that all outcome variables are independent of one another. If multiple outcomes are to be analyzed at once, or if correlations exist between individual outcomes, then they can be concatenated into a matrix $\boldsymbol{Y}$. Examples of these outcomes include yields, final concentrations or flowrates, extents of reaction, removal rates, etc. When the input data are expressed as a matrix $\boldsymbol{X}$, its N samples are oriented as rows and its d_x features as columns, i.e.,

$$\boldsymbol{X} = \begin{bmatrix} —[\boldsymbol{x}^{(1)}]^\top— \\ \vdots \\ —[\boldsymbol{x}^{(i)}]^\top— \\ \vdots \\ —[\boldsymbol{x}^{(N)}]^\top— \end{bmatrix} = \begin{bmatrix} | & & | & & | \\ \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_j & \cdots & \boldsymbol{x}_{d_x} \\ | & & | & & | \end{bmatrix}. \tag{1}$$

The bracket-enclosed superscript $^{(i)}$ denotes samples, which differentiates it from the subscript $_j$ which denotes features. From a physically-intuitive perspective, these input features can be further differentiated into *macro* variables and *micro* variables. *Macro* variables mostly consist of sensor-measurable quantities, such as temperatures, flowrates, pressures, or pH. However, they can also include *inferential* or *soft-sensed* variables [6], which are not directly measurable but can be inferred from other easily-measurable variables. An example of this is the Chemical Oxygen Demand (COD), which is measured by extracting liquid samples from the system and performing analytical laboratory tests. On the other hand, *micro* variables are related to microbial properties, such as abundance counts or Spearman's/Pearson's correlations (which account for microbial interactions). In most cases, *micro* variables are *inferential*; a good example is Operational Taxonomic Unit (OTU) counts, which are obtained via 16S gene sequencing.

1.2 Model training, validation, and testing

Training, **validation**, and **testing** sets are defined with subtle differences in the scientific, engineering, and machine learning communities. This paper will adhere to the definitions accepted by the machine learning communities, which are as follows:

- **Training:** Samples used to obtain mathematical mappings (or *models*) between the input and output data.
- **Validation (or development):** Samples used to select optimal values of *hyperaparameters* - for example: model complexity (or order), regularization constants, etc. Systematic methods such as k-fold cross-validation are used.
- **Testing:** Samples restricted for assessing the performance (ex. accuracy) of the selected model. This reflects its capability of generalizing to new, unseen samples.

When building a model, the test set cannot influence the selection of model structure, parameters or hyperparameters in any way. This is known as the **"Golden Rule of Machine Learning"** [7]. Both [7] and [9] recommend a training/validation/testing split ratio between 50/25/25 and 90/5/5, for data containing up to a few thousand samples, the. For data with more than a million samples, split ratios between 90/10/10 and 98/1/1 are recommended. In these data-abundant cases, the goal is to use as many samples as possible for training, while maintaining a respectable number of samples available for validation and testing.

Training, validation and testing errors are usually evaluated in two different forms, depending on whether the models are of a classification or regressive nature:

$$\text{Error Fraction} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(\hat{y}^{(i)} \neq y^{(i)}) \text{ (Classification)}, \tag{2}$$

$$\text{Error Rate} = \frac{1}{n}\sum_{i=1}^{n} f(\hat{y}^{(i)} - y^{(i)}) \text{ (Regression)}, \tag{3}$$

The symbol $\mathbb{1}$ represents the indicator function and $\hat{y}^{(i)}$ the estimated output for the i^{th} sample using the selected model. In the case of classification, the error is calculated as a fraction of mismatched samples. In the case of regression, the error is computed as an average sum of errors in with respect to the selected error function f (ex. mean squared error).

Finally, the *bias-variance tradeoff* is another important consideration when building a predictive model. The user must compromise between a simple model which *"under-fits"* (high bias, small variance) and a complex model that *"over-fits"* (small bias, high variance). The optimal point of balance can be determined by techniques such as *k-fold cross validation* or *information criteria measures*.

1.3 Importance of Data Pre-Treatment

Bioreactor data, like data in any other application, is masked by noise which can originate from any of the following factors:

- Uncalibrated, aging, or malfunctioning sensors
- Unexpected plant disruptions or shutdowns
- Human errors in data recording (either incorrect or missing values)
- Unmeasured, drifting disturbances (such as seasonal ambient temperatures)

Data must be cleaned prior to any modelling task, as the model quality is directly influenced by the data quality. The following approaches are well-known and straightforward to employ, but are of paramount importance in terms of obtaining high-quality predictive models:

1. **Outlier removal based on human intuitions:** the elimination of spurious sensor values (ex. negative flowrates recorded through a valve) using *a priori* knowledge. These values can either be replaced by *NaN* (missing) values, or estimates via imputation.
2. **Standardization:** the scaling of each feature to zero-mean and unit variance, equalizing the effect of each individual feature. This prevents features with relatively large ranges (ex. flowrate with range ±1000) from dominating model weights over features with relatively small ranges (ex. pH with range ±0.1).
3. **Imputation:** the estimation of missing values, using *a priori* knowledge if available, or using standard techniques such as interpolation - for example, *Zero-Order-Hold (ZOH)* or linear interpolation.
4. **Smoothing:** the flattening of spiky measurements due to sensor noise, using techniques such as Moving-Average (MA) filters.
5. **Common time-grid alignment:** the unification of sampling intervals for time-series data. For example, consider a variable measured every second, and another measured every 0.5 seconds. In order to model using both variables, each variable must contain the same number of samples. Therefore the uniform time-grid can either be taken at every second (losing half the resolution of the second variable) or every 0.5 seconds (requiring interpolation of the first variable).

Although the aforementioned techniques are usually sufficient in removing simple abberations, additional methods must also be considered in the case of more complex data. For example, the interactive and non-linear nature bioreactor systems may require the following approaches:

1. **Log-transforms of population counts:** if population distributions are severely skewed towards low or high counts, then it is more practical to express them as powers of suitable bases, such as 10 or e.
2. **Removal of low-population species:** this is akin to outlier removal based on *a priori* knowledge. Species with low counts can be removed by defining an absolute cut-off (ex. any value below 1000), or by comparative magnitudes (ex. less than 5% of the next smallest value).
3. **Correlations between species:** In a microbiological community, individual members rarely act independently. The effect of each individual upon all others can be quantified in terms of co-existence, using linear correlation (ex. Pearson's) or non-linear (ex. Spearman's). The co-behaviour of microorganisms can produce valuable insights into observed process outcomes.

1.4 Predictive Models

The task of prediction is an important one, especially in bio-remediation processes. Examples of important process outcomes or outputs $\boldsymbol{y}$ include effluent pollutant concentrations, pollutant reduction rates, etc. These variables are generally continuous, i.e. the goal is to predict their exact values, rather than discrete categories. Prediction is the first step towards adequate control - if these values cannot be predicted, then appropriate control actions based on these values cannot be introduced. Prediction is usually accomplished using **supervised learning**: a model is constructed between known, historical training inputs $\boldsymbol{X}^{(\text{train})}$ and matching outputs $\boldsymbol{y}^{(\text{train})}$. The model's *hyperparameters*, which are parameters that determine model complexity (ex. model order or structure, the type and magnitude of regularization used, etc.) are determined using a validation set. Finally, the new outputs $\hat{\boldsymbol{y}}$ are predicted for new inputs $\hat{\boldsymbol{X}}$ using the constructed model. In many applications within the machine learning field, data are usually assumed to be *independently-and-identically-distributed (IID)*. In other words, the individual samples are not correlated with each other in time, and the probability of observing each sample can be modelled by the same, stationary distribution. Therefore, standard machine learning algorithms such as least-squares, Support Vector Machines (SVMs), etc. can be directly applied on the raw data. On the other hand, in processes involving chemical and biological interactions, data are often correlated both with respect to features and time. In these cases, the IID assumption does not apply. Instead, the two following approaches are commonly employed:

- Obtain time-averaged values of each feature for each experiment or run, and treat all averages as IID. This works for experiments which are fairly isolated and collect few samples per run (ex. 5 or less samples), but fails for experiments which are sampled at a high resolution (ex. 10s of samples).
- Collect time-samples for each value of each feature, and employ time-series modelling techniques. These approaches either account for temporal correlations directly, or use latent (hidden) variables to indirectly characterize temporal dependencies.

For general process control, a time-series modelling framework has been established by [10]. These methods apply to data which is assumed to be *Linear-Time-Invariant (LTI)*: the model explaining the data obeys the principle of superposition, and is stationary. Although this assumption does not hold perfectly for most chemical and biological systems, it holds approximately for systems excited by small perturbations, or small sections of data corresponding to locally-linear periods of operation. Prominent LTI time-series models include *Finite-Impulse-Response (FIR)*, *Autoregressive-with-Exogenous-Inputs (ARX)*, and *Autoregressive-with-Moving-Average (ARMA)*. Successful applications of these techniques are demonstrated in [11] and [12]. In these papers, dominant members of the microbial communities were first identified using networks, then an *Autoregressive-with-Integrated-Moving-Average (ARIMA)* model was constructed to characterize their temporal behaviour. Their co-existences were then associated with tangible features, such as diet or presence of inflammations. Another application involving time-series modelling includes [13], which first uses Bayesian networks to capture a probabilistic model describing interactions between members of communities within microbial fuel cells. Then, Artificial Neural Networks (ANNs) are used to predict important process outcomes such as Coulombic efficiency, power generation, and removal rates. Besides outcome prediction, however, ANNs can also be used to estimate interim parameters. For example, [14], [15], and [16] used ANNs to estimate the optimal controller parameters in terms of controller performance and stability, for wastewater treatment applications which are similar to the one outlined in this paper.

1.5 Clustering and Dimensionality Reduction

Clustering can be formulated as both **supervised** or **unsupervised learning** tasks. In the case of **supervised learning**, clustering delves into the existing data to identify groups or classes of data samples, or features, that belong together. One example is the segregation of process variables (such as temperature, yield, etc.) into discrete bins, with class labels such as "good," "intermediate," or "poor." The discretization is usually based on cut-off values, which can be determined *ad hoc*, by intuition, or by statistical measures such as percentiles. Models are constructed between inputs and outputs belonging to a training set, with its hyperparameters determined using a validation set, then used to predict unknown outputs corresponding to new input samples on a test set.

On the other hand, clustering can also be performed as using **unsupervised learning** approaches. An example is the *k*-means algorithm, which assigns data to a user-defined *k* clusters based on Euclidean distances between individual samples. In this case, no new samples are involved, as prediction is not the main goal; rather, the goal is to locate patterns within the data samples at hand. Another example is the grouping of dogs into Carnivora (order), Canidae (family), and Canis (genus) using hierarchical clustering. These represent three ranks of clusters which are formed using a user-defined similarity metric (ex. Euclidean distance) between sample features.

Finally, clustering can also be applied to data features instead of samples. Examples include Pearson's and Spearman's correlations within features, and association networks between bacterial species given their abundance counts. [17] provides an extensive review of clustering within the application of gene sequencing and phylogentic marking. Two main methods in the paper include clustering based on similarity measures (such as Bray-Curtis) or probabilistic distributions (such as mixture models). The *Dirichlet Multinomial Model (DMM)* approach is explored in greater detail in [18] and [19]. This approach is preferred over hierarchical clustering or *k*-means for sparse data, where the distribution values of the individual features (ex. taxa abundance counts) are skewed towards either low or high numbers.

Regardless of which clustering method is employed, the common goal is to either identify groups of samples which are "alike," or groups of features which are strongly associated with a certain process outcome. In the current project, clustering will be used specifically to identify dominant OTUs which are associated with satisfactory and poor reduction rates of pollutants.

1.6 Network Analysis

Individual OTUs of a microbial community act cocurrently rather than independently. Their direct affects on the water chemistry variables are difficult to isolate, due to the stochastic nature of such a community. Specifically, members may be protagonistic and antagonistic to one another, or they may be neutral altogether as bystanders. Therefore, the exact effect of individual or groups of OTUs on the final process outcome is extremely confounded. However, several recent papers have attempted to bring clarity to the microbial effects. The approach is known as *network analysis*, and the overall strategy is known as **"networks to models"**[11]. The more recent work of [12] showed that, not only can dominant microbial groups be directly linked to a certain outcome, but indirect players (i.e. those facilitating the interaction between two dominant groups) can also be identified. Such a modelling strategy can provide the engineer with second layer of knowledge, on top of existing process variables, the importance of each microbial species on the process outcomes.

Network analysis attempts to find associations, given the abundance or population counts of microbial species. This is a step-up from using the abundance counts themselves, since the counts alone may not be entirely descriptive of the underlying chemical-biological effects. On the other hand, associations (which are derived from abundance counts) indicate the nature of relationships between individual species. Network models can be derived, for example, using Lotka-Volterra [20], [21] (i.e. predatory vs co-existing relationships) or a more modern approach known as *netassoc*. The *netassoc* model is statistical verification of co-cocurrence using an algorithm developed by [22]. It estimates the true partial correlation between two pairwise OTU species by isolating and mitigating the indirect effect of possible third (or more) species. Advantages of this approach include the ability to visualize the total number of positive and negative links each OTU has with all other OTUs, thus determining whether it is an *aggregator* or *predator*. From a process control perspective, the *aggregator* OTUs associated with a positive process outcome (i.e. high removal rate) should be maintained, while the corresponding *predator* OTUs should be inhibited as much as possible.

2 Background and Methods

Prediction can be performed using two main approaches, **regression** (continuous outcomes) and **classification** (discrete, categorical outcomes). Both approaches produce estimates of new outcomes given a new set of process inputs. In all prediction models, the goal is to minimize some form of error or loss function between predictions $\hat{y}$ and real outputs y within the training set. The final goal is to predict new outcomes matching a given set of new inputs. The three predictions used in this project are outlined in the following subsections.

2.1 Supervised Learning Methods

2.1.1 Random Forests (RFs)

The first predictive model used in this work is *Random Forests* [23]. These are a class of models which determines the final categorical outcome based on conditional binary splits of each feature. Since it is computationally impractical to produce binary splits on an extremely large feature space, random subsets of features are split on instead. The final outcome label is selected by taking a majority vote. Refer to Section B for more details about this model.

2.1.2 Support Vector Machines (SVMs)

The second predictive model used is *Support Vector Machines* [24]. An SVM attempts to find the *separating boundaries* between classes in the feature-space of the provided training data. Refer to Section C for the details behind this model.

2.1.3 Artificial Neural Networks (ANNs) and Deep Learning (DL)

The final method used for prediction in this paper is *Artificial Neural Networks*. The more popularly-known term *Deep Learning* refers to ANNs that have more than 10-20 hidden layers [9]. If the data sample-size N is abundant, a well-tuned ANN model can vastly outperform simpler ones (such as RFs or SVMs) in terms of prediction accuracy. This is due to ANN activation functions (such as *ReLU* or *sigmoid*) being *universal approximators* of any continuous function, linear or non-linear [25]. Modern ANNs are usually constructed using the well-known *Python* package *Tensorflow* [5]. Refer to Section D for the details behind this this model.

An interesting, recent advancement in this field is the work of [26]. The authors developed the *Stochastic Configuration Network*, which is an improved, adaptive version of ANNs. On each training iteration, it learns not only the optimal parameters (i.e. weights and biases) that minimize prediction error, but also the optimal architecture (i.e. number of layers, number of neurons in each layer).

2.2 Unsupervised Learning Methods

In many cases, the goal of data analysis is to not only make accurate predictions, but to also look within the existing data to identify *latent* features responsible for the observed outcomes. As an example relevant to the case study at hand, biological systems often deal with the analysis of *Operational Taxonomic Units (OTUs)*. These represent groups of micro-organisms which have been clustered genetically using 16S *r*RNA sequencing [27]. When the macro (i.e. water-chemistry) variables are brought into the analysis, the resulting chemical-microbiological interactions cannot be ignored. However, these profound coupling effects are difficult to identify as closed-form expressions, due to the stochastic nature of OTU communities. Raw biological data such as OTU abundances cannot be used in its original form, for the following main reasons:

1. **High dimensionality:** Thousands of OTUs may be present in bioreactors, and hence it is not feasible to include all of them as separate variables.
2. **Dominant groups:** Similar OTUs like to co-exist, while dissimilar OTUs like to “repel" or perhaps even destroy one another. These interactions are difficult to characterize by examining abundance counts alone.
3. **Coupled interactions between micro and macro features:** Chemical and biological variables seldomly act in isolation. Their confounding effects should also be characterized in some manner.
4. **Process insight/knowledge:** Knowing which group(s) of OTUs are dominant and responsible for good or poor outputs is invaluable, especially for subsequent process monitoring and control.

Therefore, several carefully-selected clustering and dimensionality reduction tools are required to extract meaningful information from a chemical-biological system. These methods are outlined in the following subsections.

2.2.1 Hierarchical Clustering

Hierarchical clustering performs grouping on microbial species, based on similarity measures between pairwise species. The underlying assumption is that all species are similar to others, and that the extent of similarity can be characterized using a *ranking* system. The similarities are quantified using some popular metrics in the following Table 1:

Table 1. Typical similarity formulas used

Type	$S(x^{(i)}, x^{(j)})$
Euclidean Distance	$\Vert x^{(i)} - x^{(j)} \Vert_2$
Manhattan Distance	$\Vert x^{(i)} - x^{(j)} \Vert_1$
Cosine Similarity	$\frac{x^{(i)\top} x^{(j)}}{\Vert x^{(i)}\Vert_2 \cdot \Vert x^{(j)}\Vert_2}$
Jaccard Similarity	$\frac{\mathbf{1}(x^{(i)}=c \cap x^{(j)}=c)}{\mathbf{1}(x^{(i)}=c \cup x^{(j)}=c)}, c \in [1, \cdots, C]$
Bray-Curtis Similarity	$\frac{\sum \lvert x^{(i)} - x^{(j)} \rvert}{\sum \lvert x^{(i)} + x^{(j)} \rvert}$

The result of a hierarchical clustering can be expressed using a tree-like structure known as a *dendrogram*, which shows the overall *hierarchy* or ranking of clusters. Obviously, different similarity metrics result in different-looking dendrograms. Moreover, each dendrogram has various “depths" which represent sample clusters of various sizes. The corresponding labels for new samples can be quickly identified by determining which clusters these samples are closest to, based on the desired similarity metric. Finally, dendrograms can be drawn using the following two different methods:

- **Agglomerative (bottom-up):** Start with individual samples, then gradually merge them into clusters until one big cluster remains. *This is the most common method.*
- **Divisive (top-down):** Samples start as one big cluster, then gradually diverge into an increasing number of clusters, until one cluster is formed for each individual sample.

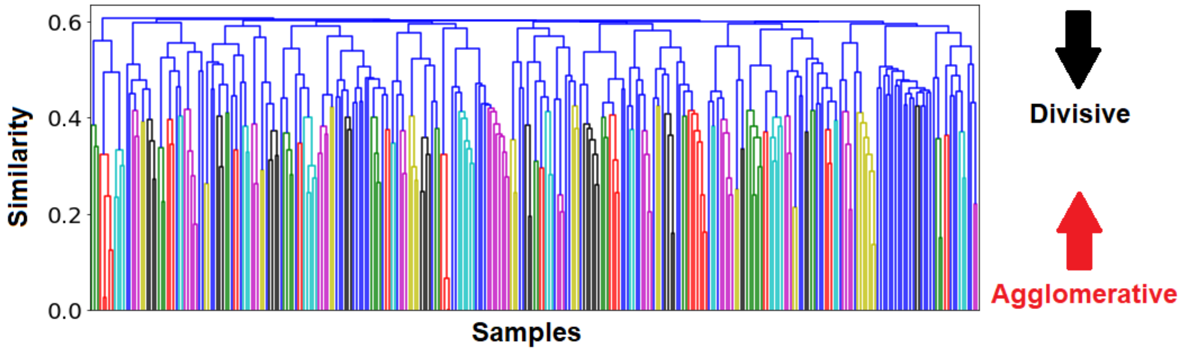


Figure 2. A dendrogram representation of hierarchical clustering. At the bottom, each individual sample belongs to its own cluster. Going up the dendrogram, samples are merged together based on the desired distance metric. At the top, all samples are merged into one giant cluster.

Four main types of hierarchical clustering are commonly used [28]. These are accompanied by two metrics, which determine the optimal clustering method among the four (i.e. *Cophenetic correlations* [29]) as well as the optimal number of clusters (i.e. *Silhouette analysis* [30]). Details of these methods can be found in Section E.

2.2.2 Probabilistic Mixture Models

The motivation behind using *probabilistic mixtures* is to model the underlying distributions of the given data. Models using one distribution are sufficient for uni-modal systems, but fails to capture multi-modal systems effectively. Therefore, data are usually modelled as the *sums* of various probabilistic distributions, with the structure of said distributions specified as a prior assumption. Mixture models are different from the hierarchical models mentioned in 2.2.1. The difference lies in the assumption that in mixtures, each individual species is assigned a group to which it is similar, but overlaps may occur between multiple groups. In other words, each species may belong to more than one group. This introduces a degree of stochasticity which makes these models more flexible. The two mixtures used in this paper are:

1. **Gaussian Mixtures [8]:** $p(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$; underlying distribution is assumed to be a sum of K weighted multivariate Gaussians with individual means and covariances. The term w_k represents the weighting factor for each Gaussian. Each Gaussian has the formula $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^{d_x} \cdot \det(\boldsymbol{\Sigma}_k)}} \cdot exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \cdot \boldsymbol{\Sigma}_k^{-1} \cdot (\boldsymbol{x} - \boldsymbol{\mu}_k) \right]$.
2. **Dirichlet Mixtures [19]:** Define $\boldsymbol{p}^{(i)}$ as a vector containing the probabilities that sample $\boldsymbol{x}^{(i)}$ belongs to each community species. The Dirichlet mixture *prior* over K distributions is $\boldsymbol{P}(\boldsymbol{p}^{(i)}) = \sum_{k=1}^{K} Dir(\boldsymbol{p}^{(i)} \mid \alpha_k)\pi_k$, where α_k are the *Dirichlet parameters* and π_k are the *Dirichlet weights*.

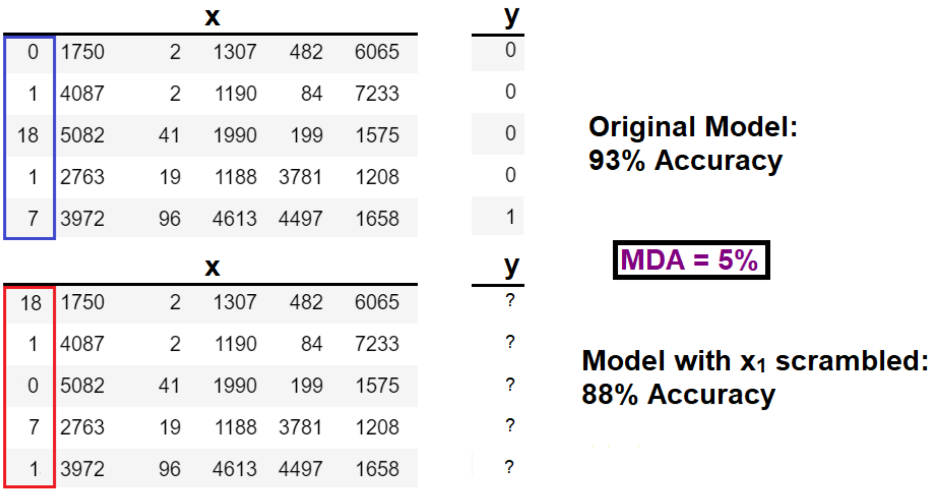
The Gaussian assumption is reasonable for most natural processes, which assumes that the underlying distributions are symmetric. When little a priori knowledge is available, it is a popular choice. However, if domain knowledge is available, it should be used to guide the choice of distribution used. For example, if OTU data mostly contains abundances skewed towards low counts, then the Dirichlet mixture will model the data more accurately than Gaussian. Details behind the Dirichlet distribution can be found in Section G.

2.3 Feature Selection

If an outcome is predicted using a set of features, a natural question arises: "Which of these features contribute the most to the observed predictions?" Although most feature selection approaches in literature are often customized on a case-by-case basis, two overarching groups of methods can be identified:

1. Hypothesis testing: A model is trained with all features left untouched. Then, features are either removed or permutated (scrambled), either individually or conditionally according to other features. The model is re-trained, and its accuracy is compared to the base-case accuracy. The features which cause the largest decreases in model accuracy are considered “most important," and vice versa.
2. Scoring: A metric or “score" based on information or cross-entropy is defined and calculated for all features. Features with the highest scores are identified as “most relevant," and vice versa.

In the hypothesis testing framework, univariate (or single-feature) algorithms such as *Mean Decrease in Accuracy (MDA)*, *Mean Gini Impurity (MGI)* [31] have been developed for simple models such as random forests. The MDA method can be visualized in Fig. 3.



x						y
0	1750	2	1307	482	6065	0
1	4087	2	1190	84	7233	0
18	5082	41	1990	199	1575	0
1	2763	19	1188	3781	1208	0
7	3972	96	4613	4497	1658	1

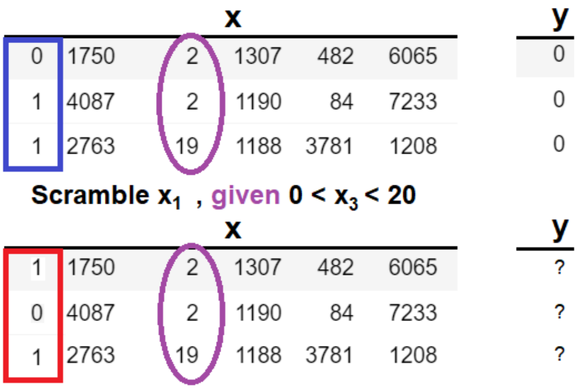
x						y
18	1750	2	1307	482	6065	?
1	4087	2	1190	84	7233	?
0	5082	41	1990	199	1575	?
7	2763	19	1188	3781	1208	?
1	3972	96	4613	4497	1658	?

Figure 3. MDA applied on a dataset with 6 features. During each outer iteration, the values of a single feature are scrambled or *permutated* sample-wise. The model accuracy with the scrambled feature is compared against the base-case model accuracy. If the accuracy decreases significantly, then the feature is considered "important." On the other hand, if the accuracy decreases negligibly, then the feature is "irrelevant" to the model.

Unfortunately, these univariate approaches have the following shortcomings:

- Inability to recognize coupling effects between multiple features, such as correlations or redundancies [32].
- Inability to distinguish conditional effects between features, i.e. whether a feature is “relevant" given the presence of other feature(s).

The second point above confounds the definition of “relevance." A classic example is the prediction of presence of genetic disease (the outcome) using the genetic information of a person’s mother and grandmother. If information from the mother is absent, then the grandmother’s genes may be identified as a “relevant" feature. However, if genetic information is present from both the mother and grandmother, then the grandmother’s genes may become “redundant" and thus an “irrelevant" feature. Therefore, the “relevance" of a feature contingent or *conditional* on the presence of other features. [33] has made significant contribution in the modelling of conditional dependencies. Its proposed *Conditional Mean Decrease in Accuracy (C-MDA)* approach is a variation on classic MDA, where *conditional permutations* are performed *given* the presence of other features. The conditional is defined as the appearance of secondary features within specified ranges of values. The difference in permutation between MDA and C-MDA can be realized in the following Fig. 4.



x						y
0	1750	2	1307	482	6065	0
1	4087	2	1190	84	7233	0
1	2763	19	1188	3781	1208	0

Scramble x_1 , given $0 < x_3 < 20$

x						y
1	1750	2	1307	482	6065	?
0	4087	2	1190	84	7233	?
1	2763	19	1188	3781	1208	?

Figure 4. In CMDA, the permutation is only performed on the values of a feature *given* the presence of another feature falling within a range of values. By contrast, permutation in traditional MDA (as shown in Fig. 3) is performed on all values of a feature, with no consideration of other features.

3 Process Description

The case study pertains to a wastewater treatment process located downstream of a mining operation. Due to proprietary reasons, the description is kept at a general level.

3.1 Process Flow Diagram and Description

The process can be visualized as the general bioreactor shown in Fig. 5:

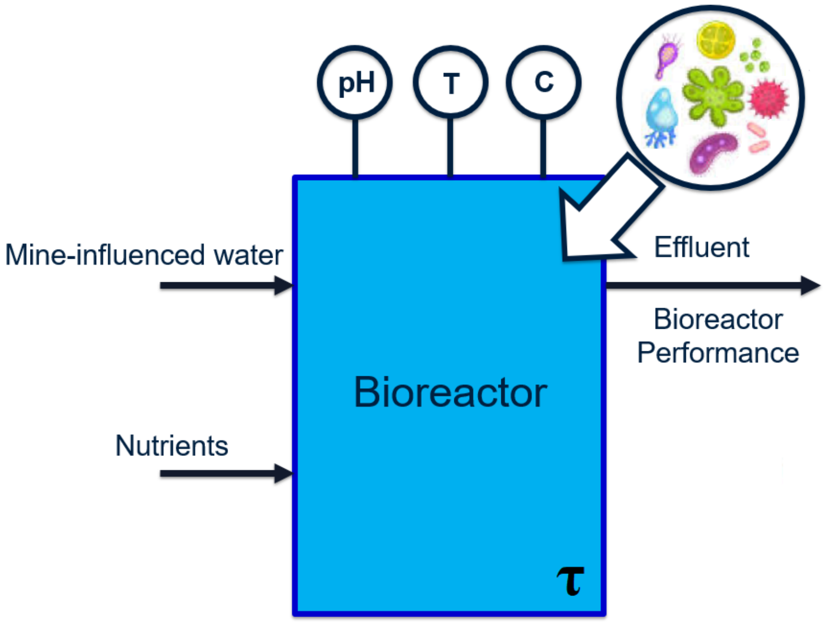


Figure 5. Simple bioreactor schematic, with wastewater and biological nutrients as inlets, and treated effluent as outlet. The system contains directly-measurable *macro* variables related to water chemistry (such as contact time τ), and difficult-to-measure *micro* variables reflecting the metabolism of micro-organisms.

Selenate and *nitrate* concentrations in the bioreactor effluent must be reduced to below $10\frac{\mu g}{L}$ and $3\frac{mg}{L}$, respectively [34], [35]. These chemical species bio-accumulate in the marine ecosystem [36] and thus reach harmful levels at the top of the food chain.

3.2 Water Chemistry Details

The feed to the first reactor is wastewater, which contains the main pollutant *selenate* (SeO_4^{2-}). The selenate is to be reduced to elemental *selenium* (Se) by a series of two bioreactors. Samples are extracted from the bioreactors during each operating stage (at irregular intervals) and analyzed, in order to determine and record values of various water chemistry variables. These features are summarized in the following Table 2.

Table 2. Water Chemistry Variables

Variable	Description
τ or $EBCT$	Empty-Bed-Contact-Time $= \frac{\text{volume}}{\text{flowrate}}$ (min)
$Ammonia_{out}$	Concentration of NH_3 in effluent $(\frac{mg}{L})$
$Nitrate_{in}$	Concentration of NO_3^- in influent $(\frac{mg}{L})$
$Nitrite_{out}$	Concentration of NO_2^- in effluent $(\frac{mg}{L})$
SeD_{in}	Concentration of total dissolved Se in influent $(\frac{\mu g}{L})$
COD_{in}	Chemical oxygen demand in the influent $(\frac{mg}{L})$
MicroC	Equal to 1 if *MicroC* is added as carbon source, otherwise 0
Acetate	Equal to 1 if *Acetate* is added as carbon source, otherwise 0
Reactor 1	Equal to 1 if *Reactor 1* is the relevant bioreactor, otherwise 0
Reactor 2	Equal to 1 if *Reactor 2* is the relevant bioreactor, otherwise 0

3.3 Microbiology Details

In addition to the water chemistry data, data pertaining to the microbial presence is available in the form of *Operational Taxonomic Units (OTUs)*. An OTU is a cluster of 16S *r*RNA gene biomarkers that are more than 97% similar to one another. Therefore, each OTU is considered to represent one bacterial species [27]. In this case study, the numerical values associated with each OTU are known as *raw abundance counts*. These counts can be considered normalized population counts of each bacterial species, which fall within the range of $0 \sim 16000$.

4 Data Pretreatment

Before the water chemistry and micro-biological data can be used for modelling or feature analysis, they must be pre-processed. The steps involved can be visualized as a workflow in the following Fig. 6.

The data pre-processing was performed using Jupyter *iPython* notebooks. The raw dataset originally consists of two files: one containing water chemistry data, and one containing OTU counts. First, samples containing missing or NaN values were removed using the *dropna* function in *pandas*. Then, spurious process values (such as negative flowrates) were removed by Boolean functions. The remaining samples were then cross-matched between the water chemistry and OTU files, by use of $SampleID$ tags which identify common operating stages. This results in a total of $N = 56$ samples containing both water chemistry and microbial information. Although this is a small sample-size, it is unfortunately all the data that could be collected from this treatment plant.

4.1 Pretreatment of Water Chemistry Data

Each water chemistry variable outlined in Table 2 (except *SampleID*) are *standardized* via the following two steps:

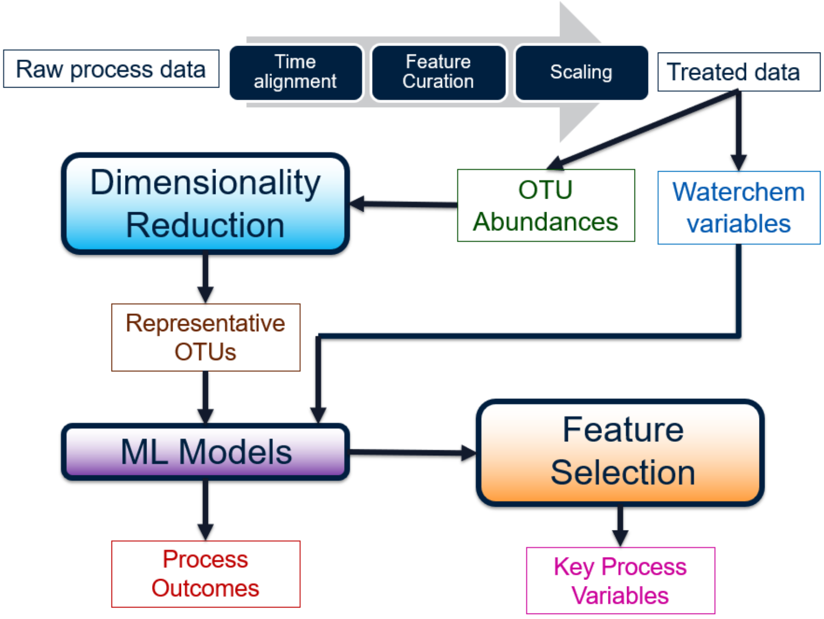


Figure 6. Workflow of the pre-processing, dimensionality reduction, modelling, and feature selection steps. The final goal is to transform the input data into predicted outcomes, as well as key variables responsible for said outcomes.

1. **Mean-centering:** For each feature j, subtract its sample values by its mean value, i.e. $\boldsymbol{x}_j^{(i)} \longleftarrow \boldsymbol{x}_j^{(i)} - \mu_j$
2. **Unifying-variance:** Divide the values from the previous step by the corresponding feature standard deviations, i.e. $\boldsymbol{x}_j^{(i)} \longleftarrow \frac{\boldsymbol{x}_j^{(i)}}{\sigma_j}$

In the *standardized* data matrix, each feature (or column) has a mean of $\mu_j = 0$ and variance $\sigma_j^2 = 1$, which removes any weight-skewing effects during model construction due to varying feature ranges.

4.2 Pretreatment of Microbiological Data

The OTU raw abundance counts are recorded in a matrix where the number of samples and number of OTUs are $N = 56$ and $d_{OTU} = 305$, respectively. The raw counts fall within the range of $0 \sim 16000$. As observed in Fig. 7 below, the abundance distribution is heavily skewed towards the lower numbers, which means that any model built using these raw counts would be heavily biased towards the lightly-populated OTU species:

The skew is partially remedied by applying a log_{10}-transformation to all raw counts. Since many raw counts are equal to zero, 1 is added to every value before the log_{10} transformation, to ensure the log_{10} operation is valid. Counts equal to zero would still remain zero after transformation, since $log_{10}(0+1) = 0$. The overall operation is:

$$\text{Count}_{\text{scaled}} = log_{10}(\text{Count}_{\text{raw}} + 1) \tag{4}$$

The resulting distribution of the scaled counts can be observed in Fig. 8: it is still skewed towards the low end, but not as severely as the raw counts.

These counts are now in a suitable form for data analysis outlined in the following Section 5.

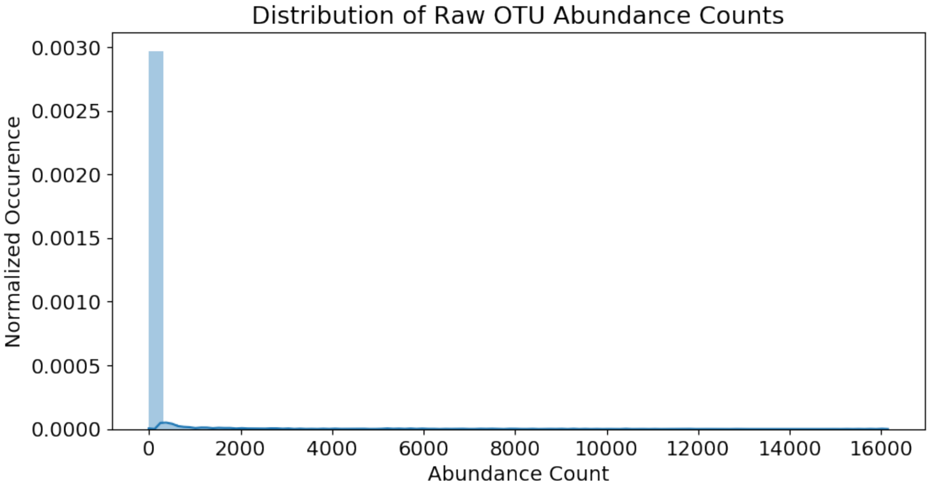


Figure 7. Distribution of all available raw OTU abundance counts.

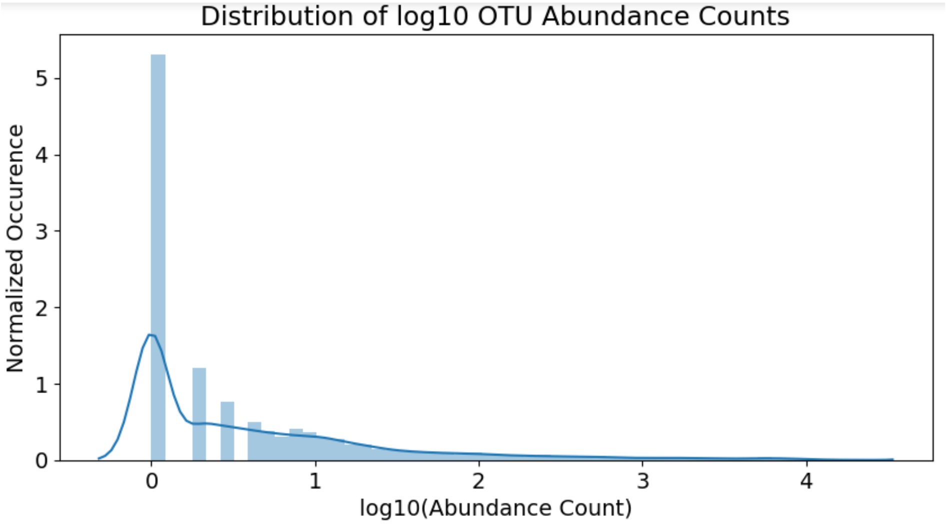


Figure 8. Distribution of OTU abundance counts, after log_{10} transformation.

5 Data Analysis

This section compares and contrasts the pertinent results from various *supervised*, *unsupervised*, and *feature selection* methods. Please visit the main author's *GitHub* repository to access the data and code.

5.1 Hierarchical Clustering of OTUs

The log_{10}-transformed counts obtained from pre-processing are first analyzed in terms of biological associations. This provides preliminary knowledge into the possible *co-existing* and/or *antagonistic* interactions between OTUs. In order to prevent spurious correlations (which are possible using methods such as Pearson or Spearman correlations), the *netassoc* algorithm by [22] is used. The result is a 305-by-305 matrix acting as a "pseudo" distance matrix between all OTUs, which can then be used for hierarchical clustering.

Before the *netassoc* distances can be used, however, it must undergo one final transformation: normalization of values between 0 and 1. This follows the concept of *similarity* being analogous to small distances (i.e. distances close to zero), and *dissimilarity* being analogous to large distances. The operation in Eq. 5 accomplishes this scaling:

$$\mathrm{dist}_{\mathrm{scaled}} = \frac{\mathrm{dist} - \mathrm{dist}_{\mathrm{min}}}{\mathrm{dist}_{\mathrm{max}} - \mathrm{dist}_{\mathrm{min}}} \tag{5}$$

At this point, the hierarchical clustering models can finally be constructed. First, the following four hierarchical clustering methods are performed on the scaled *netassoc* distance matrix:

1. **U**nweighted **P**air-**G**roup **M**ethod with **A**rithmetic Means (UPGMA)
2. Ward's Minimum Variance Method (Ward)
3. Nearest-Neighbour Method (Single-Linkage)
4. Farthest-Neighbour Method (Complete-Linkage)

This was accomplished using the *scipy* package *cluster.hierarchy*. In order to determine the "optimal" clustering method out of the four, the Cophenetic correlation values (see Section E) were obtained using the *cluster.cophenet* command, for all four methods. The results are shown in the following Table 3:

Table 3. Cophenetic correlations

Method	Coph. correlation
UPGMA	0.51
Ward	0.41
Single-linkage	0.08
Complete-linkage	0.22

Cophenetic correlations can be thought of as *how well a clustering method preserves the similarites between raw samples*. Since the UPGMA method has the highest Cophenetic correlation, it was selected as the most suitable clustering method. A dendrogram was then constructed using this method, and it can be visualized in the following Fig. 9:

The optimal number of clusters on this UPGMA dendrogram is determined by Silhouette analysis (see Section E), which is a measure of *how well cluster members belong to their respective clusters, given the number of desired clusters K*. Silhouette values are computed for cluster numbers $K = 2$ through $K = 100$, and the results are plotted on the following Fig. 10:

From Silhouette analysis, $K = 45$ groups appears to be the "optimal" cut-off with the overall highest Silhouette value. However, this is assuming that all *netassoc* distances are suitable for use. Recall that a normalized distance of 0 resembles similarity, and a distance of 1 resembles dissimilarity. A distance of 0.5 corresponds to neither similarity or dissimilarity. Values in that vicinity represent "neutral" OTU interactions which act as noise, confounding the clustering model. To remedy this issue, a *distance cut-off* approach inspired by [37] was employed. If a hierarchy with a *distance cut-off* value of $dist_{cut}$ is constructed, it means that no cluster contains members which are spread apart by a distance greater than $dist_{cut}$. This reduces the amount of overlap between distinct clusters. To determine the precise value of $dist_{cut}$, several UPGMA hierarchies were constructed using distance cutoffs within the set of values $dist_{\mathrm{cut}} \in [0.4, 0.6]$. The resulting Silhouette values are reported in the following Fig. 11:

The optimal distance cut-off is located at 0.54 with a corresponding maximum Silhouette value of 0.068. By constructing a UPGMA hierarchy with this cut-off, no two members within any cluster are spread apart by a normalized distance of 0.54. This UPGMA hierarchy yields a total of $K = 37$ clusters, and its dendrogram is provided in the following Fig. 12:

In each cluster, the "dominant" OTU was determined as the one closest (in terms of normalized *netassoc* distance) to the cluster centroid. The coordinates of each centroid were readily calculated

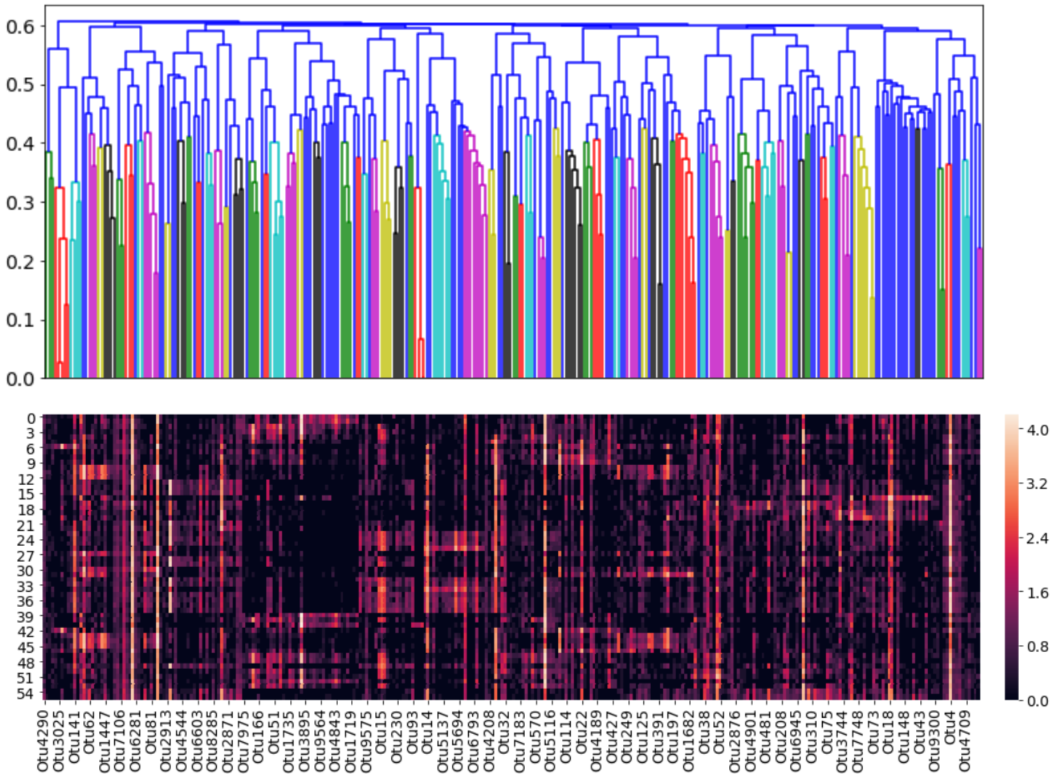


Figure 9. UPGMA dendrogram (top) and heatmap (bottom) showing log-transformed OTU abundances. The rows of the heatmap represent individual samples, while the columns represent individual OTUs. *Dark* colours on the heatmap represent distances close to zero and hence *similar* OTUs, while *light* colours represent large distances and hence *dissimilar* OTUs.

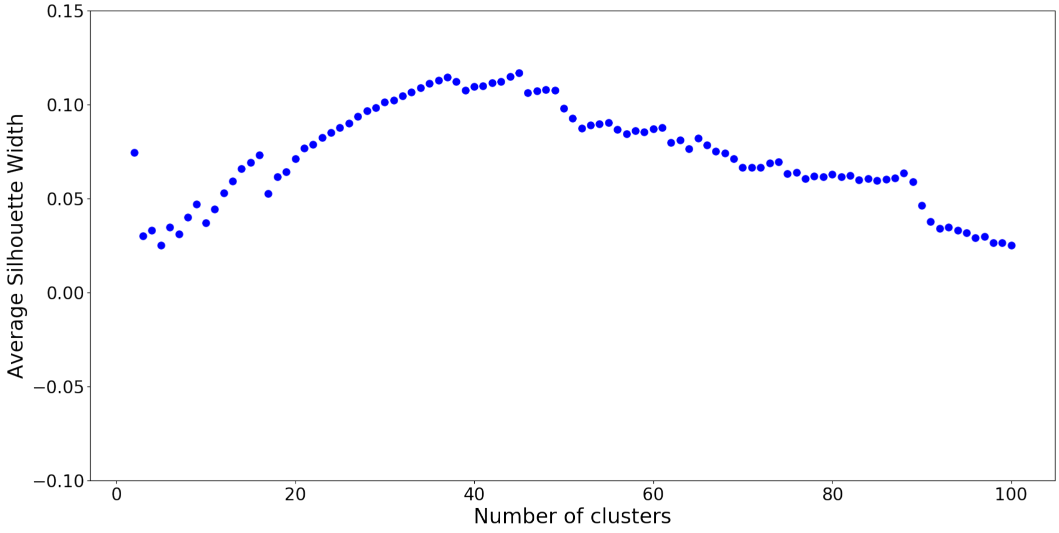


Figure 10. Silhouette numbers for clusters $2 < K < 100$. The highest value of 0.117 occurs at $K = 45$.

using the distances in the dendrogram. The remaining OTUs in the cluster were therefore considered “followers." The entire cluster could then be considered a co-existing community of OTUs. In the following Fig. 13, the number of members in each clusters (which is also shown in Fig. 12) is plotted against the cluster number:

On one hand, clusters 13 and 34 are the largest communities, with 19 OTUs in each. On the other hand, cluster 1 and 8 are the smallest communities, with 3 OTUs in each, followed by groups 16, 21,

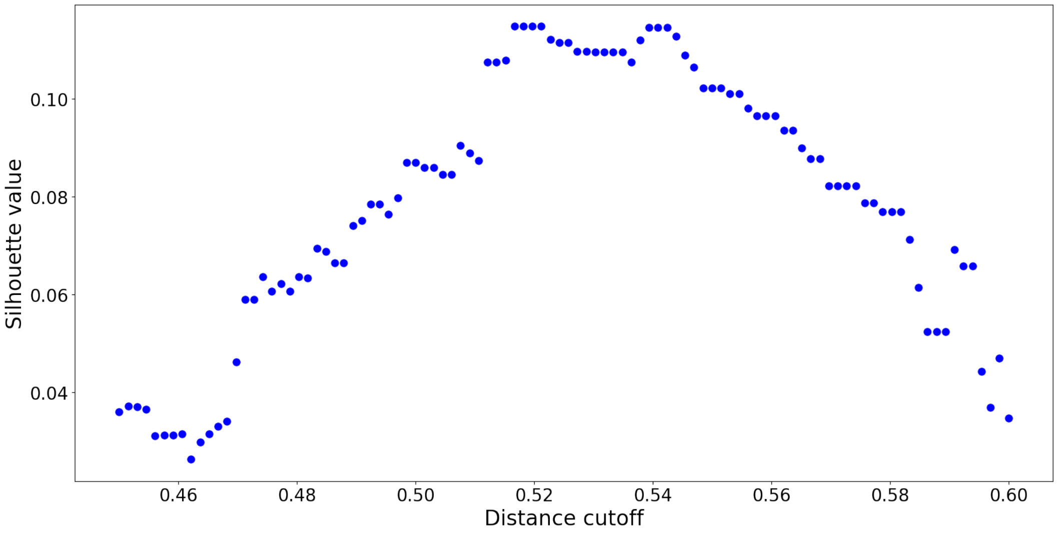


Figure 11. Silhouette values as a function of distance cut-off in UPGMA clustering. The optimal cutoff value is the one corresponding to the maximum Silhouette value.

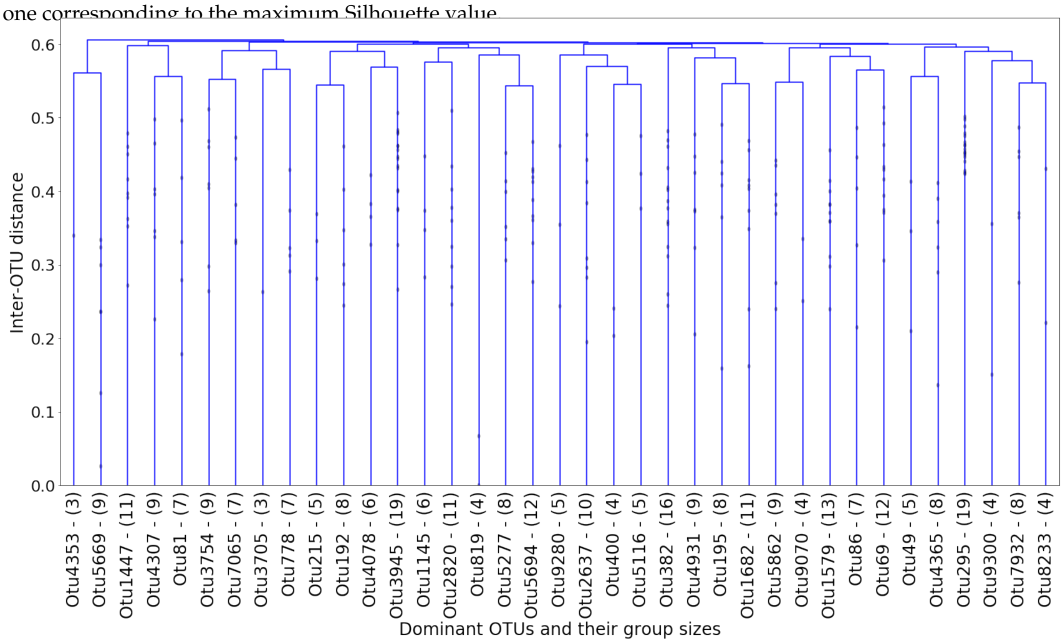


Figure 12. Dendrogram of the UPGMA hierarchy with optimal distance cut-off, at a depth of $K = 37$ groups. Each branch is labelled with the dominant OTU, and the number of its followers.

28, 35, and 37 which all contain 4 OTUs. Despite the considerable variance in community sizes, no communities contain less than 3 OTUs or more than 20 OTUs. The membership distribution can be observed in the reverse histogram, where the number of groups for each membership size is shown:

Fig. 14 shows that most clusters contain 4, 8, and 9 OTUs, followed by 5 and 7 OTUs. Most clusters have a population ranging between 4 and 12 OTUs, which indicates a healthy clustering distribution.

For the subsequent prediction and feature extraction steps, only the 37 dominant OTUs shown in Fig. 12 are considered, out of the total 305 OTUs to begin with. Although 37 is still a reasonablly large number (and not between 2 and 10, ideally), the choice is based on a combination of statistically-justified methods.

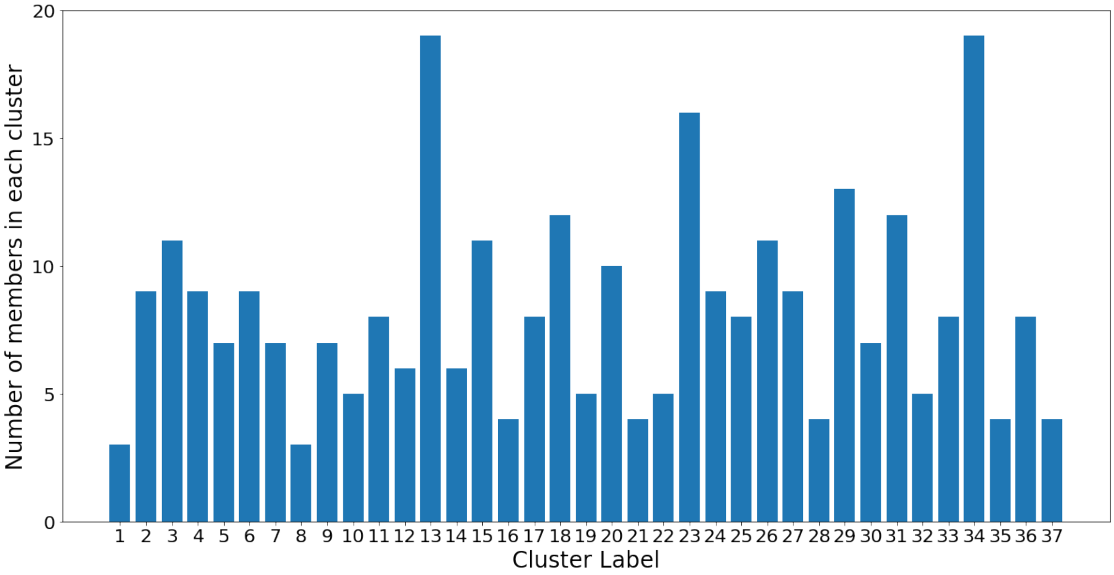


Figure 13. Cluster populations for UPGMA dendrogram with $k = 37$ groups.

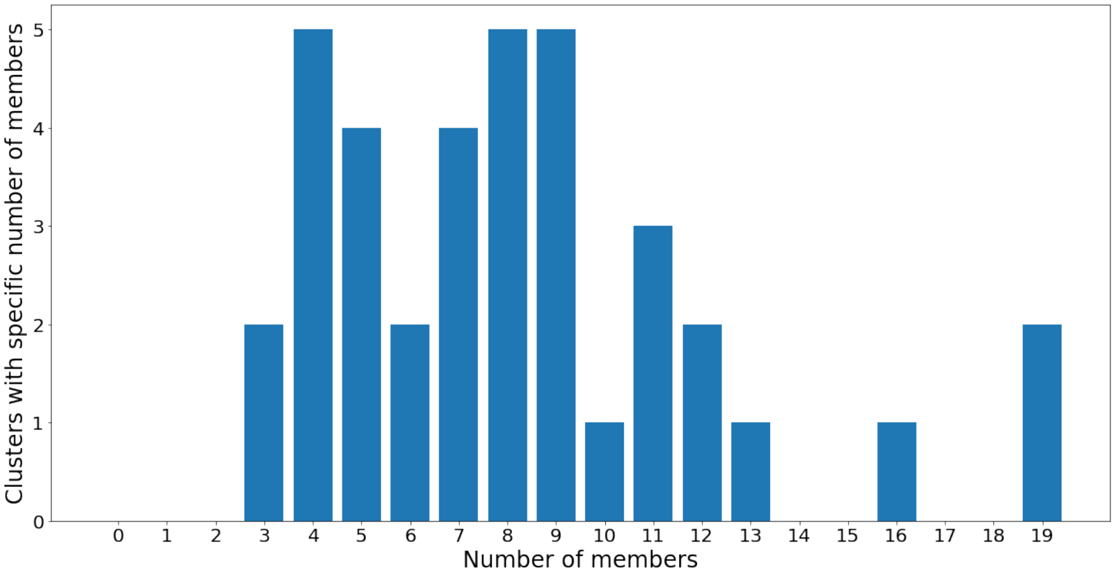


Figure 14. Membership distribution for UPGMA dendrogram with $K = 37$ clusters.

5.2 Gaussian Mixture Analysis of OTUs

Instead of using hierarchical clustering, another possible approach is to group OTUs using *Gaussian Mixture Models (GMMs)*. The assumption here is that the underlying distribution behind the OTU abundances can be modelled as a sum of multivariate Gaussians. Each Gaussian can be considered a "cluster" of OTUs, with its centroid represented by the mean, and its spread (or size) represented by its variance. The overall GMM is built using the *scikitlearn* subpackage *mixture.GaussianMixture*. In order to determine the “optimal" number of Gaussians K, the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) values are determined for each value of K. This is performed by calling the *.aic* and *.bic* attributes of the GMM models within *scikitlearn*. The results are plotted in the following Fig. 15 and Fig. 16:

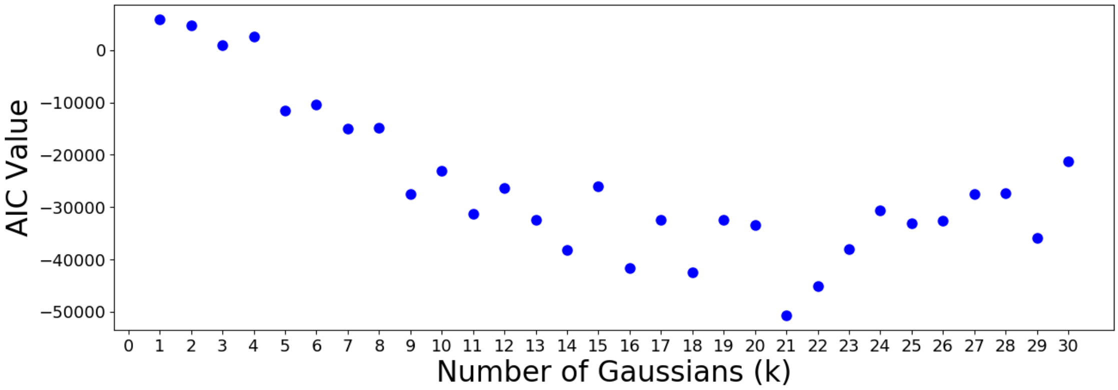


Figure 15. AIC values for GMMs with cluster sizes $1 < K < 30$. The minimum occurs at $K = 21$, which is selected as the desired number of groups.

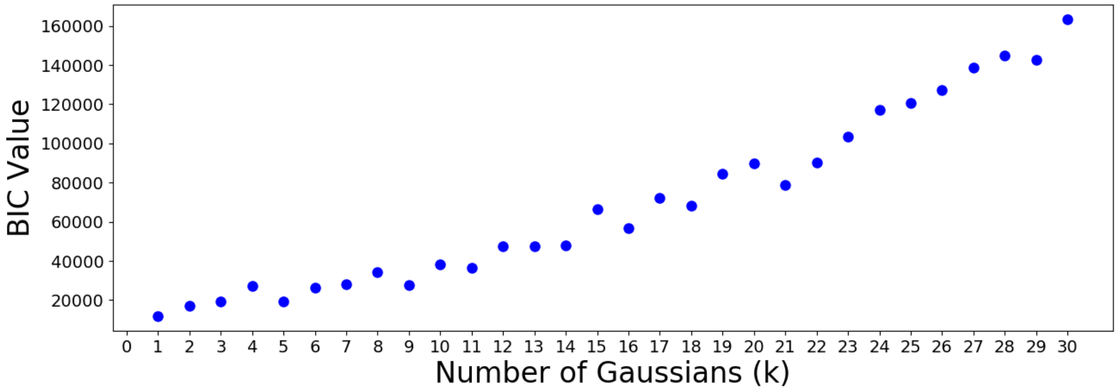


Figure 16. BIC values for GMMs of cluster sizes $1 < K < 30$. The minimum occurs at $K = 1$, indicating that one single group should be considered. This is an impractical result and is therefore discarded.

The AIC minimum suggests that the 305 OTUs should be optimally clustered into a GMM with $K = 21$ groups. On the other hand, the BIC minimum suggests that a GMM with only one cluster is optimal. This is a meaningless result which should be discarded, since it suggests that all OTUs are similar. Note that the BIC values increase almost monotonically from $K = 1$ group onwards, meaning no suitable number of clusters can be determined using this criterion. Therefore, the AIC result is used to move forward.

The cluster population and membership plots can be observed in the following Fig. 17 and Fig. 18:

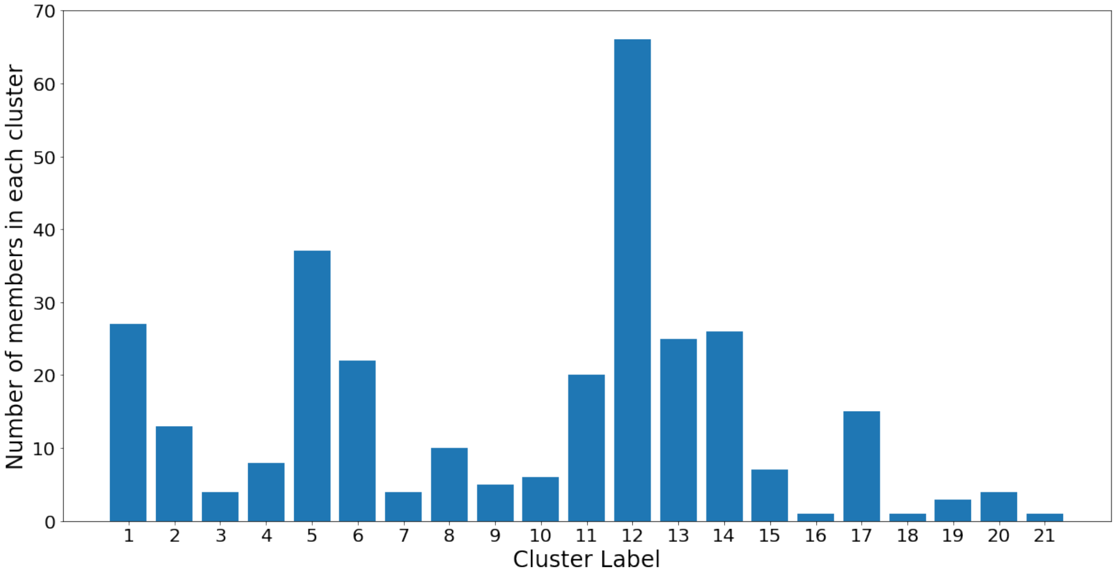


Figure 17. Populations for the AIC-optimal GMM model with $K = 21$ clusters.

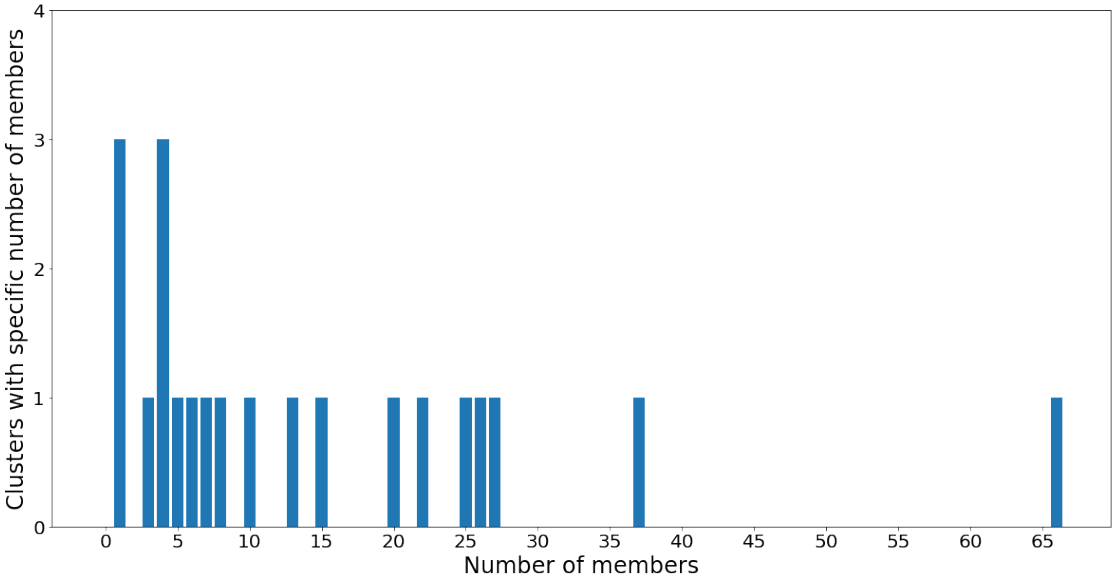


Figure 18. Membership for the AIC-optimal GMM model with $k = 21$ clusters.

Notice that the GMM cluster sizes have a much higher variance than the hierarchical clusters. Cluster 12 contains 66 out of the 305 total OTUs, while most other clusters contain between 2 and 40 OTUs. The skewed nature of the results is most likely due to the log-transformed OTU abundances being skewed towards the low counts. Therefore, the underlying Gaussian assumption (which assumes symmetrical distributions) is inaccurate. Moreover, the Gaussian mixture models were constructed using the *abundance counts* of OTUs, and not the *associations* as the hierarchical models were in Section 5.1. These two reasons alone explain why the Gaussian clusters identified are inconsistent across runs, and turned out to provide little intuition regarding the microbial community effects. Nevertheless, the results are summarized in the following Table 4, which highlights the dominant OTU in each GMM cluster as well as the cluster size.

Table 4. Dominant OTUs from Gaussian Mixture clusters

Group Number	Dominant OTU	Group Size
1	OTU200	27
2	OTU46	13
3	OTU11	4
4	OTU112	8
5	OTU3313	37
6	OTU470	22
7	OTU6	4
8	OTU2756	10
9	OTU157	5
10	OTU48	6
11	OTU3057	20
12	OTU185	66
13	OTU559	25
14	OTU778	26
15	OTU8968	7
16	OTU14	1
17	OTU105	15
18	OTU8	1
19	OTU77	3
20	OTU93	4
21	OTU1	1

5.3 Dirichlet Mixture Analysis of OTUs

In the previous Section 5.2, the OTU abundances were assumed to follow an underlying Gaussian distribution. In light of Fig. 7 and Fig. 8, this assumption is clearly inaccurate, since even the distribution of log-transformed values appears to be skewed towards the low counts. Therefore, a more suitable assumption for the OTU clusters is the *Dirichlet Multinomial Mixture (DMM)* (see Section G). Instead of using *Python*, the Dirichlet Multinomial *R* package developed by [38] is used. This algorithm is capable of constructing a set of DMM models, assessing the optimal model(s) using AIC, BIC, or Laplace Information Criterion (LIC), then producing heatmaps of the clustering results based on the Dirichlet weights of each cluster.

Unlike the hierachical or Gaussian approaches where the clustering is performed on the OTUs and not the samples, the DMM clustering is the exact opposite: the samples are clustered and not the OTUs. The results, however, can still be interpreted to identify the dominant OTUs for further analysis. For the 55 existing samples, the heatmap in Fig. 19 shows the BIC-optimal DMM clusters, labelled with the 20 OTUs of highest Dirichlet weights.

Note that the rows of the heatmap represent individual OTUs. In the first row (OTU6), a dark-shaded band exists in the mid-samples, including a commonly high abundance of OTU6 in those samples and low abundances elsewhere. Similarly, for the second row (OTU4), a common high-abundance band is observed for the first and last few samples, with low abundances elsewhere. This visual result reinforces the concept that the DMM model clusters the individual samples (columns) and not the OTUs. However, notice that going down the heatmap from OTU6 to OTU35, the dark-shaded bands appear less frequently. The colours become increasingly white, indicating an

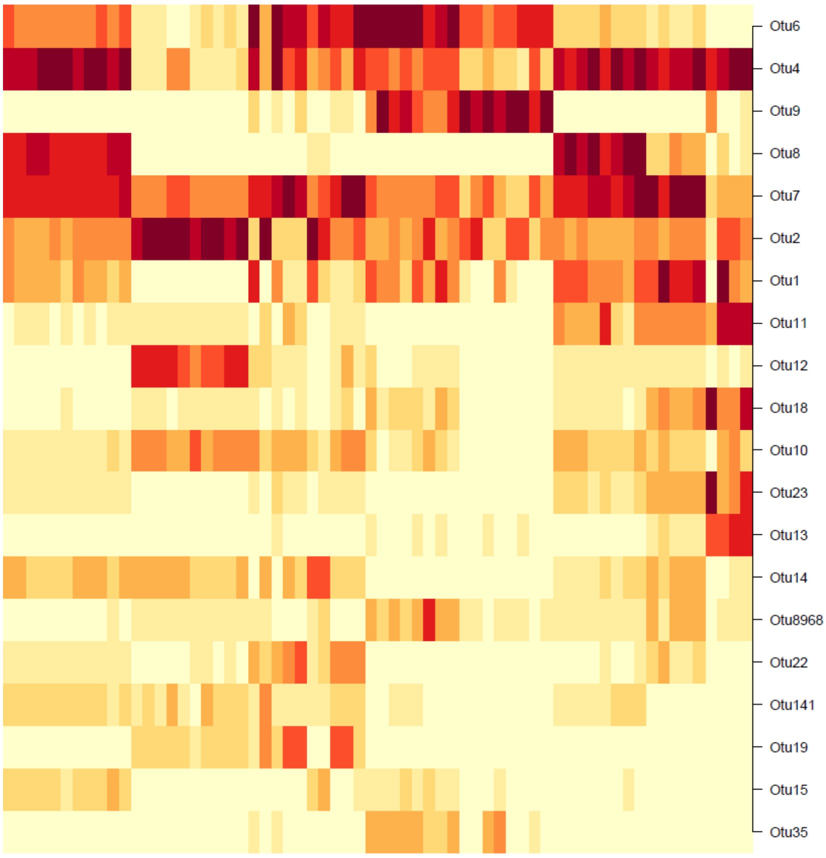


Figure 19. Heatmap of the BIC-optimal DMM model, with respect to the 20 highest-weighting OTUs. Colours are coded according to log-transformed OTU abundances; dark colour indicates high OTU abundance, and vice versa.

overall decrease in OTU abundance. Below the 20^{th}-OTU cutoff (of OTU35), the rows are entirely white with close to zero abundance, and therefore those results have been truncated from the figure. Therefore, the 20 OTUs shown in Fig. 19 are considered as the dominant OTUs, akin to those in the hierarchical and GMM clustering results. However, the followers of these 20 dominant OTUs cannot be determined, since the clustering was not performed OTU-wise.

5.4 Prediction Results

The 10 water chemistry variables (outlined in Table 2) are combined with representative OTUs obtained from Sections 5.1, 5.2, and 5.3. Together, these serve as inputs. When combined with the corresponding, labelled process outcomes of *Selenium Reduction Rate (SeRR)*, predictive models are trained for the estimation of *SeRR* of new samples.

The models can be categorized in terms of their inputs, as follows:

1. **Base case:** Water chemistry variables only.
2. **Hierarchical:** Water chemistry variables plus representative OTUs obtained using hierarchical clustering.
3. **Gaussian:** Water chemistry variables plus representative OTUs obtained using GMMs.
4. **Dirichlet:** Water chemistry variables plus representative OTUs obtained using DMMs.

The idea is to observe whether the addition of biological features improves or confounds the predictive capabilities of these models. The actual models consist of the following three types:

1. **Random Forests (RFs)**

2. **Support Vector Machines (SVMs)**
3. **Artificial Neural Nets (ANNs)**

The raw *SeRR* values obtained from plant data were normalized and discretized into two (binary) classes, 0 and 1. Class 0 *(poor)* corresponds to *SeRR* values which fall below the mean *SeRR*, and Class 1 *(satisfactory)* corresponds to values above the mean. Out of the $N = 56$ total data samples, 29 have a class label of 0 and 27 have a class label of 1, therefore the overall distribution is fairly even (i.e. not skewed towards one label).

For each model, 40% of samples from each class are randomly selected as *test samples* for performance assessment, and the remaining 60% of samples as *training samples* for model construction. Note that this approach eliminates the possibility of biased selection from either class. If the training and testing sets were instead selected arbitrarily *from the entire dataset*, then they could possibly be skewed (ex. many samples selected from Class 1, but few from Class 0).

No *validation (development)* set was required, since the hyperparameters of each model (i.e. regularization constants, model complexity, etc.) were selected to be fixed values for simplicity. The RF model was constructed using the *RandomForestClassifier* module from *scikitlearn.ensemble*, with bootstrapping disabled. Although bootstrapping is normally recommended, the data sample-size in this case ($N = 56$) is extremely small for modelling purposes. Therefore, all of the existing $56 \cdot 0.6 \simeq 34$ samples are required for training; any arbitrary selection of samples without replacement could skew the training set. The SVM model was constructed using the *sklearn.svm.svc* module, with a regularizer value of $C = 1$ and the default linear kernel. Finally, the ANN model was constructed using *tensorflow*, with 10 layers of 20 neurons each, a learning rate of $\alpha = 0.01$ and a ℓ_2-regularizer of $\alpha = 0.1$. In order to maintain reasonable computational times required by each model, a maximum of 1000 epochs (or "outer iterations") were allowed. The ANN model was allowed 50 steps (or "inner iterations") per epoch.

The prediction accuracy of each model on the test set (of $56 \cdot 0.4 \simeq 22$ samples) is reported in the following Table 5, with respect to the type of inputs used.

Table 5. Prediction results for each model type.

	Base Case	**Hierarchical**	**Gaussian**	**Dirichlet**
RF	96.3	90.6	93.4	92.2
SVM	91.8	87.2	91.3	90.6
ANN	81.7	78.6	83.4	80.2

The RF models produced the most accurate test predictions for every case, followed by SVMs then ANNs. When comparing the input types, the base case accuracy turned out to be the highest for both RF and SVM models. The addition of hierarchical OTU clusters had the largest detrimental effect on the test accuracy, as observed by the uniform, marked decreases across all three model types. The addition of Gaussian OTU clusters improved the test accuracy for the ANN model, but proved to be detrimental for the RF and SVM models, albeit with the least impact. The addition of Dirichlet OTU clusters also decreased the model accuracy for all three models, but not as much as the hierarchical. These results clearly show that the addition of biological data, which was initially expected to improve quality of prediction, actually degrades it. Even though the OTU abundances should contain valuable insight into the biological community interactions, the observed confounding effect is most likely due to the undesirable qualities of the data. These include the inherent noise present in the OTU abundances, and also the relatively low sample size ($N = 56$) to begin with. Another reason could be that the explored clustering methods are incapable of cleraly extracting information related to coupling effects between OTUs and water chemistry variables.

If a model were to be selected for actual prediction of process outcomes, it would be the RF using base-case, water chemistry variables. This model achieves a respectable $> 95\%$ accuracy on the binary classification of $SeRR$.

5.5 Feature Selection Results

The *relevant* features in the prediction framework are defined as those which contribute significantly to the accuracy of the model. The results in Section 5.4 showed the RF model as the most accurate one out of the 3 modelling approaches, and therefore it will be used for feature analysis in this section. The univariate feature selection strategy, *Mean Decrease in Accuracy (MDA)*, was first used to determine "relevant" features in terms of predicting the outcome $SeRR$. A RF model was constructed for each of the input types of hierarchical, Gaussian, and Dirichlet clustering. 10000 permutations of MDA were performed for each RF model; the averaged feature importances for each are summarized in the following Tables 6, 7, and 8. Only the top 4 water chemistry and top 5 OTU features are reported for conciseness.

Table 6. MDA feature importances for hierarchical clustering

Feature	**MDA** (%)
$Se_{D,in}$	6.3
$Ammonia_{out}$	0.3
$EBCT$	0.2
$Nitrite_{out}$	0.2
OTU215	1.5
OTU2637	0.6
OTU1579	0.6
OTU49	0.6
OTU3945	0.5

Table 7. MDA feature importances for Gaussian clustering

Feature	**MDA** (%)
$Se_{D,in}$	7.1
$EBCT$	1.2
$Ammonia_{out}$	0.7
$Nitrite_{out}$	0.7
OTU57	1.5
OTU7347	1.1
OTU2765	0.9
OTU48	0.9
OTU7	0.8

Table 8. MDA feature importances for Dirichlet clustering

Feature	**MDA** (%)
$Se_{D,in}$	5.3
$EBCT$	1.9
$Nitrite_{out}$	1.1
COD_{in}	0.7
OTU35	1.4
OTU8	1.0
OTU7	1.0
OTU1	0.6
OTU9	0.5

Notice that $Se_{D,in}$ consistently appears in each table as the most "relevant" feature, as MDAs of $5 \sim 7\%$ are observed as this feature is permutated. $EBCT$ appears to be the second contender, causing accuracy drops of $1 \sim 2\%$ in most cases when permutated. $Ammonia_{out}$ and $Nitrite_{out}$ are the next most "relevant" features, however permutating them causes negligible accuracy drops of $(< 1\%)$ on the RF models. Therefore $Se_{D,in}$ and $EBCT$ can be comfortably concluded as the main deciders of overall selenium removal rate, in terms of all water chemistry variables. This result is logical from a domain-knowledge perspective, since both variables are used to calculation for $SeRR$ using a mass-balance approach.

On the other hand, no consistent OTU features are observed. The most consistent one is $OTU7$ which appears in the top-5 lists for both GMM and DMM clustering approaches. However, most OTU features show accuracy drops close to 1%, or less. These low MDA values render the representative OTUs indistinguishable from random-noise features, therefore no clear conclusion can be formed in terms of the OTU features. This result falls in line with those obtained in Section 5.4, which showed that the biological variables confounded the models, rather than providing clarity.

Note that the MDA approach is univariate, which means it ignores possible correlations or between existing features. In order to address this issue partially, the *Conditional Mean Decrease in Accuracy (C-MDA)* approach is also explored. In C-MDA, the permutations of features are performed, given the presence of other features. For example, when the feature $Se_{D,in}$ is permutated, it is conditioned on the fact that the feature $EBCT$ falls within a certain bracket of values. The R package developed by [33] is used to perform these C-MDA experiments, since the algorithm systematically decides the best values for the secondary variables to be conditioned upon. The detailed results can be found in Fig. A22, A23, and A24 in Section I. The "relevant" variables from each RF model can be summarized in the following Table 9:

Table 9. Overall CP feature importances

Rank	Feature
1	$Se_{D,in}$
2	$EBCT$
3	$Nitrite_{out}$
4	COD_{in}
5	$Nitrate_{in}$
6	$Ammonia_{out}$

Notice that $Se_{D,in}$ and $EBCT$ are, again, identified as the primary “relevant" features. These results agree with those obtained from ordinary MDA. The dominance of these two variables is logical, given that they both appear in the mass balance for calculation of $SeRR$. Moreover, C-MDA suggests that the secondary "relevant" features are nitrite outflow, COD inflow, nitrate inflow, and ammonia outflow (to a lesser extent), which are also similar results compared to those from MDA. Therefore, both feature selection methods suggest the existence of hidden interactions between these biological features and selenium removal. However, the exact functional forms of these interactions, as well as any domain-knowledge-related interpretations, are unclear from these feature selection methods. Therefore, these secondary features serve as, at best, *recommendations* for monitoring and control.

6 Conclusions

In this work, a two-fold data analysis was performed on a wastewater-treating bioreactor. First, the binary process outcome of selenium reduction rate was predicted using three model types - RF, SVM, and ANN. For each type of model, the use of four different input types were explored - water chemistry features only (base-case), UPGMA hierarchical clusters, Gaussian clusters, and Dirichlet clusters. The clustering methods were performed in order to reduce the large initial dimensionality of the biological features. Out of all model types, the RF model was the most accurate in terms of predicting outcomes on the test set. Unfortunately, the addition of biological information (in the form of OTU abundances) detrimentally affected the test prediction accuracies compared to using only the water chemistry variables. Actually, none of the three clustering techniques identified clusters of acceptable quality. Specifically, the hierarchical clusters had low accompanying Silhouette values, while the Gaussian clusters had high membership variation and were inconsistent between runs. The Dirichlet clusters were supposedly a better reflection of the true underlying probabilistic distribution of the OTUs. However, their addition still hurt the predictive models' performance. Out of the three clustering techniques, the hierarchical clustering approach had the largest confounding effect on model accuracy. In light of these results, a significant future effort is required on the revision of selected clustering techniques for meaningful feature extraction.

The second analysis was performed in regards to the determination of "relevant" process variables, through univariate (MDA) and conditionally-univariate (C-MDA) feature selection techniques. Both techniques show that the features of retention time and selenium inlet flowrate dominantly influence the selenium removal rate. This result is expected from a domain-knowledge perspective, in terms of mass balances. Interestingly, variables such as ammonia outflow, nitrite outflow, and COD also play significant roles in affecting selenium removal rate, even though the biological intuitions behind these results are not revealed. The feature importances obtained from MDA and C-MDA are similar, despite C-MDA being more of a multi-variate method. Several representative OTU features were identified to be "relevant," but their inconsistent results across the three clustering methods yield no interpretable conclusions.

Overall, the results show that the core assumptions behind the clustering were potentially incorrect, or incomplete. Specifically, the OTUs were assumed to fall into similar groups, with a "leader" in each group. However, they may share little similarities in reality, and have completely independent (instead of agglomerative) roles in affecting the process outcome. Although the stochasticity of the microbial community was not fully understood from the results, the contributions of this work are still non-trivial. Through the comparison and critique of various ML algorithms presented here, the reader is encouraged to perceive this work as a cursory endeavour into meaningful process analytics.

6.1 Directions for future work

In terms of predictive models, the ones used in this work performed adequately in terms of test-set accuracy. However, the Stochastic Configuration Network (SCN) approach [26] could be explored in a future work, in terms of its potential benefits to prediction accuracy as well as feature information extraction.

The proposed feature analysis methods require further revision, in order to improve the reliability of their results. Specifically, the underlying reasons behind the inconsistencies between hierarchical, GMM, and DMM clusters (and hence the representative OTUs) should be investigated. One potential factor may be the marked differences between each clustering method. A future paper which theoretically explores these points could potentially shed light on which clustering method is optimal, given a specific type of raw data. Some straightforward suggestions for improvement include the use of ensemble methods to produce a majority vote, over a large number of clustering experiments. Another possible improvement is the inclusion of chemical (or process) variables in the clustering decisions, although the exact implementation of this approach is not clear presently. Finally, the feature selection methods (MDA, C-MDA) should also be expanded to include multivariate interactions. A possible strategy would be to utilize Bayesian Networks [2] to map causal relationships between variables. This has been demonstrated as a feasible approach in recent publications, for example, [39].

Once the work on data pre-processing and analysis is complete, the next step would be to implement the obtained feature knowledge into to a controller. Specifically, these ML-guided results should be used by the controller to select relevant *Manipulated Variables (MVs)*, as well as decide on the optimal control actions. Future papers can be written regarding the efficacy of this ML-guided controller compared to well-known benchmarks, such as *Proportional-Integral-Derivative (PID)* [6] or

Model Predictive Control (MPC) [40]. The autonomy (i.e. self-driving characteristic) of this controller can be developed using ideas from the field of *Reinforcement Learning (RL)* [41].

Author Contributions: The individual contributions of the listed authors are as follows: conceptualization, Y.T. and S.B.; methodology, Y.T. and S.B.; software, Y.T. and L.C.S.; formal analysis, Y.T. and S.B.; investigation, Y.T. and S.B.; resources, Y.T. and S.B. and L.C.S.; data curation, Y.T. and S.B. and L.C.S.; writing—original draft preparation, Y.T. and S.B.; writing—review and editing, Y.T. and S.B. and B.G.; visualization, Y.T. and L.C.S.; supervision, S.B. and B.G.; project administration, S.B. and B.G.; funding acquisition, S.B. and B.G.

Funding: This research was funded by the Genome British Columbia User Partnership Program, grant number UPP026 to S.A. Baldwin (P.I.) and B. Gopaluni (co P.I.)

Acknowledgments: The authors would like to gratefully acknowledge Shams Elnawawi, a senior undergraduate student whose extensive coding expertise contributed significantly to the software development in Python.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Net
ARIMA	Autoregressive with Integrated Moving Average
ARMA	Autoregressive with Moving Average
ARX	Autoregressive with Exogenous Inputs
C-MDA	Conditional Mean Decrease in Accuracy
COD	Chemical Oxygen Demand
DMM	Dirichlet Mixture Model
EBCT	Empty-Bed Contact Time
DL	Deep Learning
DR	Dimensionality Reduction
FIR	Finite Impulse Response
GMM	Gaussian Mixture Model
IID	Independent and Identically Distributed
LTI	Linear Time Invariant
MA	Moving Average
MDA	Mean Decrease in Accuracy
ML	Machine Learning
MPC	Model Predictive Control
MV	Manipulated Variable
NaN	Not a Number
OTU	Operational Taxonomic Unit
PID	Proportional-Integral-Derivative
SVM	Support Vector Machine
*r*RNA	Ribosomal Ribonucleic Acid
RF	Random Forest
RL	Reinforcement Learning
Se	Selenium
SeRR	Selenium Reduction Rate
UPGMA	Unweighted Pair-Group Method with Arithmetic Means
ZOH	Zero-Order Hold

A Standardization of data

Prior to predictive modelling, features of a dataset are often *standardized* or *normalized* in order to homogenize the importance of each feature, such that each ends up with zero mean and unit variance. Mathematically, each feature is scaled by the operation $x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j}$, using the respective feature means $\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_j^{(i)}$ and feature standard deviations $\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_j^{(i)} - \mu_j)^2}$.

B Details of the Random Forest (RF) model

Random Forests are a well-known model covered in many texts, such as [42] and [8]. Its main advantage is the convenience of implementation; many optimized packages (such as *scikitlearn*) exist which allow users to obtain results quickly even for large datasets. The goal of RFs is to map the raw features of a dataset to outcomes, which are discrete class labels $c \in [1, C]$. Each of the original d_x variables is split into two regions, one above and one below a threshold value θ. These regions become the *branches* of the first *split* on said variable. Each split is a conditional partition of a variable, which decides the final outcome. If a split on the first feature is insufficient to decide the final outcome, then a second split is performed off of the two branches from the first split. This continues until a clear outcome is realized.

A simple example demonstrating the partitioning of 2 features is provided in the following Table A1.

Table A1. Feature partitioning for a 2-feature decision tree, with $2^2 = 4$ possible partitions. Each partition is labelled using a number between 0 and 3. The threshold values θ decide which partition each sample falls under.

	$x_1 < \theta_1$	$x_1 > \theta_1$
$x_2 < \theta_2$	0	1
$x_2 > \theta_2$	2	3

To model all possible outcomes, 2^{d_x} partitions or *branches* are potentially required in total (where d_x is the total number of features). The computation cost of this calculation becomes impractically large for common computing devices (such as PCs or laptops), as d_x approaches numbers as small as 15. If d_x is extremely large (ex. hundreds or thousands), the outcome-space cannot be feasibly mapped out in its entirety. However, it can be approximately sampled using the concept of *Random Forests (RFs)* [43]. In this approach, a *random* subset of all d features is selected and split on; the tree constructed using these arbitrarily-selected features is called a RF. Since not all d_x features can be accounted for in a single RF, a large number of RFs are constructed (i.e. thousands or more) and the predicted class labels are determined by taking a majority vote across all obtained outcomes. An example of this is illustrated in the following Fig A1.

The threshold value θ used for each split is determined by a simple scoring rule in most cases [23]. For example, the feature x_1 may have a range of values $x_{1,\text{min}} < x_1 < x_{1,\text{max}}$. A computational routine would define an arbitrary step-size ϵ (usually a fraction of the gap $x_{1,\text{max}} - x_{1,\text{min}}$), then start with the threshold value $x_{1,\text{min}} + \epsilon$ and work all the way up to $x_{1,\text{max}} - \epsilon$. The final threshold value is selected as the one resulting in the highest model accuracy (in terms of training).

In order to construct RFs which produce an unbiased estimate of the true class label for each given data sample, a technique known as *bootstrapping* or *bootstrap aggregating ("bagging")* can be used [23] [44]. Each RF randomly selects from the total N data samples to train on, *with replacement*, such that over a large number of RFs the total number of samples selected turns out to be approximately $0.63N$. Bootstrapping also mitigates numerical instabilities, which can occur with RFs and are especially common in complex models such as ANNs. However, it is only viable if the sample size N is sufficiently

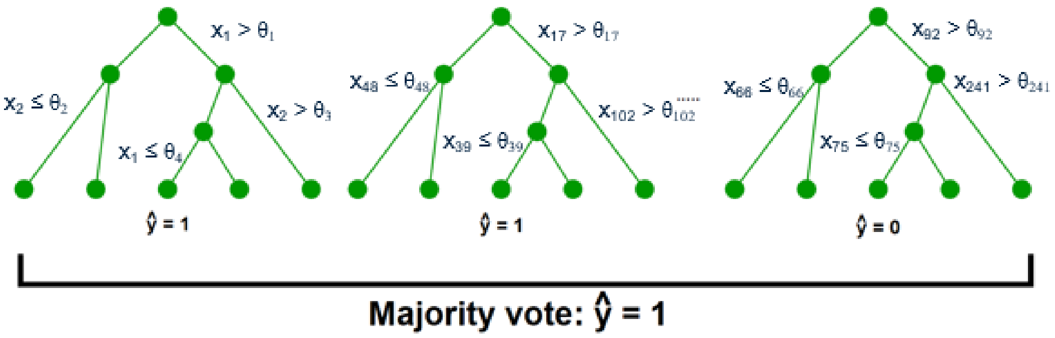


Figure A1. Multiple random forests constructed for a binary-class problem. The outcomes (either Class 0 or 1) are decided by combining sequential splits of d_k randomly selected features, from the original d_x-dimensional feature space. The final outcome is determined by a majority vote of individual outcomes from all trees.

large (thousands or more). When bootstrapping on small datasets (N on the order of hundreds or less), special care must be taken to bootstrap over a large number of iterations.

C Details of the Support Vector Machine (SVM) model

The Support Vector Machine maps existing samples of a training set to their corresponding given classes, such that the classes of new samples can be predicted. However, instead of partitioning on binary splits of each feature like RFs, SVMs directly find the *separating boundaries* between the classes of data. *Support vectors* are the vectors between the closest data sample in each class to the separating boundaries [24]; the distances of these vectors are maximized in order to optimize the extent of separation between classes. Although the boundaries can be found using the *hinge-loss* function, computational routines today instead use the *softmax* function as a smooth approximation of the hinge-loss. This approximation improves the numerical stability in solving for the SVM model via *gradient descent*, while not hindering its accuracy [45]. The softmax calculates the probability p that each sample $\boldsymbol{x}^{(i)}$ belongs in class $c \in [1, C]$. The well-known *logistic regression* is the special-case of softmax for the binary (2-class) scenario. The parameters $\boldsymbol{w}$ represents the model coefficients corresponding to each specific class. Specifically, the $\boldsymbol{w}_c$ terms represent the model weights, assuming sample $\boldsymbol{x}^{(i)}$ belongs in class c. Similarly, the terms $\boldsymbol{w}_{y^{(i)}}$ represent coefficients assuming sample $\boldsymbol{x}^{(i)}$ has a class label $y^{(i)} \in [1, C]$. Using these concepts, the softmax probability for any sample can be expressed as:

$$p(y^{(i)}|\boldsymbol{w}, \boldsymbol{x}^{(i)}) = \frac{exp(\boldsymbol{w}_{y^{(i)}}^{\top} \boldsymbol{x}^{(i)})}{\sum\limits_{c=1}^{C} exp(\boldsymbol{w}_c^{\top} \boldsymbol{x}^{(i)})}. \tag{A1}$$

An example of multi-class SVM with 4 classes ($C = 4$) is shown in Fig. A2.

Data that is *linearly-separable* allows linear boundaries to be drawn to separate the different classes. The equations of these separating hyperplanes can be obtained using methods described in [24]. On the other hand, data that is *non-linearly-separable* cannot be accurately modelled by linear separating boundaries. In these cases, the *kernel trick* [7] can be used to construct high-dimensional feature spaces in which the data becomes linearly separable.

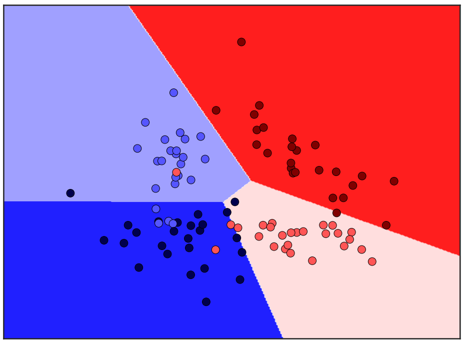


Figure A2. Multi-class SVM for 4 classes. The hyperplanes (lines) in the 2-D space clearly separate the 4 distinct classes with acceptable misclassification rates. A smooth approximation can be made using a 4-class softmax function.

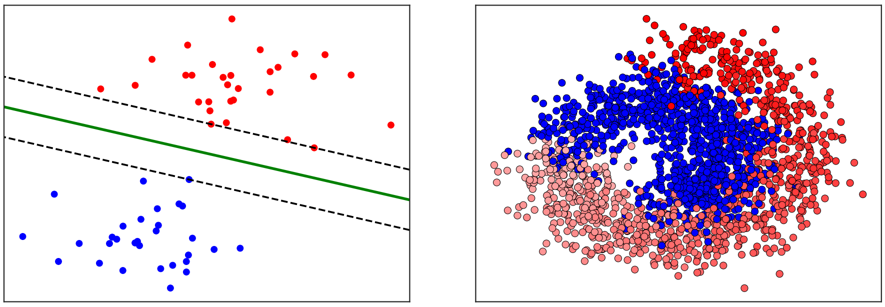


Figure A3. A linearly-separable dataset (left), versus a non-linearly-separable dataset (right), adapted from [46].

D Details behind Artificial Neural Networks (ANNs)

Unlike least squares or SVMs which can only perform regression or classification, respectively, ANNs can predict either continuous (regression) or discrete (classification) outputs. The first layer in an ANN consists of an activation function acting upon an affine, i.e. $\boldsymbol{y} = A(\boldsymbol{WX} + \boldsymbol{b})$. The function A is usually a non-linear transformation of its linear argument $(\boldsymbol{WX} + \boldsymbol{b})$. If A were chosen to be linear in every layer of the network, the whole ANN would trivially reduce to a linear least-squares model.

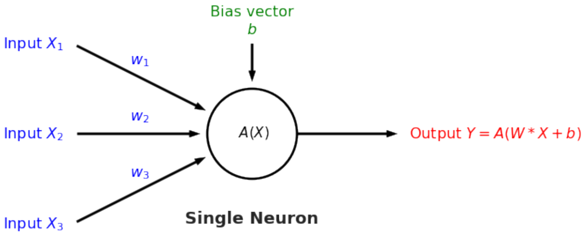


Figure A4. Visualization of the operation $\boldsymbol{y} = A(\boldsymbol{WX} + \boldsymbol{b})$ in a single ANN node. The weighted sum of its inputs is added to a bias term; the final sum is transformed by a nonlinear activation function chosen by the user.

Subsequent layers follow the same affine-activation transformation, i.e. $\boldsymbol{z}_i^{[l+1]} = \boldsymbol{A}(\boldsymbol{w}\boldsymbol{z}^{[l]} + \boldsymbol{b})$. For each neuron z, the subscript represents the neuron number, while the superscript represents the layer in which the neuron is located. The following Table A2 contains some commonly-used activation functions within ANNs:

Table A2. Typical activation functions for neural networks

Activation	Abbreviation	Formula
Affine	$aff(z)$	$wz + b$
Step	$S(z)$	$\begin{cases} 0, & \text{if } z < 0 \\ 1, & \text{if } z \geq 0 \end{cases}$
Sigmoid	$sig(z)$	$\frac{1}{1+e^{-z}}$
Hyperbolic Tangent	$tanh(z)$	$\frac{e^z - e^{-z}}{e^z + e^{-z}}$
Rectified Linear Unit	$ReLU(z)$	$max(0, z)$
Leaky Rectified Linear Unit	$LReLU(z)$	$max(\alpha z, z)$

The entire neural net can be visualized as the following structure, with inputs entering the leftmost side and outputs exiting right-most side.

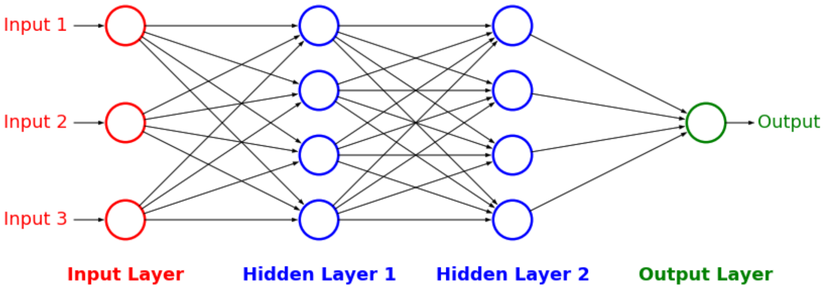


Figure A5. Conventional ANN structure with two hidden layers.

E Details of hierarchical clustering

The four main types of hierarchical clustering used in literature are [28]:

1. **Single Linkage (Nearest-Neighbour)**: "Nearest-neighbour" clustering. Initially, each sample is considered a centroid. The pair of samples with the smallest distance between them is merged together; subsequent clusters are merged according to the distances between their closest members. The linkage function is expressed as:

$$D(C_p, C_q) = \min_{\boldsymbol{x}^{(i)} \in C_p, \boldsymbol{x}^{(j)} \in C_q} d(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}). \tag{A2}$$

C_p and C_q represent two arbitrary clusters, and D the distance between them.

2. **Complete Linkage (Farthest-Neighbour)**: Also known as "farthest-neighbour" clustering. Identical to single linkage, except clusters are merged together according to distances between their farthest members. The linkage function is expressed as:

$$D(C_p, C_q) = \max_{\boldsymbol{x}^{(i)} \in C_p, \boldsymbol{x}^{(j)} \in C_q} d(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}). \tag{A3}$$

3. **Agglomerative Averages**: Also known as "average" clustering. Identical to single linkage, except clusters are merged together according to average distances between their members. The linkage function is expressed as:

$$D(C_p, C_q) = \frac{1}{|C_p||C_q|} \sum_{\boldsymbol{x}^{(i)} \in C_p} \sum_{\boldsymbol{x}^{(j)} \in C_q} d(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}), \tag{A4}$$

where $|C_p|$ represents the number of samples in each cluster, during each iteration.

4. **Ward's Method**: Also known as "minimum-variance" clustering. Instead of merging samples or clusters together based on distance, it starts by assigning "zero variance" to all clusters. Then, an Analysis of Variance (ANOVA) test is performed: two arbitrarily-selected clusters are merged together. The "increase in variance" is calculated as:

$$\Delta(C_p, C_q) = \frac{|C_p| \cdot |C_q|}{|C_p| + |C_q|}||\bar{C}_p - \bar{C}_q||_2^2 \tag{A5}$$

for all pairwise clusters. $\bar{C}_p$ represents the centroid coordinates for cluster C_p. The pair of clusters that results in the smallest increase in variance is then merged at each iteration.

The Agglomerative Average approach includes the subroutines **Unweighted Pair-Group Method with Averages (UPGMA)**, **Unweighted Pair-Group Method with Centroids (UPGMC)**, **Weighted Pair-Group Method with Averages (WPGMA)**, and **Weighted Pair-Group Method with Centroids (WPGMC)**, which are discussed in detail in [47]. The difference between these methods lie within the use of averaged Euclidean coordinates versus pre-determined centroids, and whether each data sample contribution is equal or weighted (with the weights determined by some *a priori* information).

The confidence of clustering results can be quantitatively assessed by two metrics:

1. **Cophenetic correlations [29]:** Measures how well a specified clustering method preserves original pairwise distances between samples. In other words, how similar are the average inter-cluster distances between pairwise points compared to their actual distances? The formula is:

$$\frac{\sum\limits_{i \neq j}(d(x^{(i)}, x^{(j)}) - \bar{d})}{\sqrt{[\sum\limits_{i \neq j}(d(x^{(i)}, x^{(j)}) - \bar{d})^2][\sum\limits_{i \neq j}(cd(x^{(i)}, x^{(j)}) - \bar{cd})^2]}} \tag{A6}$$

which returns a value between 0 and 1, where $\bar{d}$ represents average distances from all pairs of $x^{(i)}$,$x^{(j)}$. cd represents the **Cophenetic distance** between two pairwise points $x^{(i)}$ and $x^{(j)}$, defined as the distance from the base of the dendrogram to the first node joining $x^{(i)}$ and $x^{(j)}$.

2. **Silhouette analysis [30]:** Measures the optimal depth of a specified clustering method. Mathematically, it assesses how well each sample $x^{(i)}$ belongs to its assigned cluster C_p. Each individual Silhouette number is evaluated as:

$$s^{(i)} = \frac{\bar{x}_{C_q}^{(i)} - \bar{x}_{C_p}^{(i)}}{max(\bar{x}_{C_q}^{(i)}, \bar{x}_{C_p}^{(i)})} \tag{A7}$$

where C_q represents the closest cluster to each C_p. At each depth on the dendrogram, the average Silhouette number is evaluated across all samples and calculated as $\bar{s} = \frac{1}{N}\sum\limits_{i=1}^{N} s^{(i)}$. The depth with the highest $\bar{s}$ is then selected for that particular clustering scheme.

By combining the **Cophenetic** and **Silhouette** analyses as outlined above, the "most confident" clustering method (i.e. UPGMA vs. Ward vs. single-linkage vs. complete-linkage) and the optimal clustering depth, respectively, can both be selected.

F Details behind Probabilistic Mixtures

The motivation behind using *probabilistic mixtures* is to model the underlying distributions of the given data. Models using one distribution are sufficient for uni-modal systems, but fails to capture multi-modal systems effectively. Therefore, data are usually modelled as the *sums* of various probabilistic distributions, with the structure of said distributions specified as a prior assumption. Mixture models are different from the hierarchical models mentioned in 2.2.1. The difference lies in the assumption that in mixtures, each individual species is assigned a group to which it is similar, but overlaps may occur between multiple groups. In other words, each species may belong to more than one group. This introduces a degree of stochasticity which makes these models more flexible. The two mixtures used in this paper are:

1. **Gaussian Mixtures [8]:** $p(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$; underlying distribution is assumed to be a sum of K weighted multivariate Gaussians with individual means and covariances. The term w_k represents the weighting factor for each Gaussian. Each Gaussian has the formula $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^{d_X} \cdot \det(\boldsymbol{\Sigma}_k)}} \cdot exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \cdot \boldsymbol{\Sigma}_k^{-1} \cdot (\boldsymbol{x}-\boldsymbol{\mu}_k)\right]$.
2. **Dirichlet Mixtures [19]:** Define $\boldsymbol{p}^{(i)}$ as a vector containing the probabilities that sample $\boldsymbol{x}^{(i)}$ belongs to each community species. The Dirichlet mixture *prior* over K distributions is $\boldsymbol{P}(\boldsymbol{p}^{(i)}) = \sum_{k=1}^{K} Dir(\boldsymbol{p}^{(i)} \mid \alpha_k)\pi_k$, where α_k are the *Dirichlet parameters* and π_k are the *Dirichlet weights*.

The Gaussian assumption is reasonable for most natural processes, which assumes that the underlying distributions are symmetric. When little a priori knowledge is available, it is a popular choice. However, if domain knowledge is available, it should be used to guide the choice of distribution used. For example, if OTU data mostly contains abundances skewed towards low counts, then the Dirichlet mixture will model the data more accurately than Gaussian. This type of mixture model is discussed in greater detail in the following Section G.

G Details behind the Dirichlet mixture

The detailed modelling equations behind Dirichlet distributions and mixtures can be found in [19]. A summary of the paper's results is as follows:

- The **likelihood** of observing each sample $\boldsymbol{x}^{(i)}$ is:

$$L^{(i)}[\boldsymbol{x}^{(i)}|\boldsymbol{p}^{(i)}] = \left[\sum_{j=1}^{d_{OTU}} x_j^{(i)}\right]! \prod_{j=1}^{d_{OTU}} \frac{[p_j^{(i)}]}{x_j^{(i)}} \tag{A8}$$

  where d_{OTU} is the total number of OTU species, $\boldsymbol{p}_j^{(i)}$ the probability that sample i belongs to species j, and $\boldsymbol{X}_j^{(i)}$ the abundance count of species j in sample i.
- The **total likelihood** across all samples is therefore:

$$L(\boldsymbol{X} \mid \boldsymbol{p}^{(1)}, \cdots, \boldsymbol{p}^{(N)}) = \prod_{i=1}^{N} L^{(i)}(\boldsymbol{x}^{(i)} \mid \boldsymbol{p}^{(i)}) \tag{A9}$$

- The **Dirichlet distribution** is modelled as:

$$Dir(\boldsymbol{p}^{(i)} \mid \theta \boldsymbol{m}) = \Gamma(\theta) \prod_{j=1}^{d_{OTU}} \frac{[p_j^{(i)}]^{\theta m_j - 1}}{\Gamma(\theta \boldsymbol{m}_j)} \delta\left(\sum_{j=1}^{d_{OTU}} p_j^{(i)} - 1\right) \tag{A10}$$

  where θ represents the Dirichlet precision (i.e. large θ implies all $p_j^{(i)}$ values lie close to the mean p value, and vice versa), $\boldsymbol{m}$ a normalization constant such that $\sum_{j=1}^{d_{OTU}} m_j = 1$, and δ the Dirac delta function which ensures further normalization.

- The **Dirichlet mixture prior over *K* distributions** is:

$$\boldsymbol{P}(\boldsymbol{p}^{(i)} \mid Q) = \sum_{k=1}^{K} Dir(\boldsymbol{p}^{(i)} \mid \alpha_k)\pi_k \tag{A11}$$

where $\alpha_k = \theta \boldsymbol{m}_k$ are the *Dirichlet parameters*, π_k the *Dirichlet weights*, and $Q = (K, \alpha_1, \cdots, \alpha_K, pi_1, \cdots, \pi_K)$ the complete set of mixture hyperparameters.

- The **Dirichlet mixture posterior over *K* distributions** is:

$$\boldsymbol{P}(\boldsymbol{p}^{(i)} \mid \boldsymbol{x}^{(i)}, Q) = \frac{\sum_{k=1}^{K} L^{(i)}(\boldsymbol{x}^{(i)} \mid \boldsymbol{p}^{(i)}) Dir(\boldsymbol{p}^{(i)} \mid \alpha_k)\pi_k}{\sum_{k=1}^{K} \boldsymbol{P}(\boldsymbol{x}^{(i)} \mid \alpha_k)\pi_k} \tag{A12}$$

H Time plots of process variables over time

The time-plots of all water chemistry variables (in Table 2) are provided here. This enables a visual analysis of the variations over time. Due to proprietary reasons, the values have been *normalized* using the pre-processing procedure outlined in Section 4. The plots are separated by reactor number, i.e. $Reactor_1$ and $Reactor_2$ to distinguish the behaviour in the two different reactors. The horizontal time-axis represents duration measured in *hours*.

H.1 Time-plots from Reactor 1

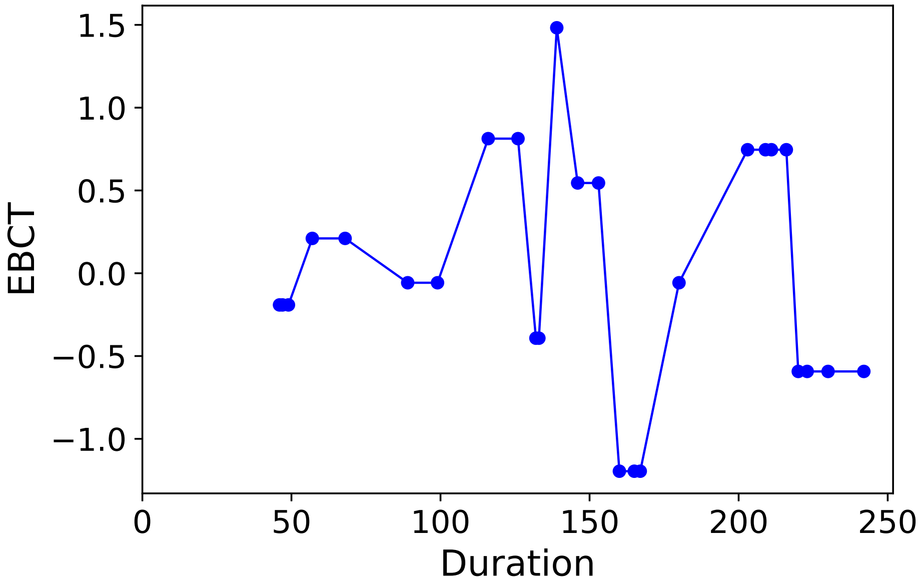


Figure A6. Normalized empty-bed contact time for **Reactor 1**.

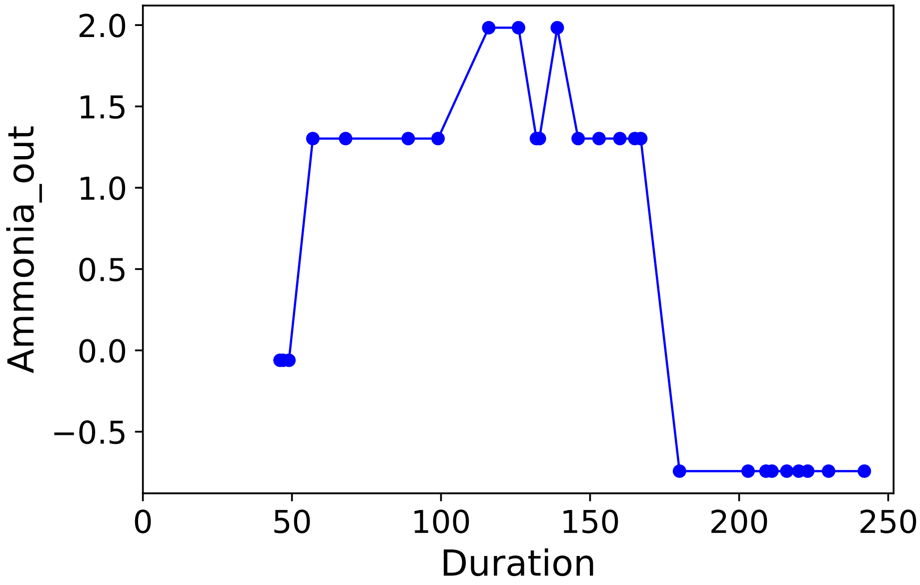


Figure A7. Normalized ammonia outlet flowrate for **Reactor 1**.

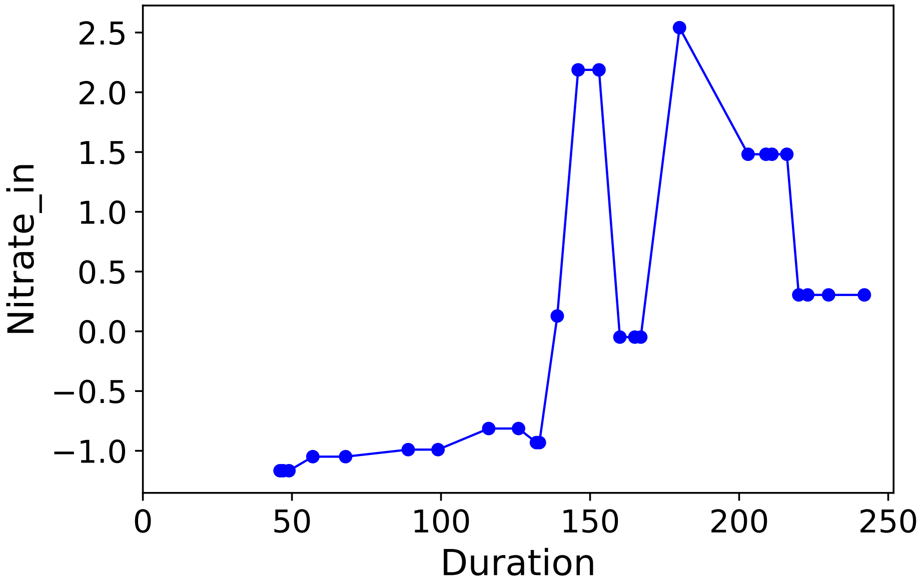


Figure A8. Normalized nitrate inlet flowrate for **Reactor 1**.

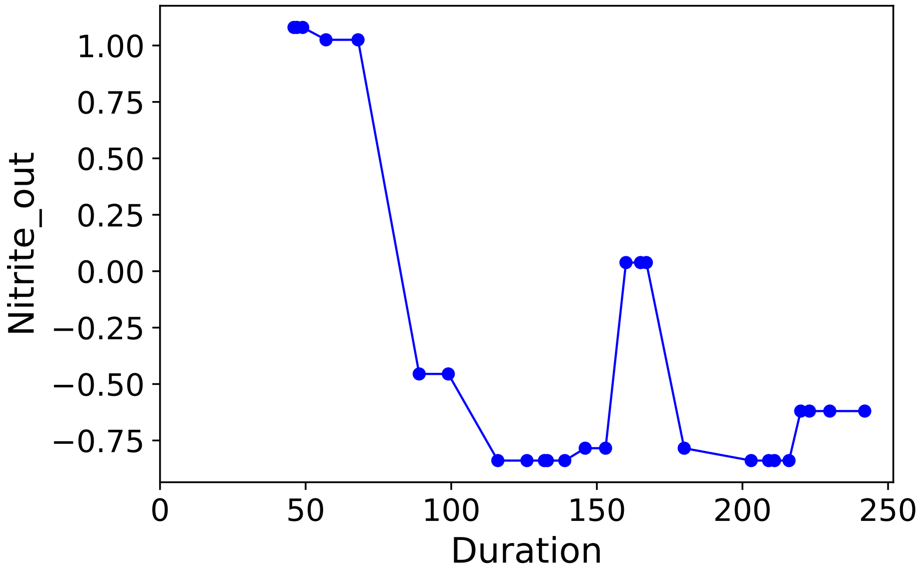


Figure A9. Normalized nitrite outlet flowrate for **Reactor 1**.

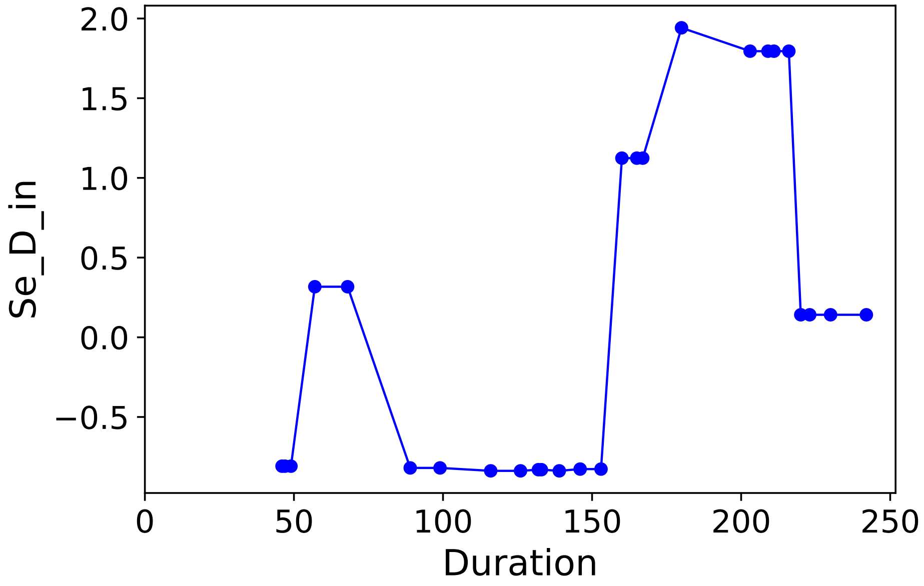


Figure A10. Normalized selenium inlet flowrate for **Reactor 1**.

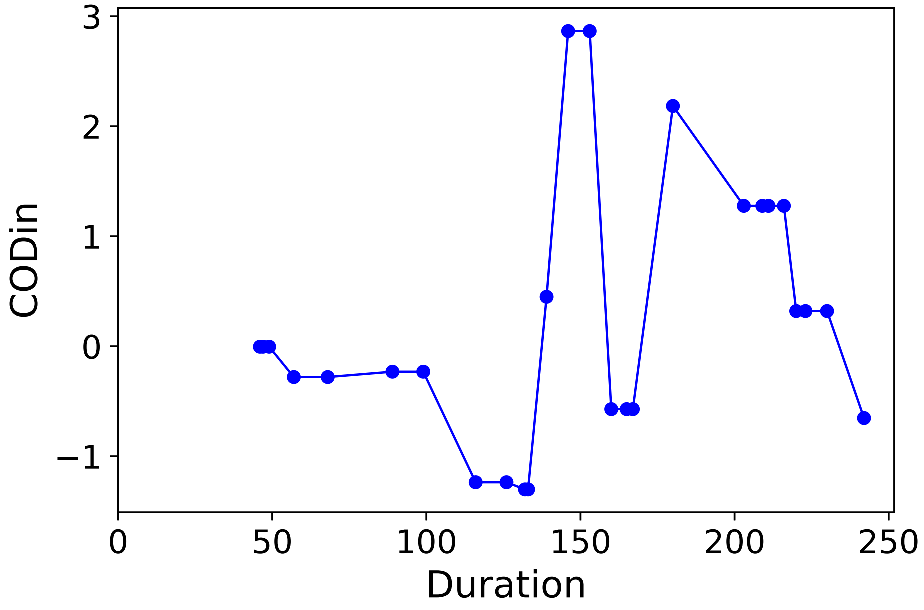


Figure A11. Normalized chemical oxygen demand for **Reactor 1**.

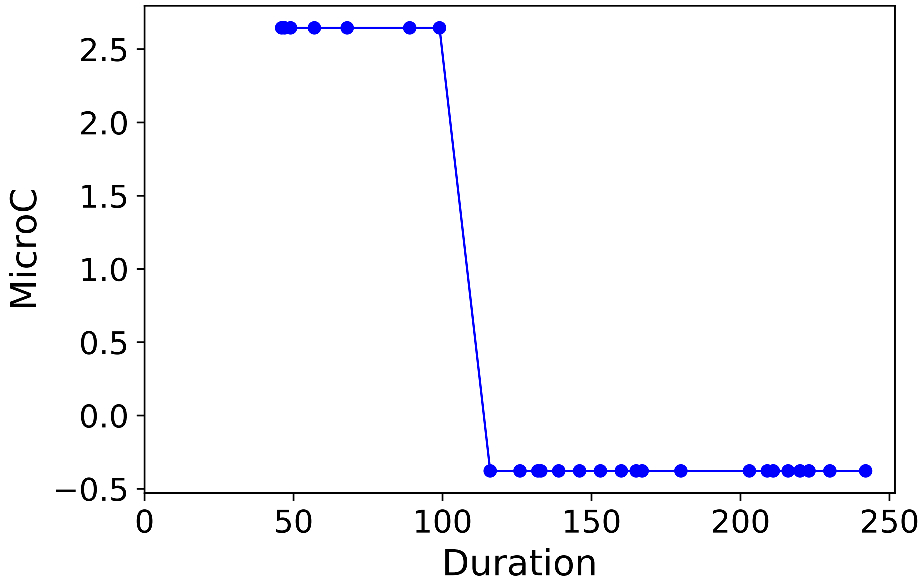


Figure A12. Normalized categorical carbon source *(MicroC)* for **Reactor 1**.

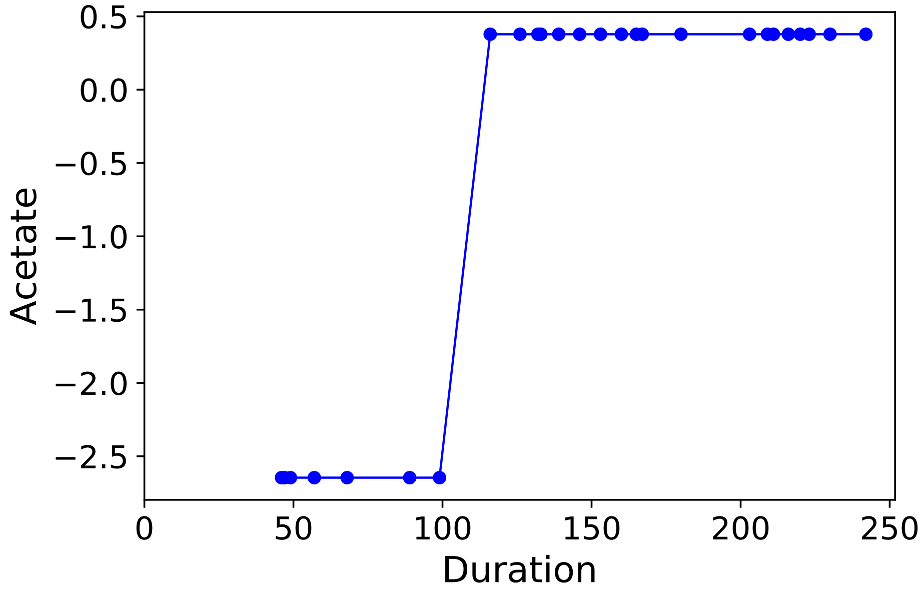


Figure A13. Normalized categorical carbon source *(Acetate)* for **Reactor 1**.

878 H.2 Time-plots from Reactor 2

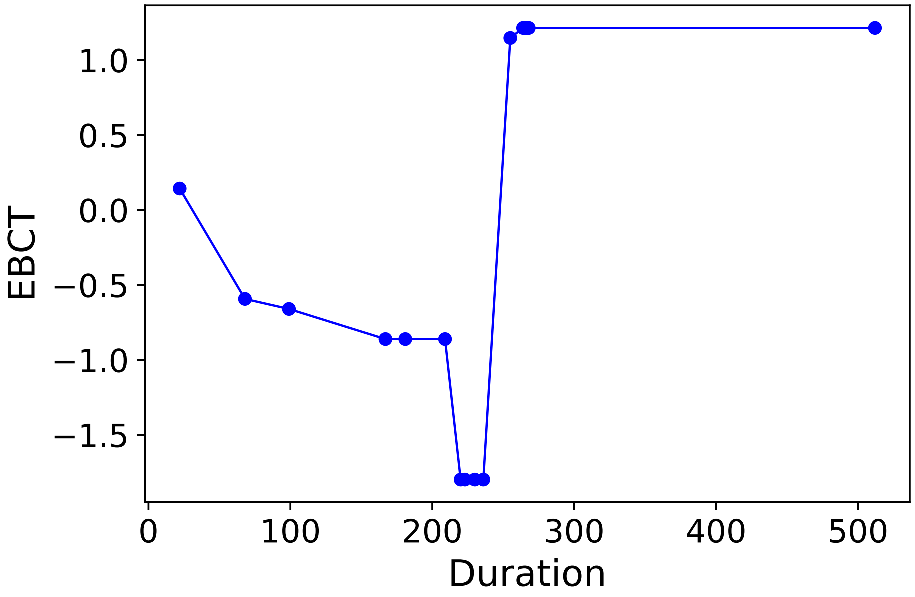


Figure A14. Normalized empty-bed contact time for **Reactor 2**.

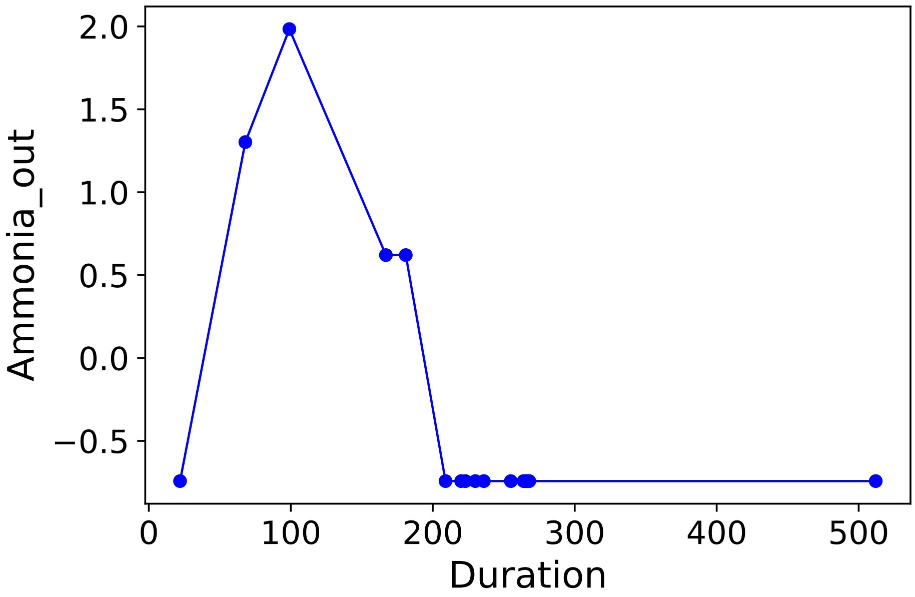


Figure A15. Normalized ammonia outlet flowrate for **Reactor 2**.

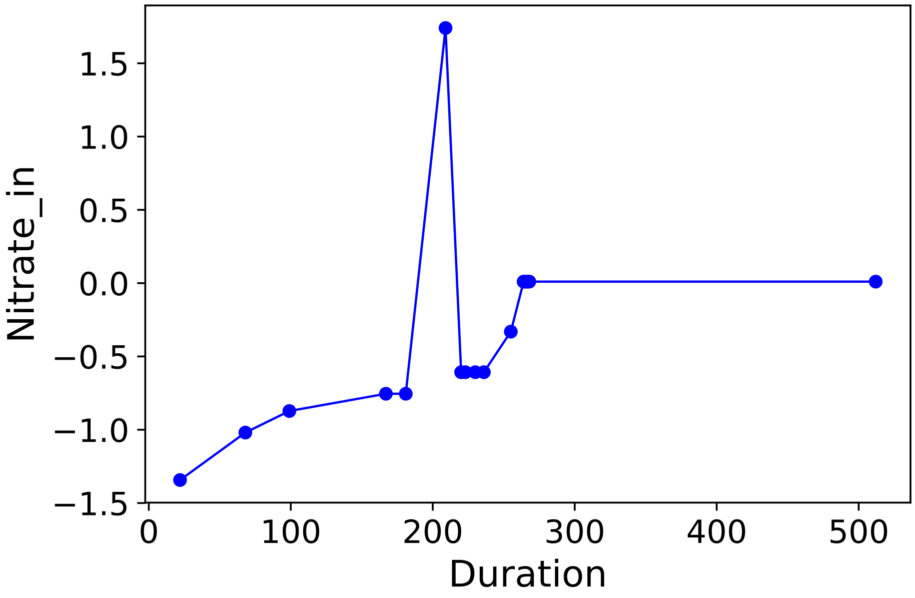


Figure A16. Normalized nitrate inlet flowrate for **Reactor 2**.

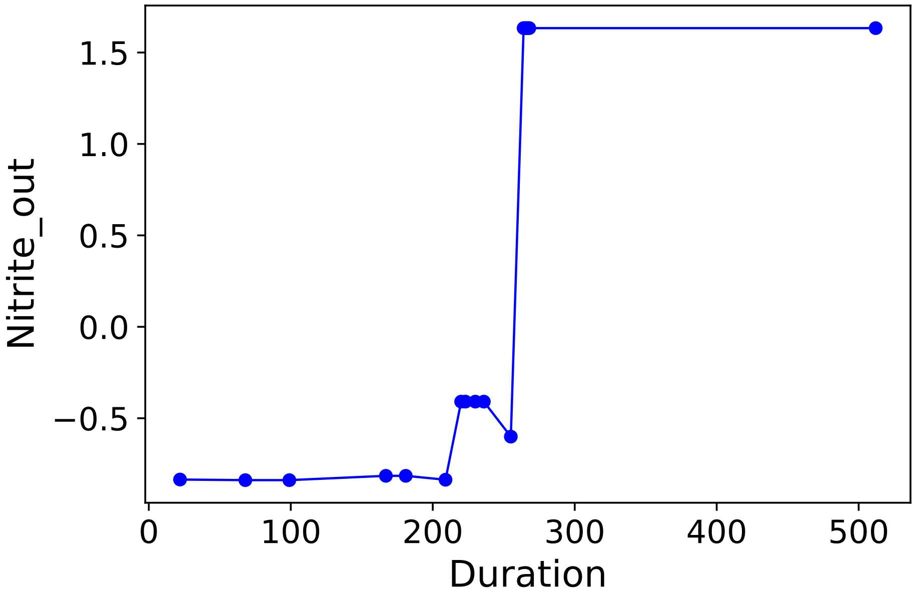


Figure A17. Normalized nitrite outlet flowrate for **Reactor 2**.

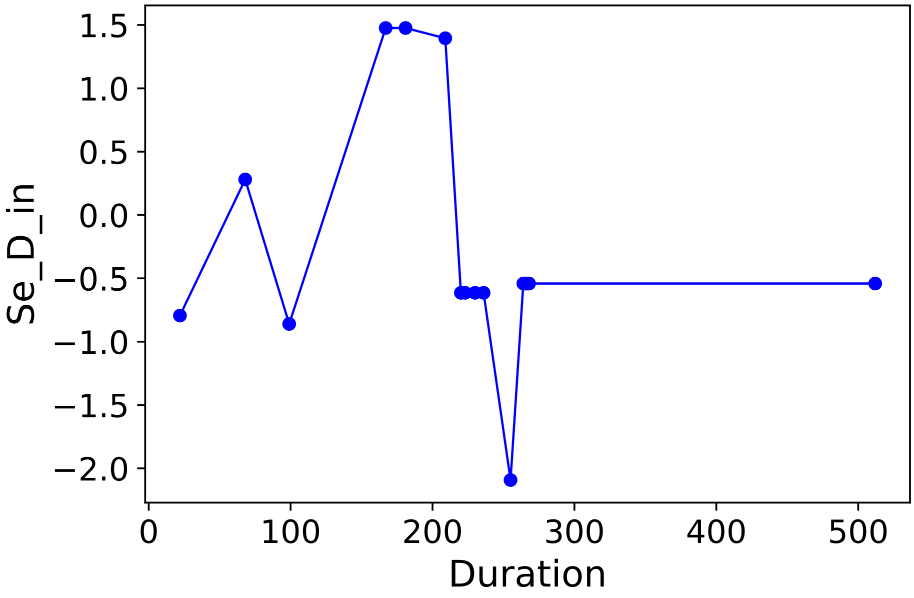


Figure A18. Normalized selenium inlet flowrate for **Reactor 2**.

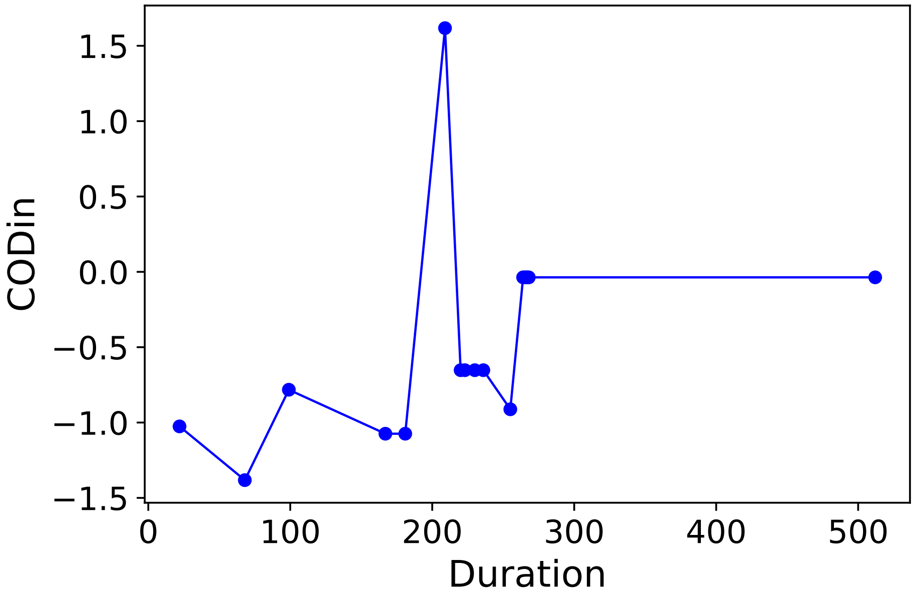


Figure A19. Normalized chemical oxygen demand for **Reactor 2**.

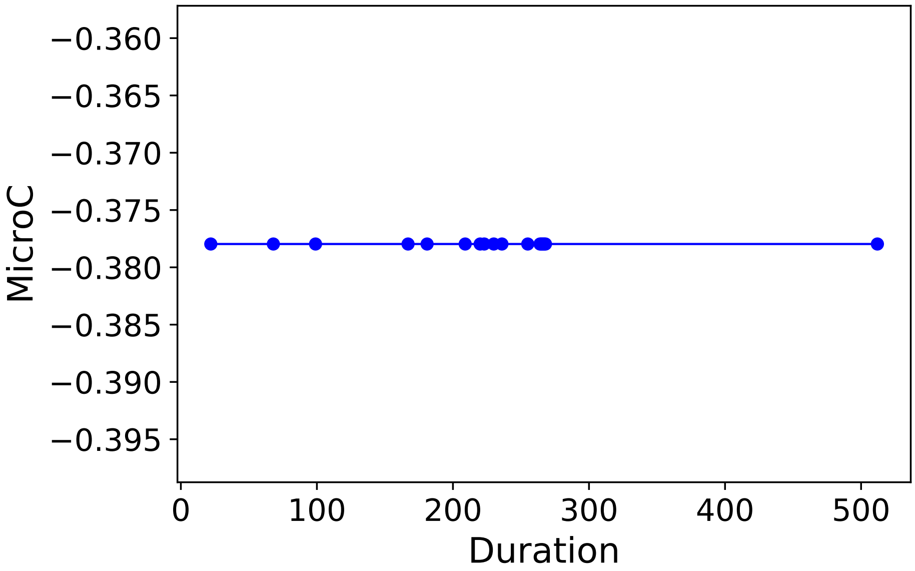


Figure A20. Normalized categorical carbon source *(MicroC)* for **Reactor 2**.

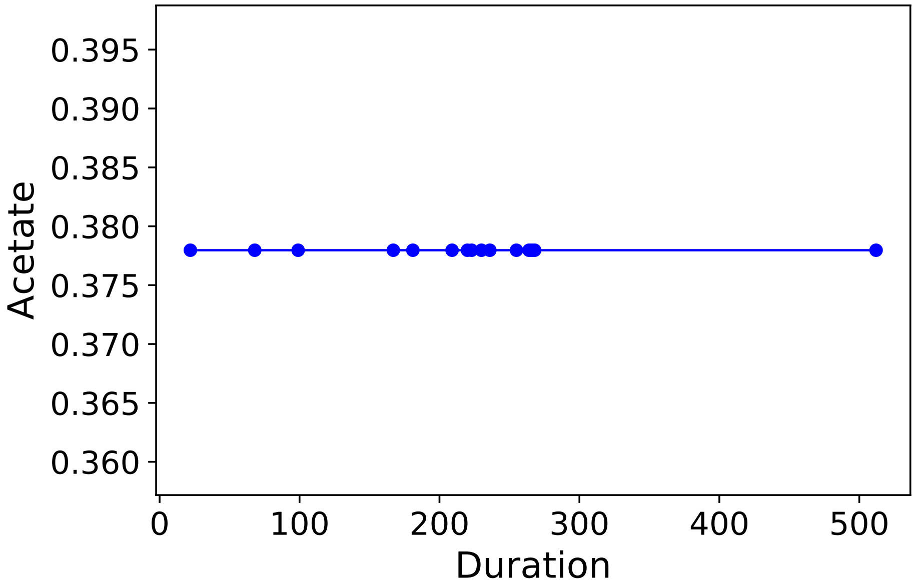


Figure A21. Normalized categorical carbon source *(Acetate)* for **Reactor 2**.

879 I Feature selection results from C-MDA

880 The following figures show the feature importances obtained using the Conditional Permutation 881 algorithm developed by [33]. The plots are separated for the 3 cases of hierarchical, Gaussian, and 882 Dirichlet OTU-clusters.

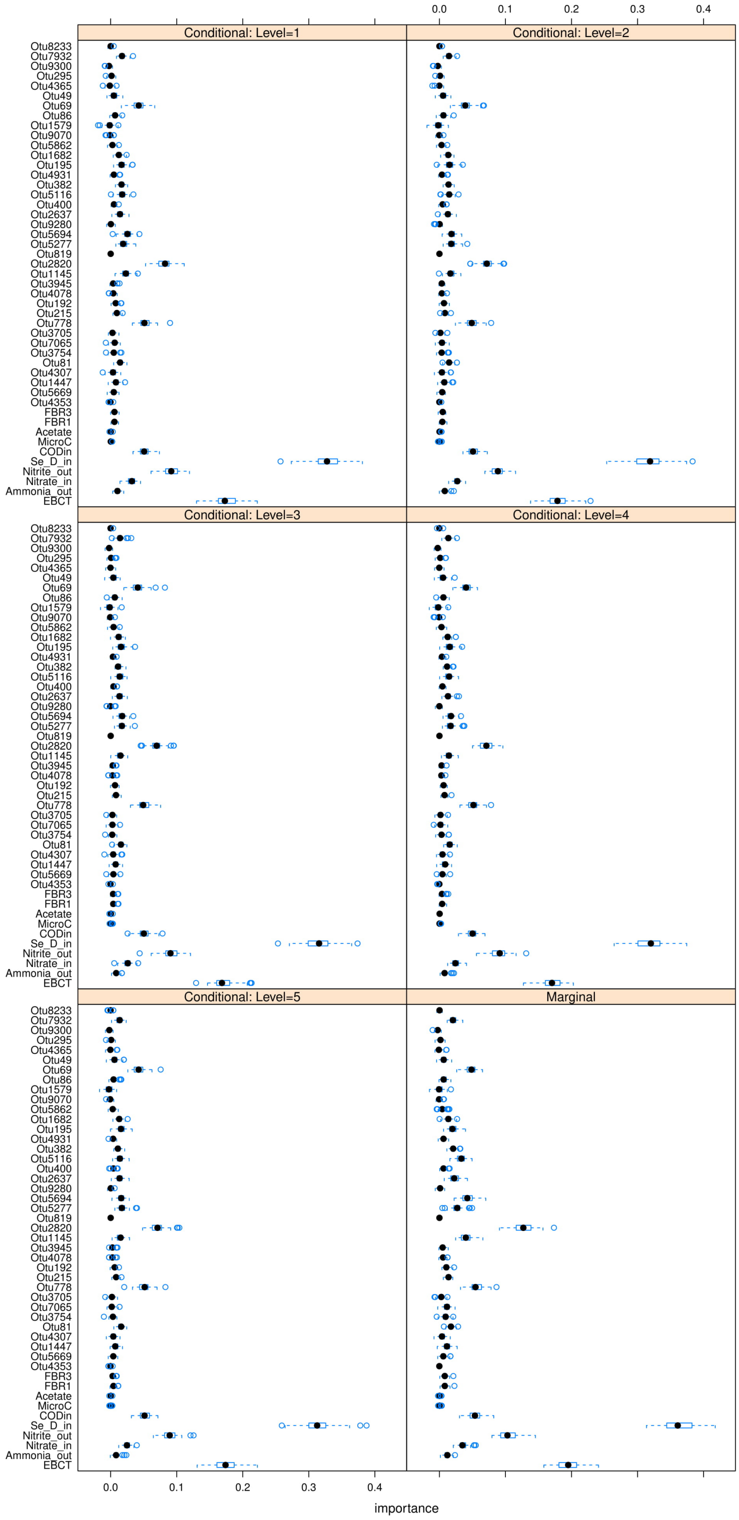


Figure A22. Conditional feature importances for hierarchical clustering.

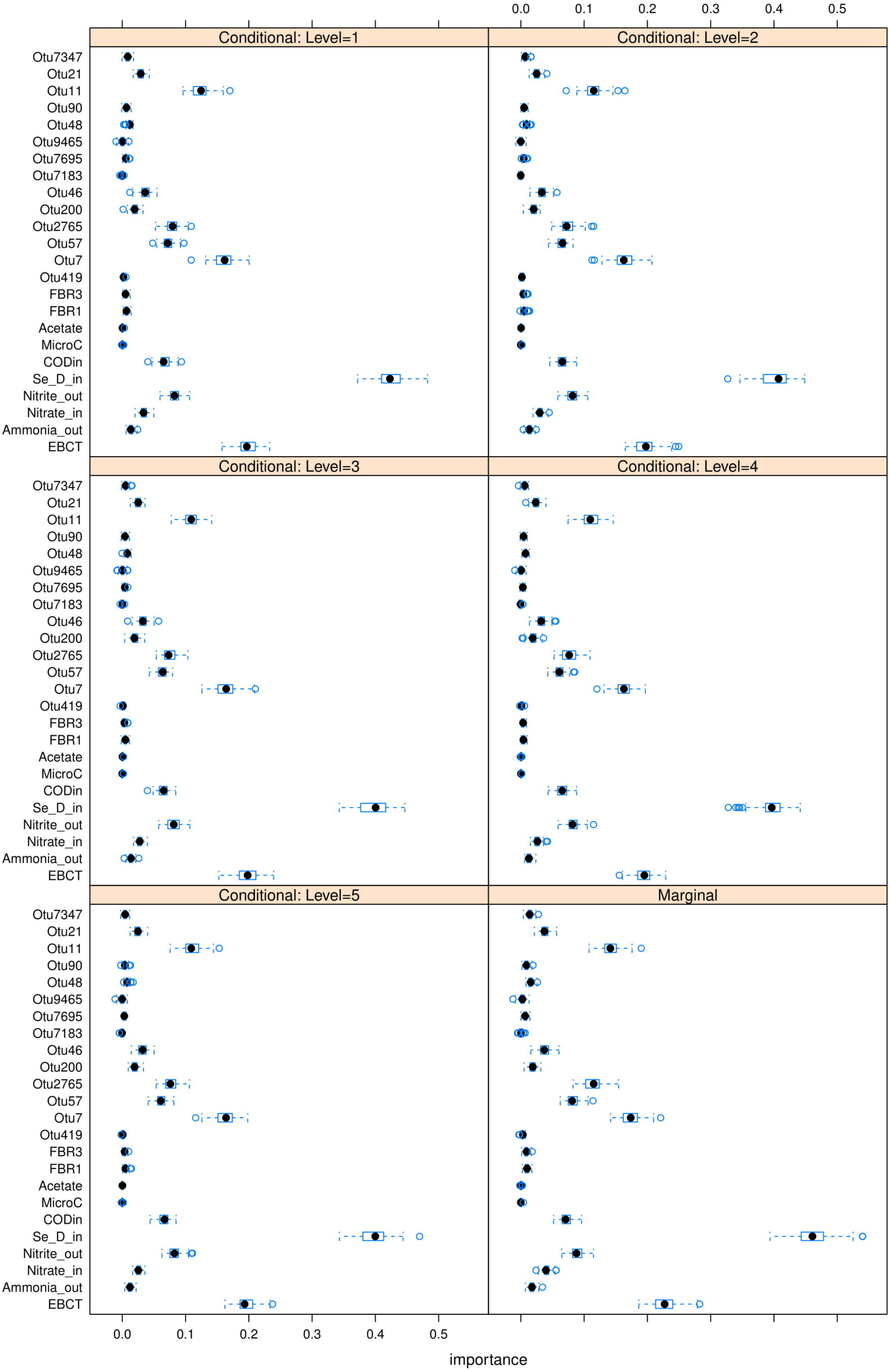


Figure A23. Conditional feature importances for GMM clustering.

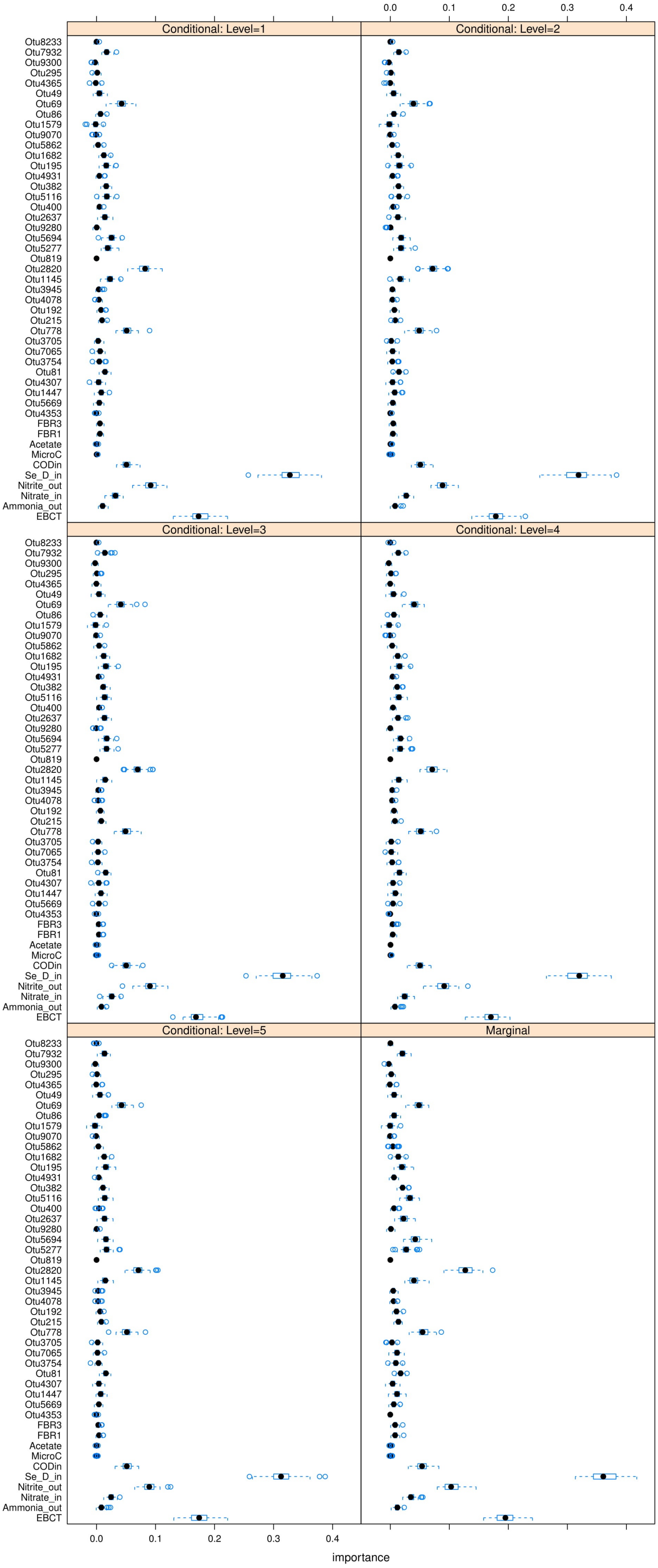


Figure A24. Conditional feature importances for DMM clustering.

1. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. nature **1986**, 323, 533.
2. Pearl, J. Probabilistic reasoning in intelligent systems: networks of plausible inference; Elsevier, 2014.
3. Kindermann, R. Markov random fields and their applications. American mathematical society **1980**.
4. Geman, S.; Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. In Readings in computer vision; Elsevier, 1987; pp. 564–584.
5. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; others. Tensorflow: a system for large-scale machine learning. OSDI, 2016, Vol. 16, pp. 265–283.
6. Seborg, D.E.; Mellichamp, D.A.; Edgar, T.F.; Doyle III, F.J. Process dynamics and control; John Wiley & Sons, 2010.
7. Murphy, K.P. Machine Learning: A Probabilistic Perspective; Vol. 1, MIT Press, 2012.
8. Bishop, C.M. Pattern Recognition and Machine Learning; Vol. 1, Springer, 2006.
9. Goodfellow, I.; Bengio, Y.; Courville, A. Deep learning; MIT press, 2016.
10. Ljung, L. System identification: theory for the user; Prentice-hall, 1987.
11. Faust, K.; Raes, J. Microbial interactions: from networks to models. Nature Reviews Microbiology **2012**, 10, 538.
12. Gonze, D.; Lahti, L.; Raes, J.; Faust, K. Multi-stability and the origin of microbial community types. The ISME journal **2017**, 11, 2159.
13. Lesnik, K.L.; Liu, H. Predicting Microbial Fuel Cell Biofilm Communities and Bioreactor Performance using Artificial Neural Networks. Environmental science & technology **2017**, 51, 10881–10892.
14. Han, H.G.; Zhang, L.; Liu, H.X.; Qiao, J.F. Multiobjective design of fuzzy neural network controller for wastewater treatment process. Applied Soft Computing **2018**, 67, 467–478.
15. Han, H.g.; Zhang, L.; Qiao, J.f. Data-based predictive control for wastewater treatment process. IEEE Access **2017**, 6, 1498–1512.
16. Qiao, J.F.; Hou, Y.; Zhang, L.; Han, H.G. Adaptive fuzzy neural network control of wastewater treatment process with multiobjective operation. Neurocomputing **2018**, 275, 383–393.
17. Li, H. Microbiome, metagenomics, and high-dimensional compositional data analysis. Annual Review of Statistics and Its Application **2015**, 2, 73–94.
18. La Rosa, P.S.; Brooks, J.P.; Deych, E.; Boone, E.L.; Edwards, D.J.; Wang, Q.; Sodergren, E.; Weinstock, G.; Shannon, W.D. Hypothesis testing and power calculations for taxonomic-based human microbiome data. PloS one **2012**, 7, e52078.
19. Holmes, I.; Harris, K.; Quince, C. Dirichlet multinomial mixtures: generative models for microbial metagenomics. PloS one **2012**, 7, e30126.
20. Matsuda, H.; Ogita, N.; Sasaki, A.; Satō, K. Statistical mechanics of population: the lattice Lotka-Volterra model. Progress of theoretical Physics **1992**, 88, 1035–1049.
21. Yasuhiro, T. Global dynamical properties of Lotka-Volterra systems; World Scientific, 1996.
22. Morueta-Holme, N.; Blonder, B.; Sandel, B.; McGill, B.J.; Peet, R.K.; Ott, J.E.; Violle, C.; Enquist, B.J.; Jørgensen, P.M.; Svenning, J.C. A network approach for inferring species associations from co-occurrence data. Ecography **2016**, 39, 1139–1150.
23. Breiman, L. Bagging predictors. Machine learning **1996**, 24, 123–140.
24. Cortes, C.; Vapnik, V. Support-vector networks. Machine learning **1995**, 20, 273–297.
25. Cybenko, G. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems **1989**, 2, 303–314.
26. Wang, D.; Li, M. Stochastic configuration networks: Fundamentals and algorithms. IEEE transactions on cybernetics **2017**, 47, 3466–3479.
27. Chen, W.; Zhang, C.K.; Cheng, Y.; Zhang, S.; Zhao, H. A comparison of methods for clustering 16S rRNA sequences into OTUs. PloS one **2013**, 8, e70837.
28. Kaufman, L.; Rousseeuw, P.J. Finding groups in data: an introduction to cluster analysis; Vol. 344, John Wiley & Sons, 2009.
29. Sokal, R.R.; Rohlf, F.J. The comparison of dendrograms by objective methods. Taxon **1962**, pp. 33–40.

30. Rousseeuw, P.J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **1987**, 20, 53–65.

31. Han, H.; Guo, X.; Yu, H. Variable selection using mean decrease accuracy and mean decrease gini based on random forest. Software Engineering and Service Science (ICSESS), 2016 7th IEEE International Conference on. IEEE, 2016, pp. 219–224.

32. Saeys, Y.; Inza, I.; Larrañaga, P. A review of feature selection techniques in bioinformatics. bioinformatics **2007**, 23, 2507–2517.

33. Strobl, C.; Boulesteix, A.L.; Kneib, T.; Augustin, T.; Zeileis, A. Conditional variable importance for random forests. BMC bioinformatics **2008**, 9, 307.

34. CCME. Canadian Water Quality Guidelines for the Protection of Aquatic Life: NITRATE ION, 2012.

35. CCME. Soil Quality Guidelines: SELENIUM Environmental and Human Health Effects, 2009.

36. Lemly, A.D. Aquatic selenium pollution is a global environmental safety issue. Ecotoxicology and environmental safety **2004**, 59, 44–56.

37. Sanderson, S.C.; Ott, J.E.; McArthur, E.D.; Harper, K.T. RCLUS, a new program for clustering associated species: a demonstration using a Mojave Desert plant community dataset. Western North American Naturalist **2006**, pp. 285–297.

38. Morgan, M. Dirichlet multinomial: Dirichlet-multinomial mixture model machine learning for microbiome data. R package. R Foundation for Statistical Computing, Vienna, Austria **2014**.

39. Runge, J.; Nowack, P.; Kretschmer, M.; Flaxman, S.; Sejdinovic, D. Detecting causal associations in large nonlinear time series datasets. arXiv preprint arXiv:1702.07007 **2017**.

40. Maciejowski, J.M. Predictive control: with constraints; Pearson education, 2002.

41. Sutton, R.S.; Barto, A.G.; others. Introduction to reinforcement learning; Vol. 135, MIT press Cambridge, 1998.

42. Breiman, L. Classification and regression trees; Routledge, 2017.

43. Breiman, L. Random forests. Machine learning **2001**, 45, 5–32.

44. Strobl, C.; Boulesteix, A.L.; Zeileis, A.; Hothorn, T. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC bioinformatics **2007**, 8, 25.

45. Vapnik, V.N.; Vapnik, V. Statistical learning theory; Vol. 1, Wiley New York, 1998.

46. Lemm, S.; Blankertz, B.; Dickhaus, T.; Müller, K.R. Introduction to machine learning for brain imaging. Neuroimage **2011**, 56, 387–399.

47. Legendre, P.; Legendre, L. Numerical Ecology, Volume 24, (Developments in Environmental Modelling); Elsevier Science Amsterdam, The Netherlands, 1998.