ACTIVibrio: Optimal Stopping Criteria for Determination of *Vibrio cholerae* Biofilm Phenotypes in Active Learning

Darin Boyes, Shivank Sadasivan, Jonathan Zhu
Ray and Stephanie Lane Computational Biology Department
Carnegie Mellon University
Pittsburgh, PA 15213
dboyes, ssadasiv, jazhu@andrew.cmu.edu

1 Introduction

1.1 Biological Significance

Biofilms are communities of bacteria engulfed in a sticky extracellular substance consisting of proteins, polysaccharides, DNA, and RNA. Biofilms confer a significant survival advantage for bacterial species that can form them, as the sticky matrix can prevent invaders, such as bacteriophages, immune cells, or other antibiotics from penetrating the biofilm and killing the entire community. In particular, our group was interested in the biofilm dynamics of *Vibrio cholerae*, a global pathogen that colonizes and forms biofilms in the human gastrointestinal tract during its infection stage. Mutants of *Vibrio cholerae* have a diverse range of phenotypes with respect to biofilm formation and dispersal, so the overall goal is to be able to classify the phenotypes into their corresponding genotype so that the mechanism of action is better understood. This would help in downstream drug development if we want to target a specific mechanism that would inhibit the ability of *Vibrio cholerae* to form biofilms.

Our group was particularly interested in the different regulatory mechanisms and pathways involved in the formation and dispersal in *Vibrio cholerae* and the phenotypes that might result from inhibiting those pathways. From this, we have a set of 8 representative mutants, each involved in inhibiting biofilm formation and dispersal. These mutants are $\Delta flaA$, $\Delta hapR$, $luxO^{D47E}$, $\Delta manA$, $\Delta rbmB$, $\Delta potD1$, $vpvC^{W240R}$, and $\Delta vpsL$. $\Delta flaA$ is a motility mutant that prevents formation of the flagellar filament. $\Delta hapR$ and $luxO^{D47E}$ are both quorum-sensing mutants that are responsible for mediating the switch between a high cell density state and a dispersal state. $\Delta manA$ and $\Delta rbmB$ are both hypothesized to impact the ability of *Vibrio cholarae* to attach to components within the biofilm matrix. $\Delta potD1$ and $vpvC^{W240R}$ are both mutants that impact the detection of signaling molecules required to regulate the switch between biofilm and planktonic or free-swimming. $\Delta vpsL$ is a mutant that prevents the synthesis of a key component that makes up the biofilm matrix. This is just a subset of possible pathways that impact biofilm formation and dispersal. We want to be able to determine if the phenotypes that are readily observable using brightfield microscopy belong to any of these existing genotypes. If these phenotypes can be easily classified, then we would have a better idea of the pathways for phenotypes with potentially unknown genotypes, such as with transposon mutagenesis data or with compound screening data.

1.2 Dataset

Our dataset contains 1296 videos of biofilm growth and dispersal of 8 mutants and a wild type version of the global bacterial pathogen *Vibrio cholerae*. Each of the mutants contain a mutation in a gene that is known to impact some aspect of the biofilm life cycle, and the goal is to train a model that can distinguish each *Vibrio cholerae* genotype based on their corresponding biofilm phenotype. Thus, this is a multiclass classification problem with each of the classes being one of the 9 genotypes.

1.2.1 Data Acquisition

The timeline for one round of data acquisition is as follows. We first start by growing the bacteria, which takes about 1 day before they reach log phase. The imaging plates are then prepared using an automated liquid handler with minimal media. The number of plates that are prepared depends on the number of slots available in the automated incubator and microscope. Generally, only 3 out of the 8 slots are available at a time due to it being a shared lab automation setup. The images for the 3 plates are then acquired, and features are extracted later in the week.

Because there are 1296 samples in total, each of the 9 biological replicates has 144 wells. 144 wells is equivalent to 1.5 plates, so only being able to do 3 plates per week means that we can only image 2 biological replicates per week. Therefore, imaging 9 biological replicates would take about 4.5 weeks in total.

This highlights the main issue with acquiring high-content video datasets like this, which is that they can take a long time, especially when accounting for plate preparation time and waiting for bacteria to grow. Therefore, the motivation for working with this dataset was to find out if 1296 videos are necessary to achieve good genotype classification results.

1.2.2 Data Preprocessing

After all the necessary video data is acquired, it still needs to be preprocessed before it is suitable for downstream analysis. The main image preprocessing steps that we perform are Gaussian normalization to remove noise and subpixel registration to resolve the jitteriness of each video. However, before we can begin exploring how active learning algorithms under various stopping criteria can reduce the number of experiments that we need to perform, we first need to have some way to convert our set of videos into a set of features that is more manageable for a traditional machine learning algorithm like SVM or random forest. To do this, we took each frame of our Gaussian normalized and subpixel registered videos and passed them through the layers of a Vision Transformer neural network, as described in Dosovitskiy et. al [4]. Then we extracted the last layer of the final transformer block and added a separate linear projection head with ReLU activation function to project the embedding space down to 512 dimensions. Now, each video is represented as a 25x512 matrix, with each video containing 25 frames, and each frame being represented as a 512 dimension learned representation / embedding. Still, this is a lot of features to work with which required us to reduce the dimensionality even further. The issue now is that we are working with video data, and video data always contains important temporal information that can be extracted. Biofilms at any point in their life cycle never occur independently of other stages of their life cycle. Thus, in order to account for temporal information, we incorporate a fixed positional encoding scheme which follows a sinusoidal pattern, which was described in Vaswani et. al [10]:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{dim}}}\right)$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{dim}}}\right)$$

This sinusoidal positional encoding scheme allows for long dependencies to be modeled in the video and is thus a really good option for incorporating positional information into the embeddings.

After extracting embeddings and applying fixed weights to model temporal information, we have a dataset of 1296 videos, each of which is represented by a 512 dimension embedding. We plan on using this dataset of learned representations for downstream classification tasks and experimentation with various stopping criteria.

1.3 Stopping Criteria

To investigate whether or not 1296 videos are necessary for genotype classification, we turn to stopping criteria. Stopping criteria, also known as stopping rules, are conditions to meet in order to determine when to stop labeling data. More mathematically speaking, a *valid* stopping criterion is defined as a random variable τ whose values consist of the items in the union $\mathbb{N} \cup \{+\infty\}$ such that $\{\tau = n\} \in \mathcal{F}_n \forall n$ with respect to the filtration $\mathbb{F} = ((\mathcal{F}_n)_{n\in\mathbb{N})}$ [5]. In other words, a mathematically

valid stopping criterion must meet two requirements: *does the criterion only depend on information of the past and present,* and *is this stopping criterion guaranteed to be reached in a finite time?*

Pertinent examples of valid stopping criteria in an active learning context include stopping after collecting 200 labeled data points (which is valid assuming the available data is sufficiently large), or stopping when a set budget is exhausted (which is not the same as stopping after n data points in the case of a variable-cost oracle). Intuitively, one might also stop when classification loss falls below a certain threshold or when classifier accuracy exceeds a certain threshold; however, these are not mathematically valid as there is often no guarantee that a classifier can reach a set accuracy or loss.

Despite this, stopping criteria that monitor loss or accuracy can often be *good* despite being mathematically invalid. When investigating good stopping criteria, our primary concerns are performance (whether utilizing this stopping criteria yield similar metrics to offline learning) and if the criterion is reached in practice (regardless of formal guarantees). In fact, the main stopping criteria for active learning heuristics primarily aim to be good rather than mathematically valid. Uncertainty sampling has the most stopping criteria designed for its case [12], with key examples being confidence-based stopping that stops collecting labeled data when uncertainty peaks [11] and gradient-based stopping, which stops collecting labeled data when the difference in median uncertainty between iterations is close to 0 [6]. A stopping criterion has also been developed for query by committee (QBC), stopping when the difference in committee disagreement between iterations is close to 0 [8]. Critically, none of these stopping criteria have formal guarantees, and none fit the requirements to be mathematically valid.

To our knowledge, no stopping criteria have been developed specifically for more modern query selection algorithms, such as Type I and Type II algorithms, or when using a deep learning model as the base learner. As such, we sought to investigate *good* stopping criteria not only for high-performance heuristics, but also for deep learning and modern algorithms with formal guarantees.

2 Methods

2.1 Query Selection Methods and Relevant Stopping Criteria

Since optimal stopping criteria may vary depending on the query selection method used, we designed and tested stopping conditions with a variety of query selection methods. Our chosen methods are briefly described in the following subsections.

2.1.1 Uncertainty Sampling (Random Forest and SGD)

Uncertainty sampling is an active learning strategy where new samples are selected based on how uncertain the current model is regarding their classifications. Labeling uncertain samples can significantly improve the accuracy of the model. We applied uncertainty sampling using Random Forest and Stochastic Gradient Descent (SGD) as base models.

Random Forest consists of multiple decision trees whose collective predictions determine the final class. We measured uncertainty by calculating the predicted probabilities from all trees. The samples selected next for labeling are those where the model has the lowest maximum predicted probability (least confident predictions).

Compared to Random Forest, SGD is a quick and iterative method commonly used for training linear classifiers. In SGD, we similarly select samples where the model is least confident (lowest maximum predicted probability). Additionally, we monitored entropy, the uncertainty across predicted classes, to ensure selected samples provided genuinely new information.

We established clear stopping conditions based on these uncertainty metrics. For Random Forest, labeling stops when uncertainty (1 – max probability) remains consistently low (below threshold = 0.1) for multiple consecutive rounds (m = 5), indicating the model is sufficiently confident. For SGD, labeling stops when uncertainty either stabilizes (less than 0.002 change across 5 rounds) or becomes consistently low (below 0.1), or when new labeled samples no longer add significant information (entropy difference < 0.05 compared to the unlabeled pool).

Additionally, we applied a global stopping criterion for both methods, stopping labeling if cross-validation accuracy improvement plateaus (less than 0.002 improvement for 5 rounds).

2.1.2 Query By Committee

Another approach to selecting informative sample points to introduce during training is called Query by Committee, which was first detailed in Seung et. al [7]. The algorithm begins by constructing a committee, training each member of the committee on the currently labeled dataset, and then evaluating the output class probabilities from each commitee member. There are various methods for evaluating the disagreement within a committee. Some metrics have methods for handling the confidence of individual committee members, such as soft vote entropy, or entropy based on Kullback-Leibler divergence, whereas others rely primarily on the final predicted class as a measure of entropy. More entropic committee predictions indicate greater disagreement among the members of the committee.

To this end, we have developed a stopping criterion that is based primarily around committee disagreement using Kullback-Leibler divergence. To reiterate what Kullback-Leibler divergence is and how it describes the relative entropy between two probability distributions, it might help to look at the formula for KL divergence below:

$$D_{KL}(P \mid\mid Q) = \int P(x) \cdot log\left(\frac{P(x)}{Q(x)}\right)$$

P describes the null distribution, while Q describes the alternate distribution. Higher KL divergence values would indicate that P is more distinct from Q across the entire probability density function because the log difference between P and Q contributes to the overall KL divergence.

Then, the conversion of this metric into a stopping criterion is straightforward. From the predicted class probabilities of each member in the committee at a particular training iteration, we can calculate how much the class probabilities for an individual committee member diverge from the class probabilities of the entire committee. If the average divergence across each committee member is below a certain threshold for a specified number of iterations, training stops. In other words, training stops when the committee begins to mostly agree on which class each sample in the unlabeled dataset belongs to.

2.1.3 Importance-Weighted Active Learning (IWAL)

One of a family of Type I algorithms that utilize query selection to eliminate hypotheses [2], Importance-Weighted Active Learning (IWAL) first described by Beygelzimer et al. [1] normally operates on stream-based sampling but can easily be adapted to pool-based sampling by randomly ordering the items in the pool. For every single data point that arrives, IWAL computes a p_t according to the predictions of a model committee; this p_t is then the probability of accepting the data point and getting its true label. Furthermore, this p_t is used to create an associated weight for the point, which is used to calculate importance-weighted loss; IWAL seeks to build a classifier that minimizes this importance-weighted loss. Unlike other known Type I algorithms, IWAL is adaptible to multi-class classifiers and has theoretical guarantees, which is why we chose to experiment with it.

To our knowledge, no stopping criteria exists for IWAL, so the one we experimented with requires three conditions to be met before stopping. The user defines some b which determines the number of data points to accept in order to make the model committee; the first condition must be that we have accepted all of these first b data points. Secondly, we must have accepted two points not within this initial set. Thirdly, the user specifies some $\epsilon > 0$ and IWAL stops when the difference in cross-validation importance-weighted loss between iterations where a point was accepted is less than ϵ (i.e., $\ell_{w,t}(x_L, y_L, h_t) - \ell_{w,t-1}(x_L, y_L, h_{t-1}) < \epsilon$).

2.1.4 DH and PLAL Algorithms

In contrast to Type I algorithms, Type II algorithms (including DH [3] and PLAL [9]) seek to exploit natural clusters within the data [2]. In loose terms, DH performs a hierarchical clustering on a pool of unlabeled data, then recursively samples from subtrees within this clustering, computing error probabilities based on the current labeled dataset and the newly sampled points' true labels; if this error is sufficiently low, the algorithm automatically labels all points in the subtree of the majority label (that is, the majority label within the sampled points).

PLAL, on the other hand, constructs a space-partitioning tree on the data which exploits the natural spatial clusters that are present. It iterates through levels of the space partitioning tree, making one

node active at a time until there are no longer any active nodes. A node becomes active if the first q data points queried in the subtree of that node do not share the same label. q can be defined as follows:

$$q = \frac{level \cdot 2 \cdot log(2) + log(\frac{1}{\sigma})}{\epsilon}$$

If the first q data points share the same label, then we can infer that the labels of the entire subtree rooted at that node are the shared label. Otherwise, we continue iterating through the currently active cells. The result is a smaller set of queried and inferred labels that are representative of the distribution of the original dataset.

From this, we developed two stopping criteria that operate on both algorithms. Firstly, we developed depth-based stopping; the user specifies a specific depth, and sampling stops when the algorithm samples from a subtree rooted at a note below that depth. Secondly, we developed what we call index-based stopping. This involves, at every iteration, k-means clustering the labeled data pool with $k =$ the number of classes, calculating the Jaccard index of each cluster with respect to its majority label, and multiplying the average index by the number of points sampled n_L. The algorithm then stops when this metric exceeds a user-specified threshold. The rationale behind this metric is that when this metric is low, we have not sampled enough data and also have not explored enough of the data's inherent clusters and structure. However, when this metric is high, we have done one or both of these things.

2.1.5 Deep Learning (MLP with MC Dropout)

Estimating uncertainty accurately with deep learning models, such as Multi-Layer Perceptrons (MLP), is challenging. Standard least confidence using softmax probabilities is unreliable due to overly confident predictions. To address this, we employed Monte Carlo (MC) Dropout. MC Dropout keeps dropout active during inference, generating multiple predictions per sample; variation in these predictions reflects true model uncertainty.

We calculated average predicted probabilities over multiple runs for each sample and computed the entropy of these averaged predictions. Samples with higher entropy (greater prediction disagreement) were selected for labeling because they represented greater uncertainty. Labeling stopped when the entropy of newly selected samples consistently fell below a threshold (entropy threshold = 0.05) for multiple consecutive rounds (m = 5). Low entropy indicated the model had become confident, suggesting additional labeling was unnecessary.

We also applied the global stopping criterion based on plateaued cross-validation accuracy (less than 0.002 improvement for 5 rounds), ensuring efficiency without sacrificing accuracy.

2.2 Evaluation Procedures

All classification models are evaluated with k-fold cross-validation, where the existing labeled data is split into a partition of k subsets, the model is trained on all but one, and the model is then tested on the remaining subset. For each query selection algorithm, we ran 10 simulations and tracked accuracy and classifier loss when relevant to the chosen algorithm. Each query selection algorithm was compared to passive learning (random sampling); since each simulation of active learning selected a different number of data points, we selected an equal number of data points in passive learning for ease of comparison.

3 Results

3.1 Stopping Criteria Evaluations

3.1.1 Uncertainty Sampling

In our initial analyses, we started labeling 20% of the dataset and progressively selected additional samples for labeling, stopping once we reached a maximum labeling budget of 50%. This initial exploration allowed us to determine baseline performance, informing subsequent refinement of uncertainty-based stopping criteria. The goal was to see if fewer samples could achieve comparable accuracy, guiding us towards more efficient experimental designs.

The Random Forest struggled to achieve our accuracy threshold of 97%, failing to reach it even after labeling the maximum allowed samples (50% of the data, around 530 samples). It showed steady improvement over time, but progress slowed significantly toward the end. SGD achieved the 97% accuracy target after labeling approximately 352 samples. It showed a quicker early improvement compared to Random Forest and maintained stable performance afterward.

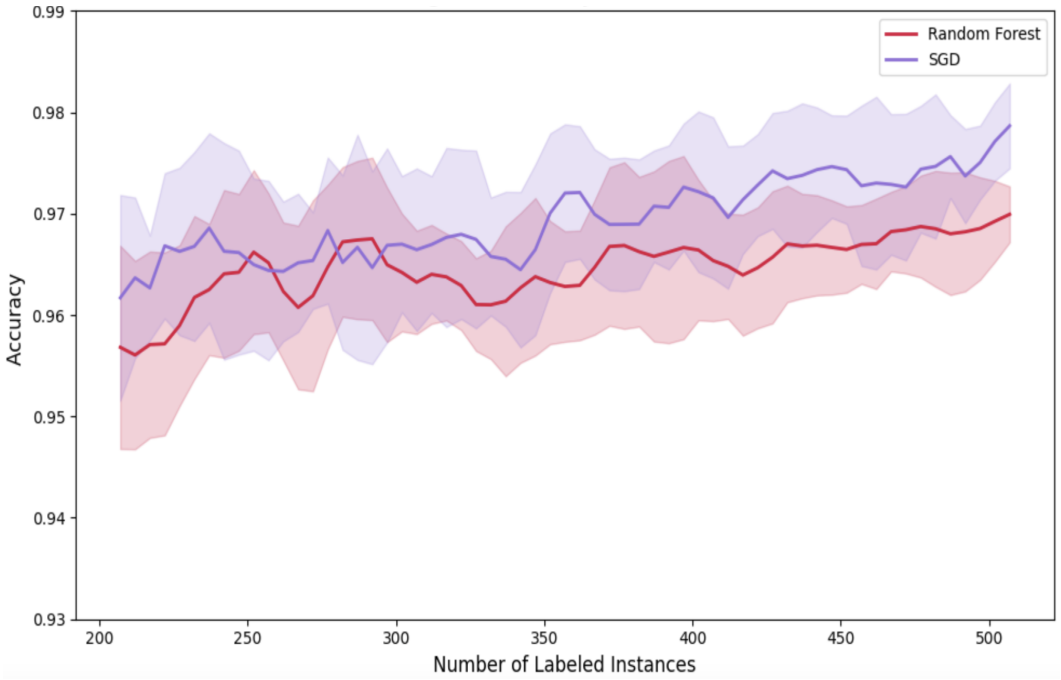

Figure 1: Random Forest and SGD performance and stopping criteria based on 50% dataset (labeling budget)

For the uncertainty collapse stopping for random forest, predictions remained uncertain for a prolonged period, indicating that this criterion may require adjustment (e.g., a higher threshold or more patience rounds) for this dataset. (Refer to Figure 7)

The combination of stopping criteria used for SGD was effective, allowing SGD to stop significantly earlier than Random Forest. Although stable and straightforward, Random Forest required extensive labeling. Uncertainty collapse was not efficient for this model/dataset combination. However, SGD effectively utilized uncertainty-based stopping rules, achieving the accuracy target using fewer samples. The method's multi-criterion stopping strategy showed strong practical performance. (Refer to Figure 7)

3.1.2 Query By Committee

Data sampling under query by committee stops when the committee disagreement is sufficiently low. The disagreement within the committee is quantified using KL divergence. If the KL divergence is below a user-defined threshold for a given number of consecutive iterations, then sampling of the data stops. We see that query by committee stops after about 180 samples on average. The threshold for KL divergence that we used was 0.65, and the number of consecutive iterations that we used was 10.

As more data points are introduced from query by committee, the commitee disagreement goes down. It also seems that random sampling stopped at around the same time point as query by committee (Figure 2). This could be a result of the data being highly differentiable, since each iteration of random sampling almost always provides more information about the inherent structure of the data. This also holds true for query by committee. When the information gained from uncertain points versus compared to random points is small, the overall disagreement of the committee over time

differs only slightly between random sampling and query by committee. Thus, this motivates further exploration of query by committee stopping using less differentiable phenotypes.

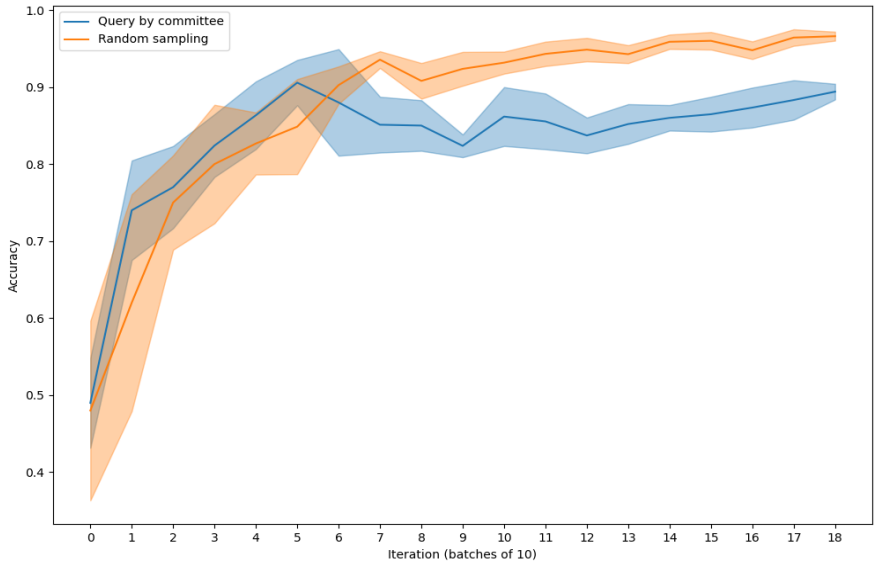

Figure 2: Query by committee cross-validation accuracy compared to random sampling with committee divergence stopping criterion. Translucent portions indicate ± 1 standard deviation across simulations.

3.1.3 Importance-Weighted Active Learning

In utilizing our stopping criteria with IWAL, we used $\epsilon = 0.002$. Due to the random nature of IWAL, the number of points sampled by IWAL varied widely, from 43 to almost 600 in some simulations. Despite this, consistent patterns emerged in IWAL cross-validation accuracy (Figure 3), with it consistently reaching 95% cross-validation accuracy within around 150-200 sampled points, much like passive learning. However, the highly variable nature of these simulations indicated that the IWAL stopping criteria we investigated was not reliable. Furthermore, the accuracy of a model trained on all the data selected by IWAL was not only highly variable but also depended largely on sample size (Figure A1), with some simulations accepting too few samples and others accepting far too many. Therefore, our investigated stopping criteria (while often reached) is often unreliable and in its worst cases can be quite bad.

3.1.4 DH

For our Type II algorithm depth stopping criteria applied to the DH algorithm, we chose a depth threshold of 70. This depth is not reflective of the number of nodes from the root in the tree clustering, but rather the distance of the node from the root in the context of a dendrogram. For our index stopping criteria, we chose an index of 10. A relatively low threshold was used in the context of DH to counteract incredibly low index metrics that arose in experimentation.

The DH algorithm was able to reach approximately 90% cross-validation accuracy, similar to passive learning, for both stopping criteria (Figure 4). While index-based sampling yielded marginally higher cross-validation accuracy, depth-based sampling consistently sampled fewer data points. However, unlike IWAL, datasets chosen by the DH algorithm often had consistent size and did not showcase the highly variable, size-dependent accuracy observed with IWAL, regardless of the stopping criteria used. Therefore, our investigated stopping criteria was not only reached across all simulations, it also turned out to be reliable in that it did not yield highly deviant selected dataset sizes.

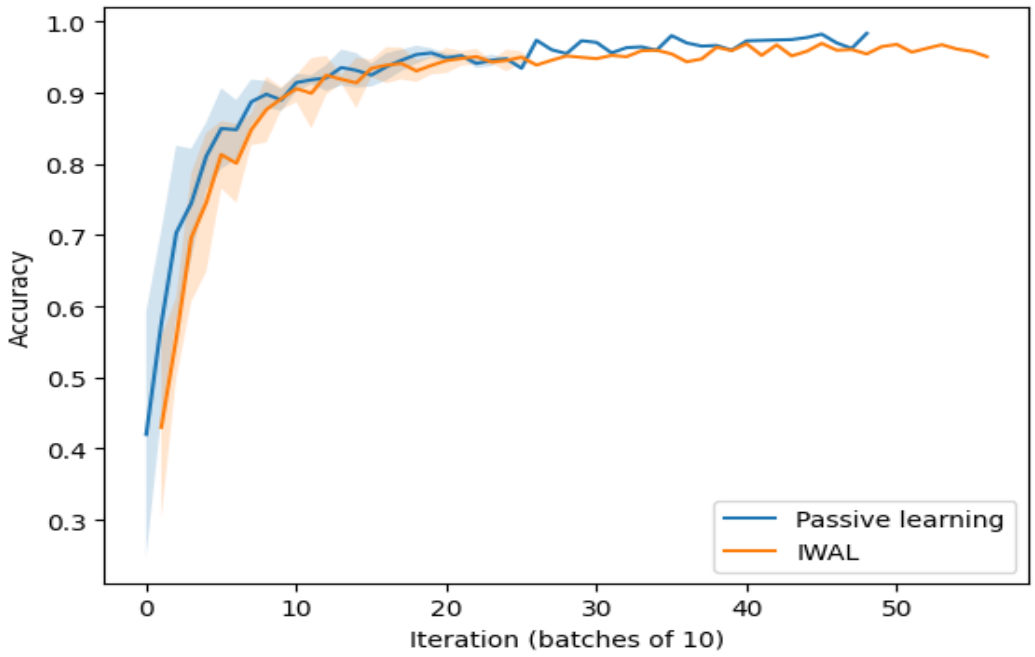

Figure 3: Importance-weighted active learning (IWAL) cross-validation accuracy compared to random sampling with importance-weighted loss stopping criterion. Translucent portions indicate ± 1 standard deviation across simulations. Note that since IWAL is not guaranteed to label a data point at every iteration, its curve runs for more iterations than comparable passive learning.

3.1.5 PLAL

For the other type II algorithm that we had implemented, PLAL, we chose a depth threshold of 9 and a Jaccard index threshold of 110. The reason why the Jaccard index threshold is so high was because the k-means clustering of the currently labeled data set did really well even with a few sampled points (Figure 5). Similarly, only a smaller depth threshold of 9 was required compared to DH to achieve a reasonable stopping point because the data is already highly clusterable. Because of this, the space-partitioning done by PLAL reaches a state where the subtree labels are homogenous very quickly, which would support the result that a smaller depth threshold is sufficient.

In both cases, PLAL can reach about 90% cross-validation accuracy. The Jaccard index threshold approach resulted in less sampling overall, and the point where PLAL stops querying labels is dependent on the threshold set by the user. Even so, we show that with good user-defined thresholds, it is possible for Type II algorithms to stop when the set of queried labels is sufficient to solve the overall problem.

3.1.6 Deep Learning

We first performed a general active learning analysis using a simple budget-based stopping criterion, labeling samples incrementally up to 50% of the available data. Monitoring cross-validation accuracy gave us insight into the accuracy achievable and the labeling needed for various models. Next, we implemented a more targeted uncertainty-aware approach using MC Dropout, specifically designed to quantify uncertainty effectively in deep learning models. The MLP model with MC Dropout quickly achieved our target accuracy (97%) using approximately 242 labeled samples, significantly fewer than the general budget-based approach.(Figure 6)

MC dropout proved robust and practical in capturing uncertainty, allowing us to efficiently identify the optimal stopping point. Unlike softmax-based standard least confidence (MLP Least Confidence), which often provided overly confident predictions and less accurate uncertainty estimates, MC

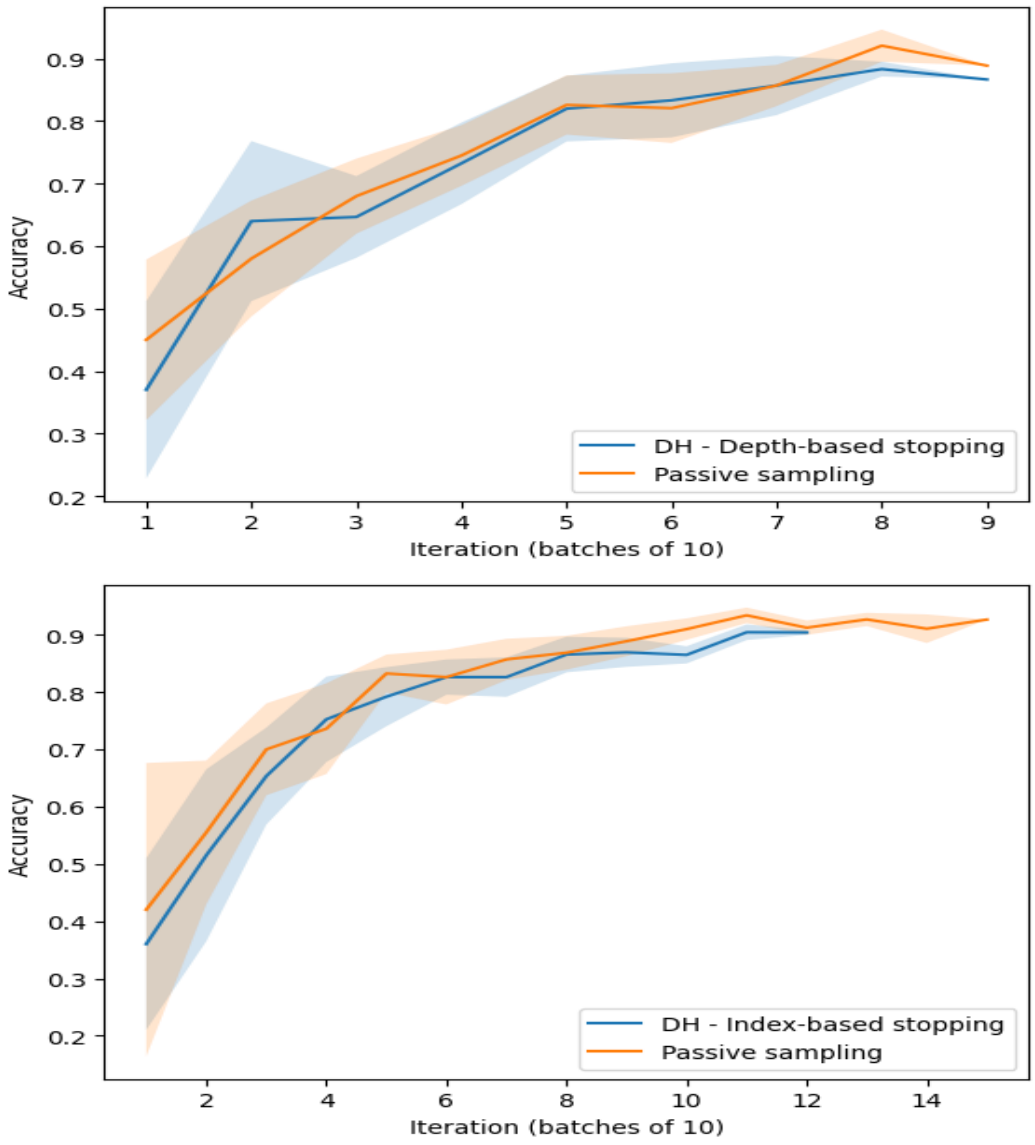

Figure 4: DH algorithm cross-validation accuracy compared to random sampling with depth stopping criterion (top) and index stopping criterion (bottom). Calculations done using a depth threshold of 70 and a Jaccard index threshold of 10. Translucent portions indicate ± 1 standard deviation across simulations. Note that since DH has the capability to label points on its own, its curve runs for fewer iterations than comparable passive learning.

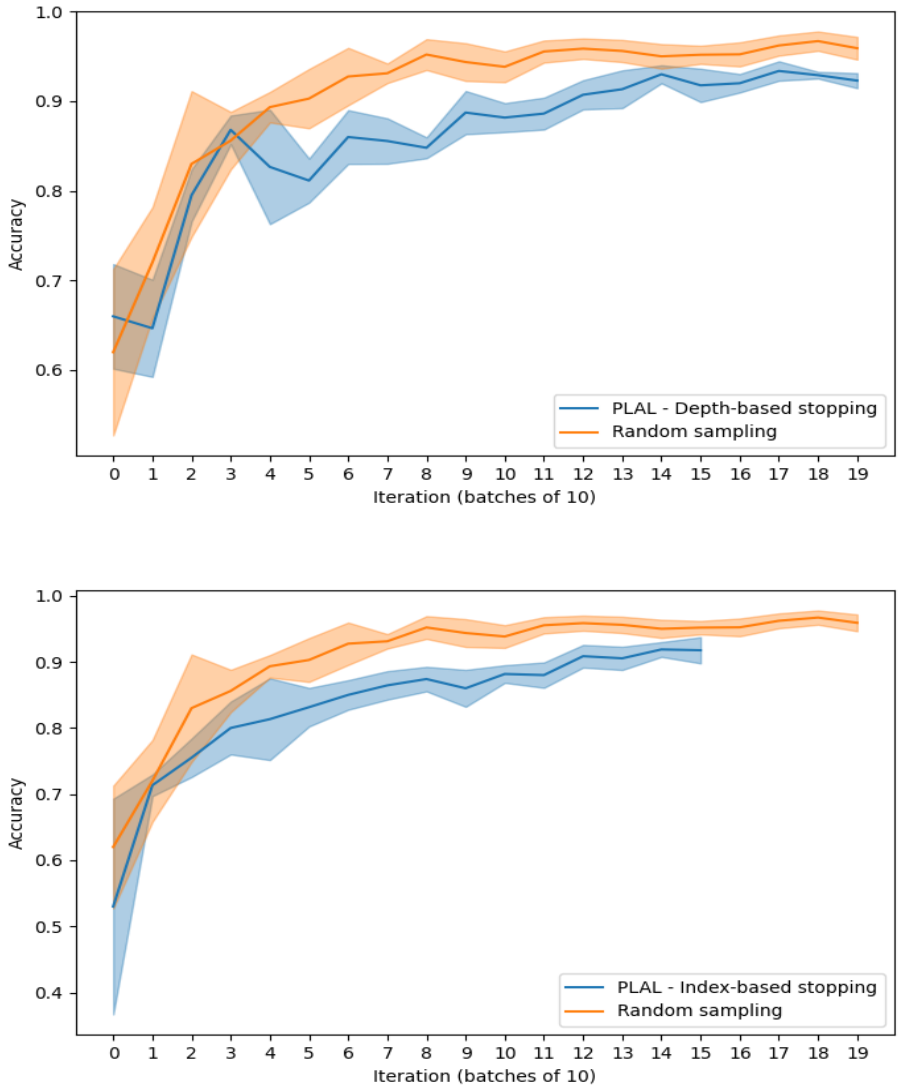

Figure 5: PLAL algorithm cross-validation accuracy compared to random sampling with depth stopping criterion (top) and index stopping criterion (bottom). Calculations done using a depth threshold of 9 and a Jaccard index threshold of 110. Translucent portions indicate ± 1 standard deviation across simulations.

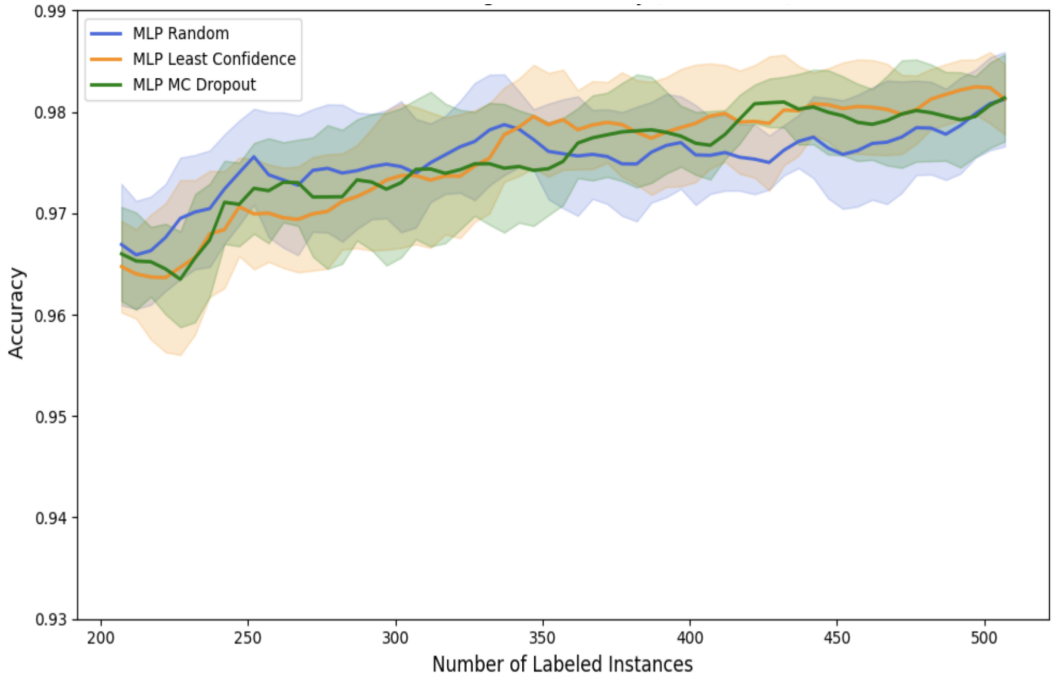

Figure 6: Active Learning accuracy curves comparing MLP uncertainty methods (Random baseline, Least Confidence, MC Dropout) using incremental labeling.

dropout consistently captured uncertainty, substantially reducing labeling efforts while maintaining accuracy.(Figure 7)

These results clearly demonstrate the practical advantages of MC dropout as an uncertainty sampling method in active learning workflows for deep learning models, leading to more efficient experimental designs for biofilm phenotype classification.

3.2 Downstream Analysis of Results

Table 1: Mean number of points sampled and estimated experimental time after applying stopping criteria. Estimated experimental time calculated as the mean number of points sampled over total number of samples taken per week.

Algorithm	Mean Number of Points Sampled	Estimated Experimental Time
Uncertainty Sampling (SGD)	280	0.97 weeks
Query by committee	196	0.68 weeks
IWAL	164.2	0.57 weeks
DH + Index-based stopping	87	0.3 weeks
DH + Depth-based stopping	69.1	0.28 weeks
PLAL + Index-based stopping	165	0.57 weeks
PLAL + Depth-based stopping	205	0.71 weeks
MLP + MC-Dropout	330	1.33 weeks

Our results highlights the practical benefit of applying stopping criteria to reduce the experimental burden of labeling (Table 1). Methods like DH with Depth-based stopping and Index-based stopping yielded the lowest sample requirements (69–87 points), translating to less than a third of a week of estimated experimental time. Uncertainty Sampling (SGD) required more samples (280), yet still cut down over half the time compared to labeling all 1296 samples. MC Dropout, while more sample-intensive (330 points), offered reliable performance in deep learning settings.

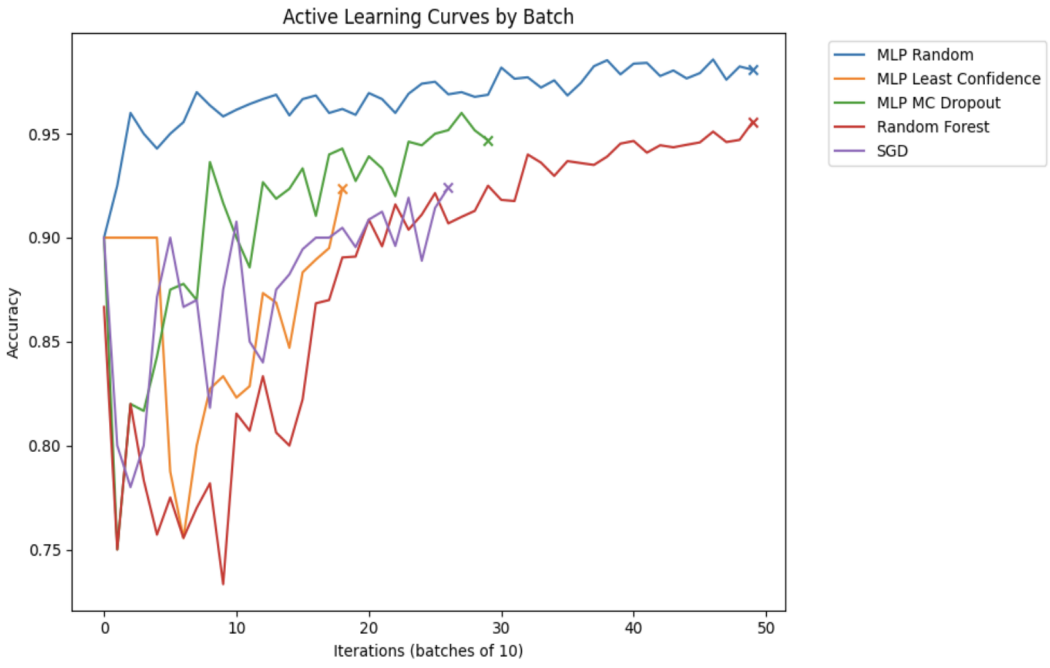

Figure 7: Overall active learning accuracy curves for various models, starting with 30 labeled samples and adding batches of 10. Markers indicate the stopping points determined by model-specific uncertainty criteria.

In comparing the accuracies of each method with offline learning, our findings placed all non-deep learning offline models at around 95% accuracy with high accuracy on each class (Figure A2). All query selection strategies yielded models with cross-validation accuracy close to this; the closest accuracies were QBC and IWAL, but DH and PLAL had results that were close. The various forms of deep learning yielded even better performance of around 98% accuracy, indicating that even for a model with many parameters, high predictive power can still be reached on a portion of the full dataset.

4 Discussion

Our work on stopping criteria show that while stopping criteria are plentiful in the context of active learning, it is often difficult to come up with good stopping criteria. While we show that stopping criteria for QBC, DH, and PLAL can often show good results in terms of the number of points selected before stopping as well as cross-validation accuracy, our stopping criteria for IWAL is inconsistent and at worst has poor performance. Like previous literature [6, 11, 12, 8], our stopping criteria have no theoretical guarantees and do not meet the criteria for mathematically valid stopping criteria [5] but nonetheless can often yield good performance, similar to offline learning in many simulations.

In the context of collecting *Vibrio cholerae* biofilm data, our results indicate that collecting 1296 videos is not necessary to reach good genotype classification results. Almost all algorithms were able to reach a high cross-validation accuracy, and as Table 1 indicates, the usage of active learning and stopping criteria can reduce estimated experimental time to collect all the necessary data by more than 4 weeks in some cases.

We also observed that the choice of uncertainty metric greatly impacts performance and efficiency. For traditional machine learning methods (Random Forest and SGD), using combined least confidence and entropy proved to be far superior compared to using least confidence alone, which often led to premature or inconsistent stopping. Furthermore, our deep learning-based uncertainty sampling method (MC Dropout) demonstrated excellent performance, consistently outperforming traditional machine learning methods in terms of accuracy and labeling efficiency. Ultimately, selecting the

best active learning strategy depends on balancing accuracy needs with practical constraints of each method.

4.1 Limitations

We recognize that while active learning often relies on heuristics and that good stopping criteria may perform well, algorithms can often benefit from formal guarantees; in fact, recent work in active learning, especially regarding the studied IWAL, DH, and PLAL algorithms [1, 3, 9] not only seeks to propose new query selection strategies but also prove theoretical guarantees about their efficacy and sample complexity. To our knowledge, little to no literature exists regarding active learning stopping criteria with theoretical guarantees. This forms a limitation not only on our work but also on stopping criteria in general, as having a theoretical guarantee of when a stopping criteria can be met may reassure scientists that a situation where the criteria fails will not occur. For example, if a stopping criterion with theoretical guarantees were applied to IWAL, we would not have observed such inconsistent stopping (Figure A1).

Alternatively, the true efficacy of our experimented stopping criteria must be tested on more datasets of varying complexity. While these stopping criteria may be applicable to collecting *Vibrio cholerae* biofilm data, there is no evidence to support that they can be applied more generally. This limitation makes our stopping criteria good for our purposes, but we cannot truly assess the goodness of a stopping criteria in more general situations.

4.2 Future Directions

Given that we've achieved theoretical stopping points that are considerably less than the size of the original dataset, we would ideally like to implement the same wet lab experimental protocol described in the data acquisition steps. If our stopping criteria are correct, then we would only need approximately one-sixth of the full dataset to actually achieve good classification results.

Additionally, we would like to implement the same stopping criteria approaches on data that might have less discriminative phenotypes. There are a few very distinct phenotypes that can be observable from the original mutant data set. Therefore, having mutants that follow a large distribution of potential phenotypes could better inform us on better stopping conditions, since in actuality phenotypes from image data tend to follow a distribution and are generally not as easily clusterable as the curated dataset that we have.

Code Availability

All code is available at https://github.com/jonazhu/ACTIVibrio.

References

[1] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 49–56, 2009.

[2] Sanjoy Dasgupta. Two faces of active learning. *Theoretical computer science*, 412(19):1767–1781, 2011.

[3] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215, 2008.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

[5] Olav Kallenberg et al. *Random measures, theory and applications*, volume 1. Springer, 2017.

[6] Florian Laws and Hinrich Schütze. Stopping criteria for active learning of named entity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 465–472, 2008.

[7] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. page 287–294. Association for Computing Machinery, 1992.

[8] Katrin Tomanek, Joachim Wermter, and Udo Hahn. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 486–495, 2007.

[9] Ruth Urner, Sharon Wulff, and Shai Ben-David. Plal: Cluster-based active learning. In *Conference on learning theory*, pages 376–397. PMLR, 2013.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2023.

[11] Andreas Vlachos. A stopping criterion for active learning. *Computer Speech & Language*, 22(3):295–312, 2008.

[12] Jingbo Zhu, Huizhen Wang, Eduard Hovy, and Matthew Ma. Confidence-based stopping criteria for active learning for data annotation. *ACM Transactions on Speech and Language Processing (TSLP)*, 6(3):1–24, 2010.

Appendix

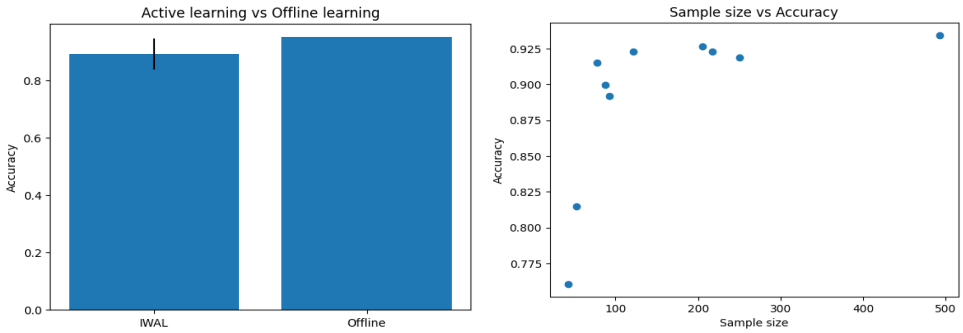

Figure A1: Accuracy of models trained on selected IWAL datasets. (left) Mean accuracy of all models trained on IWAL selected datasets. Error bar represents ± 1 SD. (right) Accuracy on unseen data as a function of dataset size.

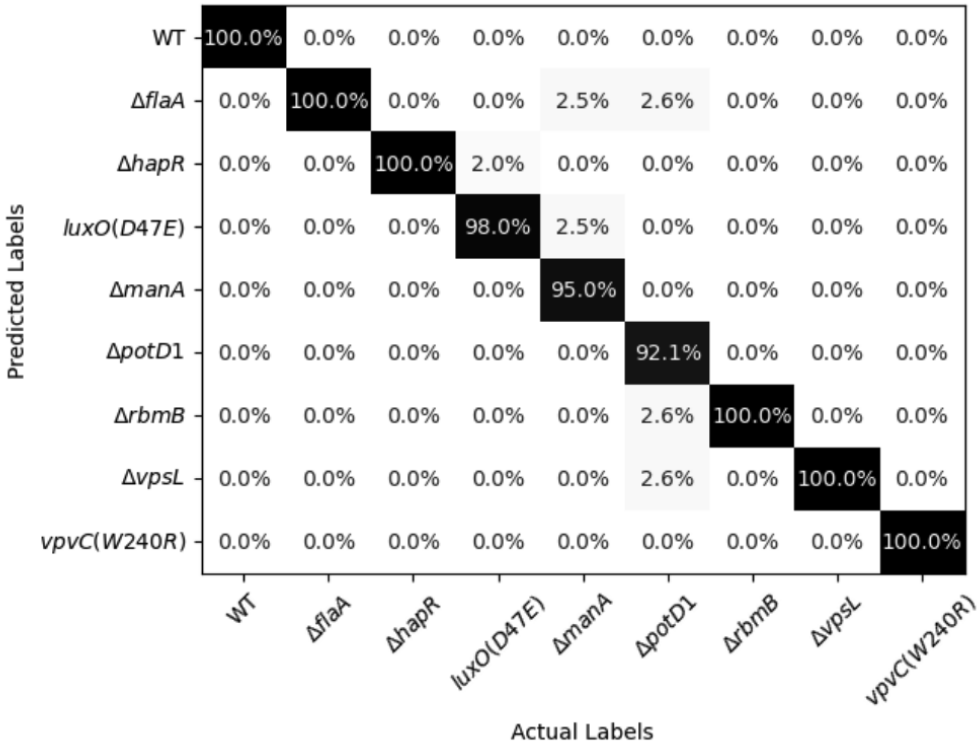

Figure A2: Confusion matrix for models trained in an offline learning setting, showing around 95% accuracy in offline learning.