Automated Detection of Bacterial Flagellar Motors

Eben Lonsdale

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Dr. Gus Hart, Advisor

Department of Physics and Astronomy

Brigham Young University

ABSTRACT

Automated Detection of Bacterial Flagellar Motors

Eben Lonsdale
Department of Physics and Astronomy, BYU
Bachelor of Science

With advances in cryogenic electron tomography, the ability to study bacterial structures in their cellular context has improved. However, 3D images of bacteria, called tomograms, have a low signal-to-noise ratio. This makes annotating structures of interest difficult, as traditional computer vision models struggle with tomograms and manual annotation is time consuming. We build on the results of the BYU Kaggle competition to create an ensemble model capable of automatically annotating flagellar models with nearly around 85% accuracy.

Keywords: cell locomotion, cryogenic electron tomography, machine learning, computer vision, flagellar motors

ACKNOWLEDGMENTS

This research would not have been possible without the financial support of the BYU Physics department and the Chan-Zuckerberg Institute. I express my gratitude for Gus Hart for being such a wonderful advisor. I am also grateful for the support and help from the members of my research group in many aspects of this project, specifically Braxton Owens and Jackson Pond. Finally, I would like to thank my friends and family for their support, encouragement, and faith.

Contents

List of Figures

Chapter 1

Introduction

Understanding the microscopic world is an important challenge, and bacteria are a key player in that world. One way we study bacteria is through cryogenic electron tomography (cryo-ET), which allows bacteria to be studied in their "native biological context" [1]. In cryo-ET, bacterial samples are flash frozen, then imaged at multiple angles, and then reconstructed into a 3D image volume from the 2D image projections [2]. This ability to image whole bacteria makes cryo-ET powerful, as it allows bacterial structures to be studied in their cellular context.

Another important part of cryo-ET is a technique called subtomogram averaging. In subtomogram averaging, "a large number of particles from the same macromolecular complexes are aligned and averaged" [1]. In structural biology, macromolecular protein complexes of interest, such as flagellar motors or ribosomes, are often referred to as particles. Aligning and averaging over many particles improves resolution and helps filter out noise in the image (see Fig. 1.1). A 3D electron density map can be constructed from the averaged particle volume, which is then matched with molecular models of the complex to try and simulate the complex's possible function in the bacteria.

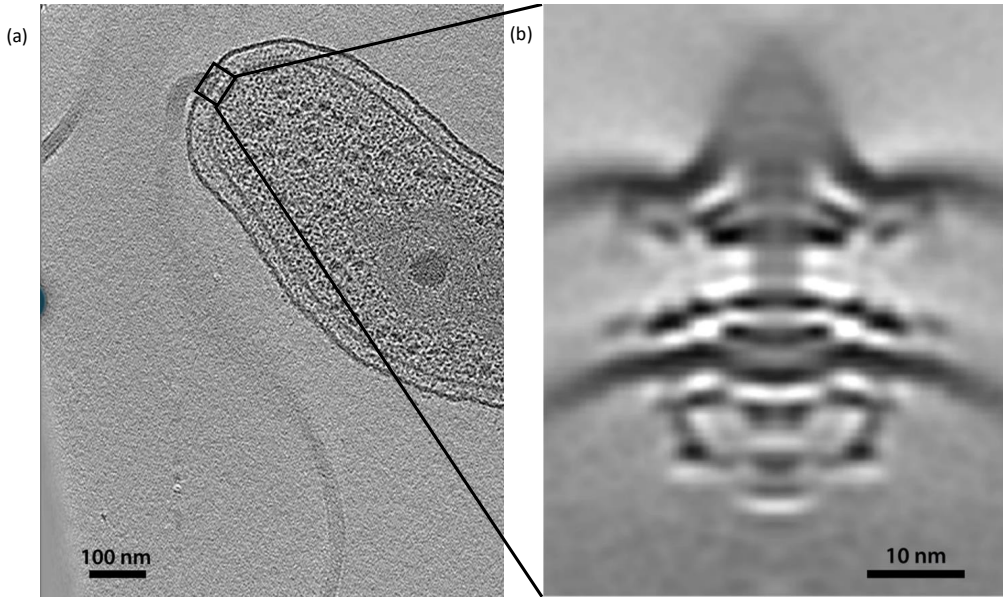

Figure 1.1 (a) Image from a single tomogram of *Bdellovibrio bacteriovorus* with the flagellar motor on the top left of the bacteria. (b) Sub-tomogram average of *Bdellovibrio bacteriovorus* flagellar motor, averaging over 1000 motors. Figure adapted from [2].

However, cryo-ET's ability to provide *in situ* information about bacteria does not come without challenges. First, the size of the data generated by cryo-ET can be challenging to work with. In our data, the average size of an uncompressed tomogram is generally between 1–2 gigabytes. Second, due to the nature of cryo-ET, the signal-to-noise ratio (SNR) is often lower than other protein structure imaging methods [1]. The low SNR in a tomogram can slow down identification of macromolecular structures in the tomogram, called particle picking, by making it difficult to identify particles of interest. The low SNR also means more particles are required to average out noise and achieve high resolution in a subtomogram average.

Traditional classification and segmentation algorithms struggle because of the large data size and low SNR, creating a particle picking bottleneck in cryo-ET. One promising avenue to accelerate particle picking is machine learning (ML), as current ML methods can be cheaper computationally and more accurate than traditional algorithms [1, 3]. Furthermore, while traditional algorithms are

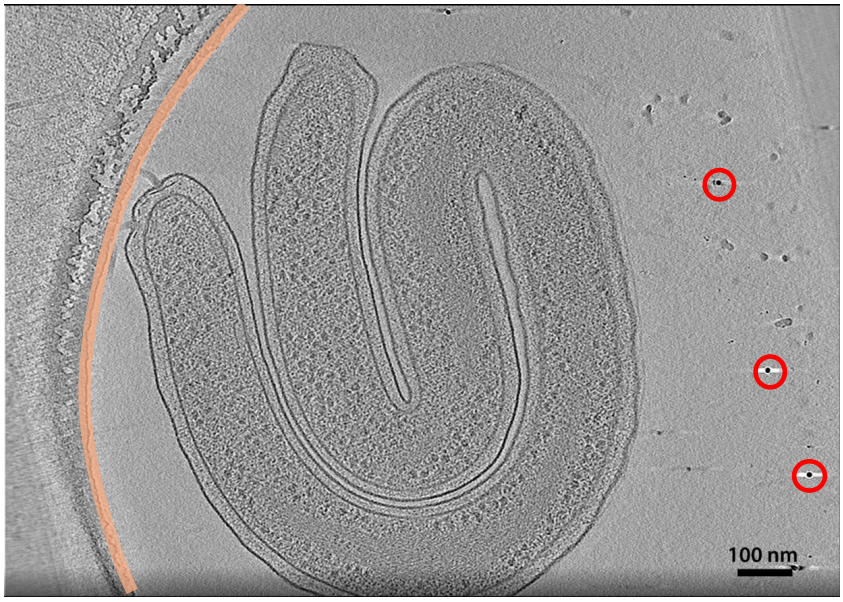

Figure 1.2 Sample tomogram of a *Hylemonella gracilis* bacteria. Note the carbon ring (orange) on the left side of the figure and the fiducial markers (red) on the right side of the figure. These common noise sources often cause computer vision algorithms to struggle. Figure from [2].

limited to identifying known particles, some machine learning algorithms have the potential to identify new particles that they have not been trained on [3].

To address this challenge, the BYU Biophysics Simulation Group hosted a Kaggle competition challenging people to build ML models that could identify flagellar motors in bacteria [4]. Kaggle is an online platform for machine learning challenges that is well known in the data science community. The goal of the competition was to increase community involvement and awareness of the open challenge presented by particle picking in bacterial tomograms. The result was a variety of machine learning models that could identify flagellar motors in the test dataset with over 85% accuracy [4].

Based on the top competition models, we have built an ensemble model to annotate flagellar motors. Ensemble models use the information provided by each base model to improve prediction accuracy [5]. In principle, each base model has learned (and possibly overfit) different features of

the data. By combining their results, the ensemble model does a better job generalizing on the data. The ensemble model described here is based on the voting and averaging method of ensembling.

Chapter 2

Methods

While an ensemble model is ideally more than the sum of its parts, it is important to understand the function of each base model. In our case, each base model took a different approach to the flagellar motor identification problem. Each model's approach is described in turn to provide a greater understanding for the ensemble model as a whole. We also discuss the our choices for the overlying ensemble model.

2.1 Kaggle Competition Base Models

Kaggle is an online platform for hosting data science competitions. A competition host uploads data to the platform and outlines a data science problem for the community. Data scientists from around the world are then able to engage in the problem. Often, this provides novel approaches to solving the problem, as people from outside the discipline bring fresh techniques and perspectives. The BYU Biophysics Group hosted a Kaggle competition challenging people to identify flagellar motors in *in situ* tomograms. For more details about the BYU competition, see [6]. At the conclusion of the competition, four of the top models were chosen based on their competition scores and the ease of running the model outside the Kaggle environment. These four models became the base models

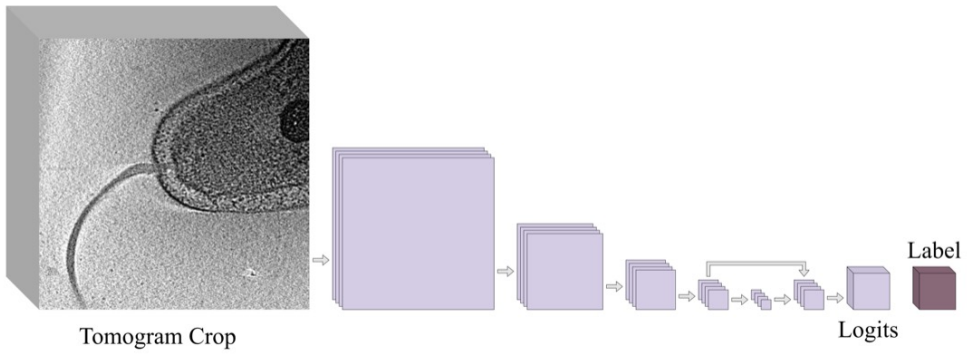

Figure 2.1 Simplified schematic of Bartley's model. Tomograms are first cropped into patches, then passed through several 3D UNet layers of decreasing size. Final logits are used to create prediction labels. Figure from [7].

for the ensemble model used in this project. Each of the following subsections is labeled by the name of the team that developed the base model.

2.1.1 Bartley

The first place model in the Kaggle competition was based on a 3D U-Net. First, the model used a ResNet200 to encode the tomogram into a lower dimension latent space and then used the 3D U-Net learn and predict motor locations [7].

Instead of using specific pixel values as annotations, Bartley used blob regression by converting the training set annotations to Gaussian heatmaps centered on the original annotations [7]. Bartley also augmented the original training data provided in the BYU competition by downloading additional data from the CZI CryoET Data Portal [8]. He then hand annotated the data, which he

generously made publicly available. He trained his model using all of this data. He used smooth binary cross entropy as his loss function, and he trained for 400 epochs . [7]

Once the model was trained, Bartley used an 8-seed ensemble model to do inference. This means he trained 8 separate models starting from different random seeds for the model weights. This stochastic variance produces slightly different final models, whose results were averaged to produce final predictions. Overlapping windows were used to slide over each tomogram in inference to produce motor predictions.

Once all the predictions were produced, the max predicted values and statistical thresholding techniques were used to choose the final predictions [7]. One option for thresholding is quantile thresholding. This method assumes that the data follows an underlying probability distribution. Instead of choosing a fixed value to be the prediction threshold, quantile thresholding chooses a certain percentage, or quantile, of the data to be considered positive. This allows the model to be more flexible. However, it can also be a drawback if the distribution of the train data does not reflect the general data distribution. In contrast, for fixed thresholding a certain score of an accuracy or loss metric is chosen and all values greater than or equal to it are considered positive predictions. While this approach is intuitive, it can struggle on variable datasets like tomograms. Because of the need for greater model flexibility, Bartley used quantile thresholding in his preditions.

2.1.2 MIC_DKFZ

The MIC_DKFZ team, which comes from the German Cancer Research Center, also based their solution on an in-house implementation of a 3D-UNet, called nnU-Net [9]. nnU-Net is a framework originally designed for 3D medical image segmentation, a closely related problem [10]. To learn more about the structure of nnU-Net, see [11].

For their training data, they also used the provided training data, Bartley's data, and additional data they curated from the CZI CryoET Data Portal [8]. Similar to Bartley, they used blob regression

for their annotations. However, they chose to use Euclidean distance transfer blobs as opposed to gaussians heatmaps. Binary cross entropy was as the loss function and models were trained for 300 epochs [9].

For inference, the MIC_DKFZ team used overlapping patches to parse each tomogram. Prediction values were normalized using a sigmoid function. Unlike Bartley, they used fixed thresholds to determine positive predictions.

2.1.3 Daddies

This model took a unique approach in the competition by combining two qualitatively different models, a classification model and an object detection model.

The classification model was built from a ResNet18 classifier (see 2.2). The basic idea behind this approach is simple, but clever. Each tomogram is flattened into a vector, and then the classifier is trained to predict if each area of the tomogram contains a motor, plus one extra class for no motor in the tomogram. This idea is interesting, as unlike the U-Net approaches, this model does not account for spatial relationships in the tomogram. The ResNet model was trained with cross entropy loss and also used a sliding window and quantile thresholding for inference prediction [12].

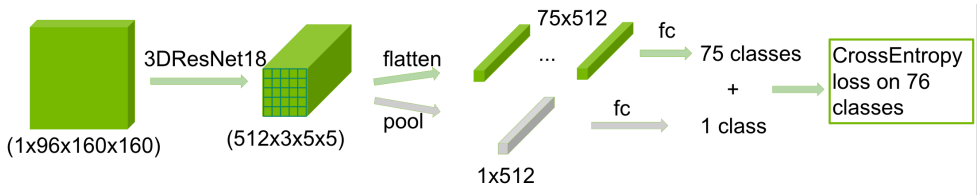

Figure 2.2 Simplified schematic of the ResNet18 classifier. Tomograms are flattened and pooled. The ResNet then uses 75 classes for positive motor presence (one for each patch) and one class for no motor. Figure from [12]

The object detection model is based on MONAI architecture, a popular framework for medical image processing. This model was built on a combination of 2D and 3D models that encode the tomogram into the feature map. This 2D/3D combination is referred to as a 2.5D model. The feature maps from the 2.5D model are used to generate predictions in place of full tomograms, accelerating inference. Sliding windows were also used for inference [13].

Once both predictions were generated, the two predictions were combined to create more accurate overall predictions [13].

2.1.4 outrunner

The final model used had the simplest approach of all the models chosen for the pipeline. This is because it relied on pretrained models. First, the tomograms were divided into stacks of 2D slices, which allows them to be passed into 2D models. 2D models are easier to implement and more popular than 3D ones due to their use in processing standard images. Once the tomograms were converted to slices, outrunner used YOLO models from Ultralytics to do classification and prediction on them [14]. YOLO (You Only Look Once) models are pretrained computer vision models that excel at object classification in images. outrunner retrained pretrained yolo8/yolo12 models on the tomograms to improve their performance. He experimented with both fixed and quantile thresholding to identify his positive predictions and found that both worked well.

2.2 Ensemble Model

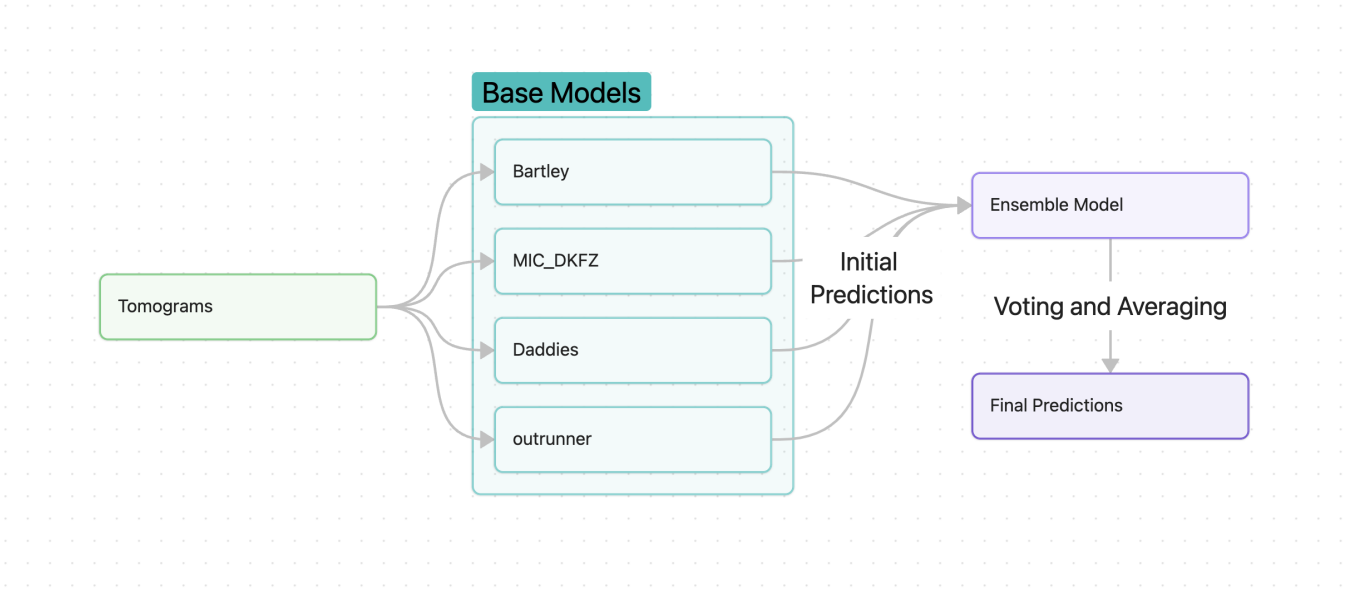

Figure 2.3 Information flow in the ensemble model. Individual tomograms are passed into each base model. Each base model runs inference in parallel on the tomogram, and the initial predictions are passed to the ensemble model. The ensemble model uses voting and averaging to produce the final predictions.

These four models were incorporated into a simple ensemble model. First, each model runs inference in parallel on the same dataset, so order does not matter. Once each model has finished inference, each model's prediction is used to cast a boolean vote on motor presence in the tomogram. If the vote is ≥ 3, then the ensemble model considers that to be a positive motor prediction.

If a positive motor prediction is been made, all the predictions from models that voted "yes" for motor presence are averaged to create the final motor coordinate prediction. Currently, this prediction consists only of a (x, y, z) coordinate and does not return any information about motor orientation. In manual annotation, this information often is provided by aligning the long motor axis with the y-axis of the image and recording the image angle needed to produce this alignment.

If the threshold for positive motor is not met, then the ensemble model marks the tomogram as having no motor. In our data, this is notated by marking the motor coordinates as $(-1, -1, -1)$. This occurs even if 1 or 2 of the models predicted a motor. The goal of this is to filter out false

positives that individual models are sensitive to. False positives occur when the model predicts a motor when none is present.

2.3 Running the Models

Models are hosted and run on facilities maintained by the BYU Office of Research Computing. This is done to be able to store the large weight files required to run each model and to have easier access to tomography data. GPUs are also essential to running these models; in the pipeline primarily NVIDIA A100 and NVIDIA H200 GPUs were used to accelerate model inference. Bash scrips are used in conjunction with Slurm to organize data flow and run the ensemble pipeline automatically.

Chapter 3

Results

Flagellar motor identification remains challenging. In the ensemble pipeline, base models were able to identify around 85% of the motors in the test set. However, the ensemble models did not seem improve significantly over some of the base models, as seen in Figure 3.1.

To explore this, we took a small test dataset of 122 tomograms that we had ground truth annotations for and ran the ensemble model on them, saving the predictions of the base and the

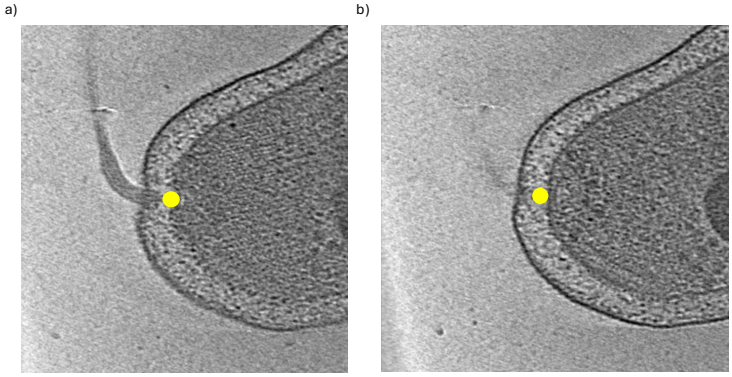

Figure 3.1 a) Motor prediction (yellow) from MIC_DKFZ's nnU-Net model b) Motor prediction (yellow) from the ensemble model. Note the flagellum and the motor structure are much clearer in the MIC_DKFZ prediction.

Table 3.1 F-β scores for the base models and the ensemble model in a test dataset of 122 tomograms.

Model	**F-β Score**
Bartley	0.958
MIC_DKFZ	0.933
Daddies	0.680
outrunner	0.584
Ensemble	**0.745**

ensemble models. To create a simple baseline for comparing the models, we decided to focus only on motor detection and neglect motor localization for this test. We used the results of this to calculate the F-β scores for each model compared to the ground truth. F-β is a metric to help us understand the performance of our model in terms of recall and precision [15]. Recall measures how many positive model predictions are accurate, while precision measures how many of the postive instances in our data the model correctly identified, and the F-β metric measures how well the model does at balancing between recall and precision [15]. β acts as a tuning parameter for the balance between the recall and precision [15]. We used $\beta = 2$, as in the Kaggle competition metric, which emphasizes recall and punishes false negatives more. In Table 3.1, we see that Bartley's model did the best, closely followed by the MIC_DKFZ model, while Daddies and outrunner's models both struggeled on this dataset. The ensemble model's score is close to the average of the scores of the base models.

To gain further insight, we plotted the motor detection results of each model, as well as the ground truth in Figure 3.2. These plots show that Barley's model tends to overpredict the number of positive motors, while Daddies' and outrunner's models tend to underpredict the number of motors.

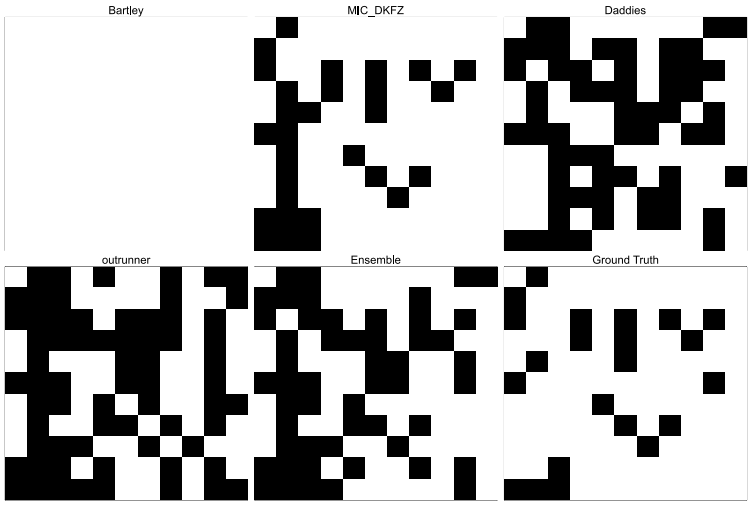

Figure 3.2 Each heatmap shows the motor detection of the given model, with the ground truth plotted for reference. Each pixel represents one tomogram. Black squares are negative motor predictions and white squares are postive motor predictons.

The large overlap in missed motors in these two models leads the ensemble model to miss some motor predictions as well.

Chapter 4

Discussion

Our results show that our voting-based ensemble model had no significant improvment over the top performing models. However, the pipeline annotated several orders of magnitude faster than manual annotation. Our results also suggest that some of the base models held back the performance of the ensemble model. Daddies' and outrunner's base models specifially did not perform well and were the likely culprits for the disappointing performance of the ensemble model. Another limitation of the ensemble model is that it can be time-prohibitive to run. Depending on the size of your dataset, running each base model's inference can take quite awhile, especially if you do not have access to a high performance comptuting cluster. However, the annotation speed still outstrips the previous standard of manual annotation.

We also noted that both the base and the ensemble model struggled with noisier tomograms, often identifying false positives. This was especially apparent near sharp boundaries in the image, such when as fiducial markers and carbon rings were close to bacterial membranes. Despite this, the models were able to identify motors that may have been missed by manual annotation. The value of these motors may be questionable however, as the image quality of these motors is often poor. Becuause of this, including these motors in a sub-tomogram average may actually decrease the final resolution, due to the high levels of noise in motor images near sharp boundaries.

Despite these limitations, the fact that automatic annotation is 3–4 times faster than manual annotation was is remarkable. While molecular models of flagellar motors existed previously, the slow annotation speed made it difficult to compare flagellar motors across species [16]. Increased annotation speed means it will be easier to create flagellar motors models for different species, allowing researchers to analyze the differences in motor structure and function. This in turn can lead to a greater understanding of bacterial motility and infection pathways. Beyond accelerating particle picking, the pipeline also found motors that would be easily missed in manual annotation. These motors often lack an obvious flagellum, are close to the edge of the tomogram, or are near an artifact obscuring the flagellum. While these motors may not always be usable in sub-tomogram averaging, this suggests that the models may have learned patterns about the shape of flagellar motors, instead of only learning contextual patterns to identify motors.

This research is similar to progress made in the Kaggle competition run by the Chan-Zuckerberg Imaging Institute (CZII). That competition focused on multi-class object detection in a small, curated dataset [17]. In comparison, our project focused only on predicting flagellar motors in a larger and more varied dataset. This makes the dataset used in our competition comparable to data structural biologists have access to. We hope that this will make machine learning techniques easier to apply in the field. Ultimately, we would love to see annotation models integrated into existing cryo-ET workflows. The goal would be to take an image, run it through an automatic reconstruction algorithm, run automatic annotation, and then have a researcher verify the results.

Chapter 5

Conclusion

In summary, the BYU Kaggle competition showed that annotating flagellar motors using machine learning is possible, but it still is limited. Currently, most of these annotations will be used to create sub-tomogram averages of the flagellar motor. Sub-tomogram averaging requires all the motors to be aligned along a common axis, yet our models do not provide any information about motor orientation, so human alignment of motors is still required before averaging can be performed. Along with this, some of the motors identified by the algorithms may not be suitable for sub-tomogram averaging. This is especially true of the motors that would have been missed by manual annotation. They are often in noisier areas of the tomogram, so they may not contain adequate information to be used in sub-tomogram averaging.

In part because of this, our model annotations still require human verification. Even if one does not need motor orientation data, human verification is still needed to filter out false positives or noisy motors. False positives generally occur on or near artifacts like fiducial markers or carbon rings. The models could be trained more to improve their accuracy, but further experimentation is required to see if this is a viable solution. We see streamlining human verification as a better option, as keeping the human element in the research process is valuable.

Another limitation of the current annotation pipeline is its ease of use. Though the models annotate faster than a human could, they require GPUs to run. This type of hardware is not always available to structural biology labs, so the researchers running the models on their own data could be challenging. Along with this, our group needs to improve the pipeline packaging to make it easier to share publicity. Currently, setting up the pipeline requires knowledge of Linux, Python, and Bash scripting to set up and run. We plan to improve the code organization so that the models can be easily shared through GitHub or similar platforms.

For future research, we want to explore other ensembling techniques to try and improve model accuracy. One idea is to use a simple neural network as the ensemble model instead of voting and averaging. This is based on the idea is that each base model will have different types of tomograms/motor locations that it does well or poorly with. One could build a simple autoencoder for tomograms, then use the autoencoder to compress the tomograms to a smaller latent space; this latent space would represent feature vectors for tomograms. These feature vectors, along with the predictions from the base models, could be used to train the simple neural network to adaptively weight the ensemble model.

While flagellar motors are fascinating, they are only a small window into the bacterial world and improving our understanding of other structures would be amazing. With the progress shown in annotating flagellar motors automatically, we are confident that the particle picking bottleneck problem is solvable. We hope to see more research applying machine learning to explore problems in structural biology.

Bibliography

[1] T. Wagner and S. Raunser, "Cryo-electron tomography: Challenges and computational strategies for particle picking," Current Opinion in Structural Biology **93,** 103113 .

[2] C. M. Oikonomou and G. J. Jensen, *The Atlas of Bacterial & Archaeal Cell Structure*, 2.4 ed.

[3] W. Wan, "A practical look at cryo-electron tomography image processing: Key considerations for new biological discoveries," Current Opinion in Structural Biology **93,** 103116 .

[4] "BYU - Locating Bacterial Flagellar Motors 2025 | Kaggle,", https://www.kaggle.com/competitions/byu-locating-bacterial-flagellar-motors-2025, 2025.

[5] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," Frontiers of Computer Science **14,** 241–258 .

[6] C. B. Owens *et al.*, "MotorBench: A Cryo-Electron Tomography Dataset of Bacterial Flagellar Motors for Testing Detection Algorithms,", 2025, pages: 2025.04.23.650258 Section: New Results.

[7] "1st Place - 3D U-Net + Quantile Thresholding | Kaggle,", https://www.kaggle.com/competitions/byu-locating-bacterial-flagellar-motors-2025/writeups/bartley-1st-place-3d-u-net-quantile-thresholding, 2025.

[8] U. Ermel *et al.*, "A data portal for providing standardized annotations for cryo-electron tomography," Nature Methods **21,** 2200–2202 (2024).

[9] "2nd place solution - 3D nnU-Net + blob regression | Kaggle,", https://www.kaggle.com/competitions/byu-locating-bacterial-flagellar-motors-2025/writeups/mic-dkfz-2nd-place-solution-3d-nnu-net-blob-regres, 2025.

[10] F. Isensee, T. Wald, C. Ulrich, M. Baumgartner, S. Roy, K. Maier-Hein, and P. F. Jaeger, "MIC-DKFZ/nnUNet,", https://github.com/MIC-DKFZ/nnUNet, 2025, original-date: 2019-04-17T08:10:56Z.

[11] F. Isensee, T. Wald, C. Ulrich, M. Baumgartner, S. Roy, K. Maier-Hein, and P. F. Jaeger, "nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation,", 2024, arXiv:2404.09556 [cs].

[12] "4th place: Simple ResNet18 classification | Kaggle,", https://www.kaggle.com//competitions/byu-locating-bacterial-flagellar-motors-2025/writeups/daddies-4th-place-simple-resnet18-classification, 2025.

[13] E. Khvedchenya, "4-th place solution (Detection Part),", https://www.kaggle.com/competitions/byu-locating-bacterial-flagellar-motors-2025/discussion/583228, 2025.

[14] "6th place solution - Ultralytics YOLO | Kaggle,", https://www.kaggle.com//competitions/byu-locating-bacterial-flagellar-motors-2025/writeups/yuan-high-tech-6th-place-solution-ultralytics-yolo, 2025.

[15] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool," BMC Med Imaging **15,** 29 (2015).

[16] P. K. Singh, P. Sharma, O. Afanzar, M. H. Goldfarb, E. Maklashina, M. Eisenbach, G. Cecchini, and T. M. Iverson, "CryoEM structures reveal how the bacterial flagellum rotates and switches direction," Nature Microbiology **9,** 1271–1281 , publisher: Nature Publishing Group.

[17] "CZII - CryoET Object Identification,", https://kaggle.com/czii-cryo-et-object-identification, 2025.

Index