BioMON: Biomedical Few-shot Meta-learning Architectures with Classification Base-Learners

Manos Chatzakis   Lluka Stojollari

Abstract

This paper presents BioMON, a few-shot meta-learning architecture that utilizes a wide range of classification base learners to perform classification tasks for Biomedical data collections. Our approach employs both classic Machine Learning and Deep Neural Network classifiers, that operate on deep neural network embeddings generated from few-shot meta-learning networks. Our experimental evaluation demonstrates that BioMON achieves great performance for the Swissprot dataset, reaching an accuracy of 66%, using a Logistic Regression base learner, overpassing state-of-the-art competitor algorithms. Furthermore, BioMON exhibits commendable results for the Tabula Muris dataset, reaching up to 88% accuracy using the same settings. We make our code publicly available on GitHub [1].

1. Introduction

Motivation The realm of biomedical machine learning confronts a notable challenge - the paucity of labeled datasets. This limitation propels us towards sophisticated approaches, with few-shot learning emerging as a strategic avenue to enhance performance across various biomedical tasks (Kotia et al., 2021),(Li et al., 2021b),(Singh et al., 2021),(Ning et al., 2021).

Challenge. The pivotal challenge of the few-shot pipeline is to develop methods that, given the input embeddings, can accurately classify examples within the query set. Past approaches have often relied on distance metrics methods (Snell, 2017). However, in the few-shot regime, discriminatively trained linear predictors tend to offer superior generalization (Chen et al., 2019b). The aim is to use these predictors as base learners to learn representations for few-shot learning, striking a better balance between feature size and performance across various few-shot recognition benchmarks. Subsequently, the goal is to learn feature embeddings that generalize efficiently under a linear classification rule for novel categories. To efficiently achieve this objective, we leverage on key property of linear classifiers: the implicit differentiation of the optimality conditions of the convex problem.

Linear Classifiers in Few-Shot Learning. Discriminatively trained linear classifiers often perform on par with nearest neighbor classifiers (Malisiewicz et al., 2011), (Caruana et al., 2008) in limited-data scenarios, benefiting from the ability to leverage more abundant negative examples for improved class boundary definition. Additionally, they can effectively utilize high-dimensional feature embeddings, with model capacity controlled by appropriate regularization like weight sparsity or norm. The convex nature of linear models facilitates efficient solving, making them well-suited for this approach.

Two additional properties arise from the convex nature, enabling efficient meta-learning: implicit differentiation of the optimization (Barratt, 2018), (Gould et al., 2016), and the low-rank nature of the classifier in the few-shot setting. The first property allows the use of off-the-shelf convex optimizers to estimate the optima and implicitly differentiate the optimality to train the embedding model. These classifiers fit the support set to the support labels and subsequently provide logits over the embeddings of the query set. However, the intricate nature of biomedical data raises a specialized challenge - determining the most suitable classifier type remains an open problem.

Approach. In this direction, we introduce BioMON, a few-shot meta-learning architecture that builds upon the foundational principles set forth by MetaOptNet (Lee et al., 2019). Instead of exclusively using SVMs or Logistic Regression (Lee et al., 2019), it explores the performance and the trade-offs of a wide spectrum of classifiers, starting from KNN(Li et al., 2021a) to classic ML classifiers, encompassing Naive Bayes, Logistic Regression, SVMs, Random Forests. It further extends into the realm of Deep Learning Neural Network classifiers, that operate on biomedical data collections. Additionally, for the embedding aspect, we introduce the integration of FCNet, alongside the implementation of R2D2, thereby improving the computation of embeddings. The motivation behind embracing such diversity lies in a nuanced comprehension of the nature inherent in biomedical datasets. Each classifier embodies distinctive strengths

[1] https://github.com/MChatzakis/BioMON

and weaknesses, and our objective is to unravel the optimal coordination of these classifiers within the meta-learning framework.

Contribution. Our contribution can be outlined as follows:

- We conceptualize the BioMON architecture inspired by MetaOptNet(Lee et al., 2019), implementing a wide variety of classifiers. This multifaceted approach aims to unravel the nuanced performance trade-offs across various classifier types in the context of biomedical data.
- We incorporate different neural network models (backbones) to generate the deep embeddings of the data, utilizing both FCNet and R2D2, to experiment with diverse architectural foundations.
- We conduct a wide experimental evaluation that compares the performance of the classifiers, as well as the performance of the best-performing variations of BioMON to other benchmark algorithms for Biomedical data, proving that BioMON achieves state-of-the-art performance.

2. Method: BioMON

2.1. Description

The architecture of BioMON is summarized in figure 1. At its core, the system comprises an embedding model tasked with mapping the input domain into a feature space. The overarching objective is to enable the embedding model to learn a high-dimensional space that can be effectively divided by the corresponding classifier, which is trained on the support set using the support labels. The backbone's role is to learn a rich embedding field, facilitating the classifier in returning accurate loss metrics using the query set for embedding learning. This approach exhibits practicality and demonstrates remarkable performance by yielding substantial gains in few-shot learning tasks.

2.2. Backbones

The backbone assumes a pivotal component in this architecture, as it represents data in an embedding space, a vital step before classifying these embeddings in a high-dimensional space with different classifiers. We carefully explore two backbones techniques FCNet (Xu et al., 2020) and R2D2 (Revaud et al., 2019).

FcNet. The FCNet (Xu et al., 2020)module consists of fully connected linear layers, where each layer is followed by batch normalization, $RELU$ activation, and dropout for regularization. The use of the fast weight parameter enables the incorporation of fast weight learning techniques, where the code dynamically adjusts the weights during forward passes (Ba et al., 2016).

R2D2. R2D2 (Revaud et al., 2019) is a multi-layered module consisting of convolutional layers, batch normalization, and $LeakyReLu$ activation specifically applied to the last block, along with dropout for regularization. The R2D2 module processes input data through a sequence of convolutional blocks, followed by a final block that combines information from previous layers. The result feature representation is formed by concatenating the outputs from the last two blocks. Additionally, the utilization of 1D convolutions and max pooling contributes to the extraction of hierarchical features.

2.3. Implemented Classifiers

In the BioMON architecture, we employed a diverse set of classifiers to handle different aspects of the learning process. The following classifiers were integrated into the system: *Logistic Regression (LR)*, *Decision Trees (DT)*, *Naive Bayes (NB)*, *K-Nearest Neighbors (KNN)*, *Gaussian Mixture Model (GMM)*. For Deep Learning classification, we implement a multi-layer perception neural network (MLP). with configurable hidden dimensions and uses the $RELU$ activations.

3. Experimental Evaluation

Environment. For our experiments, we used a Google Cloud server of 4 CPU cores, 100 GB of disk memory, and an NVIDIA T4 GPU of 60 GB of RAM.

Algorithms. We compare various versions of BioMON with other few-shot learning algorithms, including Protonet (Snell, 2017), MatchingNet (Vinyals et al., 2016), MAML (Finn et al., 2017), and standard baseline models (Chen et al., 2019a). For all our experiments we use $n\_way = 5$, $n\_shot = 5$, $n\_query = 15$, and 50 episodes for the testing.

Swissprot Dataset. Swissprot (Boutet et al., 2016) is a dataset in bioinformatics that comprises a diverse array of protein sequences. The data focuses on the prediction of Gene Ontology (GO) labels based on the amino acid sequences of proteins. Each protein sequence serves as input data, and the goal is to predict the most specific GO label associated with it. The model architecture involves embedding of lengths 1280, and the dataset is split into training (59 labels), validation(15 labels) and test (9 labels) sets based on GO labels.

Tabula Muris Dataset. Tabula Muris (Cao et al., 2020) presents a distinct challenge in few-shot learning, specifically focused on cell-type annotation across tissues within a model mouse organism. The dataset encompasses single-cell gene-expression profiles from 105,960 cells, with features representing the expression levels of 2,866 genes. The cells are categorized into 124 different cell types across 23 organs. The task involves assigning cells to different cell types, and the ultimate goal is to generalize to cell types

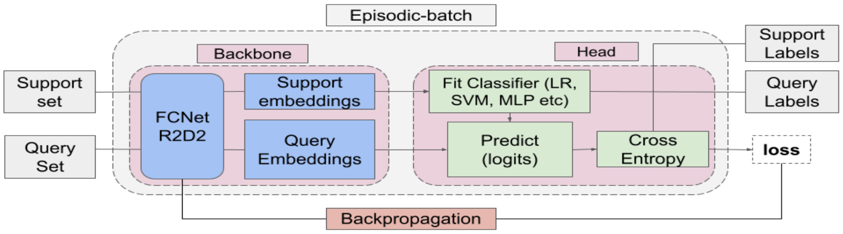

Figure 1. BioMON Architecture

in new tissues with only a few labeled examples per class. The training tissues include 59 cell types, validation tissues include 47 cell types and test tissues include 37 unseen cell types.

We begin our evaluation by seeking the best configuration for the classifiers whose results may widely vary according to their hyperparameters.

KNN Tuning. We experimented with values of k for KNN. Figure 2 presents the testing accuracy of KNN in BioMON using both backbones, for both of our datasets. Our findings suggest an almost consistent pattern: smaller k values yield better results and accuracy decreases as we increase k. Additionally, R2D2 outperforms the FCNet backbone, showing improvement of up to 10% in Tabula Muris and up to 3% in Swissprot.

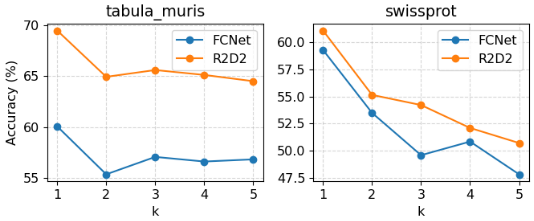

Figure 2. k-NN Tuning

RF Tuning. Random Forest performance is closely connected to the number of estimators used. Figure 3 presents the testing accuracy of BioMON using Random Forests of various estimators for both datasets. Notably, R2D2 consistently outperforms FCNet by 10% in Tabula Muris. Interestingly, a higher number of estimators enables the model to capture more intricate patterns, resulting in enhanced generalization and accuracy. However, the dynamics shift when applied to the Swissprot dataset. Surprisingly, for only 50 estimators, R2D2 yields better results compared to FCNet. For higher numbers of estimators, the accuracies are nearly identical between R2D2 and FCNet. This happened because the Swissprot dataset, being highly specialized, exhibits a unique set of features and patterns that are effectively captured by R2D2 with a modest number of estimators. The limited improvement beyond 50 estimators suggests that the dataset may have reached a saturation point where additional model complexity does not significantly contribute to better performance. This emphasizes the importance of tailoring the model's architecture to the specific characteristics of the dataset for optimal results.

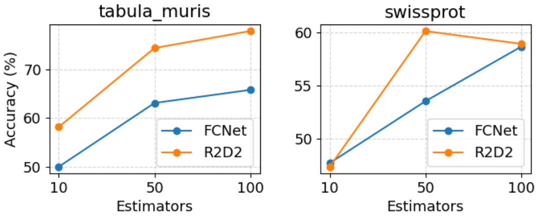

Figure 3. Random Forest Tuning

MLP Tuning. Deep Neural Networks have a wide range of tunable hyperparameters. Given our resource availability, we tuned our networks by varying the layer size and number, and epochs trained, and keeping the learning rate to 0.001 with a weight decay of 0.000001 and a dropout rate of 0.4. The results are presented in figure 4 for Tabula Muris data, for both available backbones(for Swissprot, which had the same trends, refer to Appendix). The findings indicate that more complex and deeper architectures exhibit similarities to each other across both datasets, but notably lag behind the lightweight architecture. Also, the observed trend indicates that the model's accuracy improves with an increase in the number of epochs. It is worth mentioning that, in this context, the term number of epochs refers to the epochs a model is trained for, in each batch-episodic step within the end-to-end training.

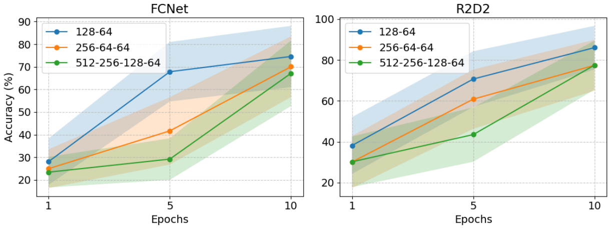

Figure 4. Tabula Muris MLP Tuning

Classifier Accuracy. After the fine-tuning, we compare the performance of the BioMON classifiers. We compare all the classifiers presented in section 2, taking the best-scoring versions of KNN, RF, and MLP from the previous experiments. The results are depicted in figure 5 for Tabula Muris and in figure 6 for Swissprot using R2D2(for FCNet, refer to Appendix). The findings indicate that for the Tabula Muris dataset, MLP, SVM, and LR outperform other classifier models using both backbones. On the other hand, for the Swissprot dataset with R2D2(for FCNet, refer to Appendix), there is no significant increase in accuracy anymore compared to FCNet, but LR, SVM, and MLP stand out again as the top-performing models. In all scenarios, the results suggest that, when combined with the R2D2 backbone, certain models can achieve accuracies comparable to benchmark algorithms.

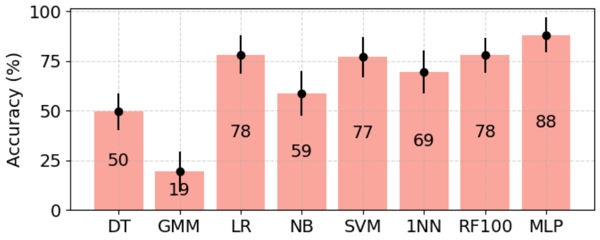

Figure 5. Classifier accuracy with R2D2 for Tabula Muris

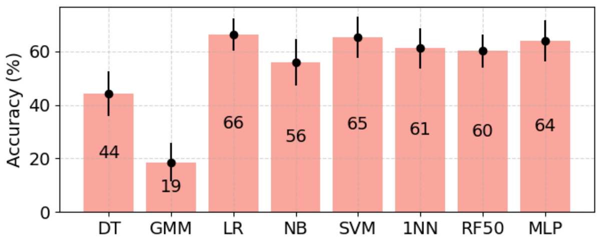

Figure 6. Classifier accuracy with R2D2 for Swissprot

Comparison with competitors. We are now comparing the best-scoring variations of BioMON with other few-shot learning algorithms, by using R2D2 as the backbone(for FCNet, refer to Appendix) and datasets available. The results are depicted in figure 7 for Tabula Muris and figure 8 for Swissprot. In Tabula Muris 7, our standout models, particularly MLP, demonstrate competitiveness with state-of-the-art performance. Notably MLP, despite resource constraints, exhibits robust performance even with limited fine-tuning, showcasing its potential as a well-defined field for exploration. However, it's essential to acknowledge that LR lags slightly behind in performance.

Conversely, the analysis of Swissprot results reveals a contrasting scenario. In both backbone settings, baseline, and Protonet secure the top positions with accuracies around 64% to 65%. Remarkably, in the R2D2 setting, BioMON achieves state-of-the-art results, reaching up to 66% of test accuracies. Notably, in the FCNet setting, our MLP outperforms Protonet, almost reaching the baseline accuracy. This emphasizes the efficacy of discriminatively trained linear classifiers, showcasing their comparable performance with nearest-neighbor classifiers

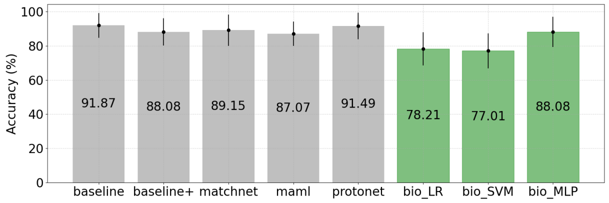

Figure 7. Few-shot algorithms for Tabula Muris using R2D2

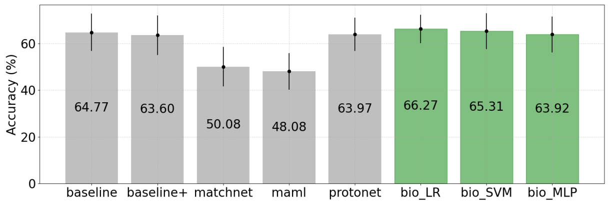

Figure 8. Few-shot algorithms for Swissprot using R2D2

4. Results

The comprehensive evaluation of BioMon, our few-shot meta-learning architecture, unveils promising performance across diverse scenarios. In fine-tuning experiments, BioMON exhibits nuances sensitivity to hyperparameters, with R2D2 consistently outperforming FCNet across various classifiers. MLP emerges as a standout performer, showcasing potential in the realm of few-shot learning. Classifier comparison highlights the effectiveness of discriminatively trained linear predictors, with MLP, LR, and SVM leading the pack, particularly in Tabula Muris. When pitted against benchmark algorithms, BioMON, especially with the R2D2 backbone, achieves competitive accuracies, demonstrating its efficacy in biomedical few-shot tasks. Notably, in the Swissprot dataset, BioMON with R2D2 achieves a state-of-the-art(compared to benchmark algorithms) accuracy of 66%,surpassing benchmark algorithms.

5. Conclusion

In conclusion, BioMON presents a versatile approach to biomedical few-shot meta-learning, leveraging a diverse array of classifiers and backbones. The findings underscore the importance of carefully tuning the hyperparameters and the suitability of discriminatively trained linear classifiers in biomedical data. MLP, LR, and SVM emerge as robust choices, showing the adaptability of BioMON to different scenarios. While achieving competitive performance, BioMON's modular architecture opens avenues for future exploration, promising advancements in few-shot learning for biomedical applications.

References

Ba, J., Hinton, G. E., Mnih, V., Leibo, J. Z., and Ionescu, C. Using fast weights to attend to the recent past. *Advances in neural information processing systems*, 29, 2016.

Barratt, S. On the differentiability of the solution to convex optimization problems. 04 2018.

Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., Bansal, P., Bridge, A. J., Poux, S., Bougueleret, L., and Xenarios, I. Uniprotkb/swiss-prot, the manually annotated section of the uniprot knowledgebase: how to use the entry view. *Plant bioinformatics: methods and protocols*, pp. 23–54, 2016.

Cao, K., Brbic, M., and Leskovec, J. Concept learners for few-shot learning. *arXiv preprint arXiv:2007.07375*, 2020.

Caruana, R., Karampatziakis, N., and Yessenalina, A. An empirical evaluation of supervised learning in high dimensions. pp. 96–103, 01 2008. doi: 10.1145/1390156.1390169.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A closer look at few-shot classification. 04 2019a.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019b.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. 03 2017.

Gould, S., Fernando, B., Cherian, A., Anderson, P., Cruz, R., and Guo, E. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. 07 2016.

Kotia, J., Kotwal, A., Bharti, R., and Mangrulkar, R. Few shot learning for medical imaging. *Machine learning algorithms for industrial applications*, pp. 107–132, 2021.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10657–10665, 2019.

Li, L., Song, D., Ma, R., Qiu, X., and Huang, X. Knn-bert: fine-tuning pre-trained models with knn classifier. *arXiv preprint arXiv:2110.02523*, 2021a.

Li, Z., Wang, H., and Liu, X. A one-dimensional siamese few-shot learning approach for ecg classification under limited data. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 455–458. IEEE, 2021b.

Malisiewicz, T., Gupta, A., and Efros, A. A. Ensemble of exemplar-svms for object detection and beyond. In *2011 International Conference on Computer Vision*, pp. 89–96, 2011. doi: 10.1109/ICCV.2011.6126229.

Ning, R., Chen, C. P., and Zhang, T. Cross-subject eeg emotion recognition using domain adaptive few-shot learning networks. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1468–1472. IEEE, 2021.

Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., and Humenberger, M. R2d2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019.

Singh, R., Bharti, V., Purohit, V., Kumar, A., Singh, A. K., and Singh, S. K. Metamed: Few-shot medical image classification using gradient-based meta-learning. *Pattern Recognition*, 120:108111, 2021.

Snell, Swersky, Z. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. 06 2016.

Xu, C., Shen, J., and Du, X. A method of few-shot network intrusion detection based on meta-learning framework. *IEEE Transactions on Information Forensics and Security*, 15:3540–3552, 2020. doi: 10.1109/TIFS.2020.2991876.

Appendix

For our appendix, we provide some interesting additional information and graphs, to provide better insights into our work. We are not able to include the whole set of experiments conducted, but we make all our graphs available in our repository[2].

5.1. Supporting Material

Our code, report, and slides are available in the following GitHub repository. We provide scripts that reproduce and regenerate the presented results, as well as information about how to use them. The benchmark datasets are publicly available. We want to thank the Deep Learning in Biomedicine course of EPFL for providing credits for Google Cloud.

5.2. Classifiers

Here, we briefly define the classifiers used in the paper.

Support Vector Machines (SVM). Our SVM classifier aims to find the optimal separating hyperplane for our multi-class

[2] https://github.com/MChatzakis/BioMON/blob/main/bioMON.ipynb

classification task. We use a linear kernel, as it achieves the best results compared to alternative kernels, with standard regularization.

Logistic Regression (LR). We use a multi-class Logistic Regression classifier with L2-regularization that prevents overfitting.

Decision Trees (DT) and Random Forest (RF). We use both simple Decision Trees and Random Forests of various estimator values.

Naive Bayes (NB). We use Naive Bayes, a probabilistic model based on Bayes' theorem, which operates by assuming independence between features.

K-Nearest Neighbors (KNN). KNN performs instance-based learning where classification is based on the majority class of k-nearest neighbors.

Gaussian Mixture Model (GMM). GMM is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. It is mainly used for clustering, but we employ a variation of the model suitable for classification, by clustering in groups equal to the number of classes.

Deep Learning Classifiers For Deep Learning classification, we implement a multi-layer perception neural network (MLP). Our network has a configurable number and size of the hidden dimensions and uses the $RELU$ activation function for each layer.

5.3. Additional Experiments

Swissprot MLP Tuning Figure 9 presents the MLP tuning for Swissprot. Following a methodology akin to the Tabula Muris experiments, we explored various configurations by adjusting layer size, the number of layers, and training epochs. The hyperparameters were consistently set, maintaining a learning rate of 0.001 with a weight decay of 0.000001.The observed pattern of accuracy development mirrors that of Tabula Muris, albeit with generally lower accuracy levels in the Swissprot dataset.

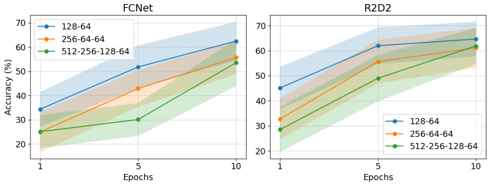

Figure 9. Swissprot MLP tuning

MLP Training Curves Figure 10 presents the per-epoch training summary of BioMON when the different variations of MLPs are utilized as classifiers, for Tabula Muris.

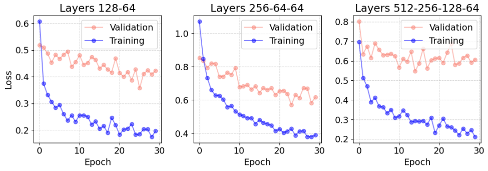

Figure 10. BioMON training curves for MLP classifiers for FCNet and Tabula Muris

Classifier Performance for FCNet Here, we provide the performance of the classifiers for FCNet. The results for Tabula Muris are presented in figure 11 while for Swissprot in figure 12.

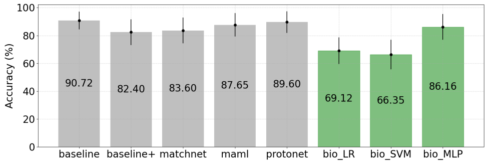

Figure 11. Classifier accuracy for Tabula Muris with FCNet

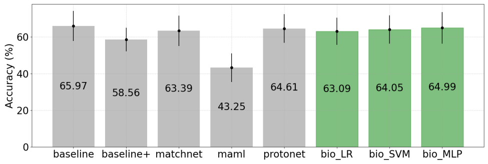

Figure 12. Classifier accuracy comparison for Swissprot with FCNet

Classifier Fitting Times Here, we provide the fitting times of the classifiers used by BioMON. Figures 13 and 14 present the results for Tabula Muris, and figures 15 and 16 present the results for Swissprot. Notably, our top-performing model, MLP, exhibits a relatively higher time consumption compared to other classifiers in both datasets.

Algorithm Accuracy Comparison with FCNet Here, we provide the results of the few-shot algorithm comparisons with FCNet. The results for Tabula Muris are presented in figure 17, while the results for Swissprot are gathered in figure 18.

Algorithm Fitting Times Here we provide the total training time for each one of the few-shot algorithms we compared. Figures 19 and 20 present the total training time for Tabula Muris, while figures 21 and 22 for Swissprot. In a broader benchmark algorithm fitting time analysis, MLP consistently shows higher time requirements in contrast to alternative methods. However, our second-best model displays a more favorable trade-off between performance and time consumption. It's worth mentioning that training MLP involves a more extensive set of hyperparameters compared to LR, contributing to its higher time requirements.

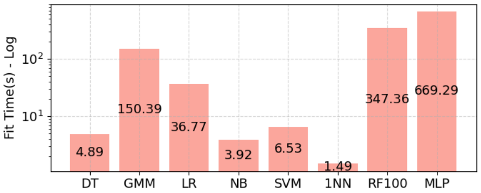

Figure 13. Classifier fitting time for Tabula Muris with R2D2

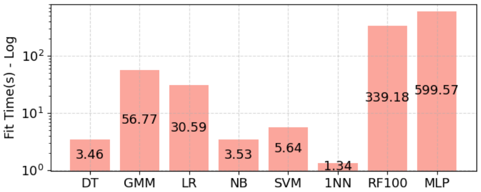

Figure 14. Classifier fitting time for Tabula Muris with FCNet

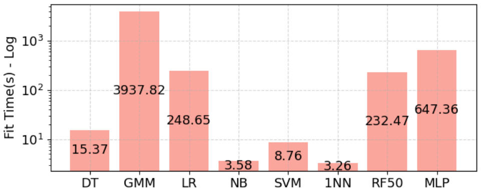

Figure 15. Classifier fitting time for Swissprot with R2D2

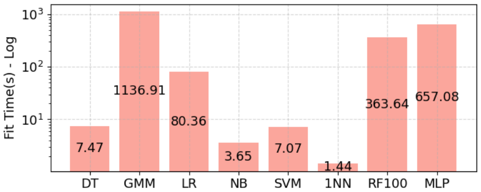

Figure 16. Classifier fitting time for Tabula Muris with FCNet

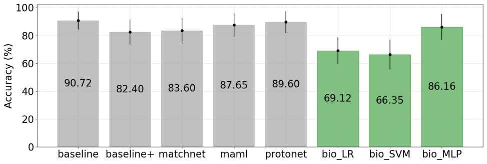

Figure 17. Few-shot algorithms for Tabula Muris using FCNet

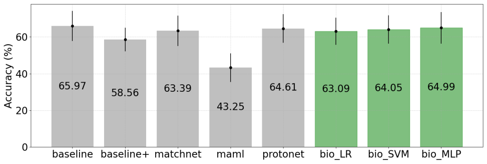

Figure 18. Few-shot algorithms for Swissprot using FCNet

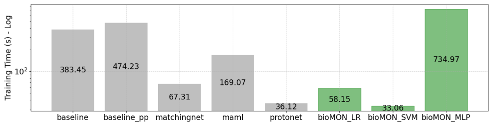

Figure 19. Training time for Tabula Muris with R2D2

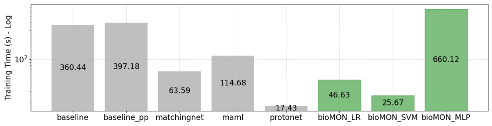

Figure 20. Training time for Tabula Muris with FCNet

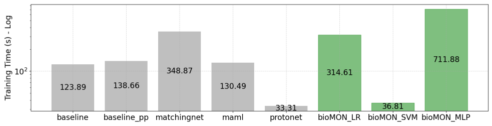

Figure 21. Training time for Swissprot with R2D2

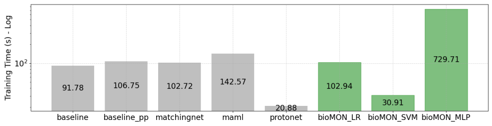

Figure 22. Training time for Swissprot with FCNet