
Enhancing Few-Shot Learning in Biomedicine: Benchmarking Relation Network

Ozan Güven Arnaud Poletto

Abstract

This report investigates the application of a Relation Network, a deep learning model designed for comparing data instances in few-shot learning scenarios, in the field of biomedicine. Our work involves customizing and optimizing the Relation Network to analyze biomedical data with limited examples. We aim to compare the efficacy of the Relation Network against various other few-shot learning algorithms. This comparison is pivotal in understanding the model's relative strengths and areas for improvement within biomedical data analysis.

1. Introduction

The field of biomedicine frequently encounters the challenge of deriving insights from limited data. This challenge is particularly pronounced due to the rarity of specific medical conditions and the practical constraints in data collection. Few-shot learning, which enables learning from minimal data, is thus highly relevant in this context. The Relation Network (RN) (Sung et al., 2018), a model designed for few-shot learning scenarios, is particularly promising for biomedical applications. It excels in comparing small data instances through a learned deep distance metric.

In this report, we focus on adapting and optimizing RN for biomedical data analysis. Our primary objective is to benchmark RN against other few-shot learning algorithms. This comparative study aims to evaluate the model's effectiveness and identify areas for potential improvement, thereby contributing to the advancement of few-shot learning applications in biomedicine.

To this end, we utilize two key datasets in our analysis: Tabula Muris (TM) (Tabula Muris Consortium, 2020) and SwissProt (SP) (Consortium, 2022). TM is a comprehensive resource that provides single-cell gene expression data across multiple tissues. SP, on the other hand, is a curated protein sequence database that includes a wealth of gene function annotations, presenting a different set of challenges and opportunities. The use of these datasets allows us to thoroughly evaluate the performance and applicability of RN against other few-shot learning algorithms in diverse and realistic biomedical scenarios.

2. Related Works

This section reviews some of the key few-shot learning algorithms that have influenced our study, particularly focusing on their application and relevance in the biomedical domain.

Meta-learning algorithms, or learning to learn, form a significant part of few-shot learning research. One prominent algorithm in this category is Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017), which seeks to find model parameters that can be quickly adapted to new tasks with minimal training data.

Another notable method is Prototypical Networks (Snell et al., 2017) which represent classes in a learned metric space, making classification decisions based on the distance to class prototypes. This approach is particularly relevant in classifying data where class representations can be derived from few examples.

Matching Networks (Vinyals et al., 2017) employ an attention mechanism to weigh the importance of support set examples when classifying new instances. This method's ability to adaptively focus on relevant examples makes it a strong candidate where contextual data relevance is key.

Baseline and Baseline++ (Chen et al., 2020) serve as simpler yet effective points of comparison. These methods involve pre-training a feature extractor and then performing transfer learning, offering a straightforward approach to few-shot learning.

The comparison of these diverse few-shot learning techniques, including our RN, in the context of biomedicine forms the cornerstone of our study. By evaluating their performance on biomedical datasets, we aim to contribute meaningful insights into the potential of these methods in handling the unique challenges of this domain.

3. Methodology

3.1. Data Preprocessing and Preparation

Our study utilizes input features of biomedical data from TM and SP datasets. Each data instance is represented by a vector of size 1280 for SP, and 2866 for TM. Due to the nature of these datasets, additional preprocessing steps were minimal, focusing primarily on utilizing these pre-existing

representations for training RN.

3.2. Network Architecture

RN, as illustrated in Figure A.1, operates on the principle of learning to compare. It consists of two primary components: an embedding module f_φ and a relation module g_ϕ . The embedding module processes each input data instance, both from the support set and the query set, and transforms them into an embedding space. This module is typically composed of convolutional or fully connected layers, depending on the nature of the input data. For K -shots, where $K > 1$, a class representative is computed by aggregating the embedding of each element in the class, represented by module \mathcal{A} in Figure A.1. For each query instance z , we concatenate it with each representative $x^{(j)}$, illustrated by module \mathcal{C} in Figure A.1, and compute their relational score s_j using g_ϕ . This score reflects the similarity between the pair, guiding the classification decision. In other words, for input query z , the relation score s_j associated with class j is given by:

$$x^{(j)} = \mathcal{A}(f_\varphi(x_1^{(j)}), f_\varphi(x_2^{(j)}), \dots, f_\varphi(x_K^{(j)})) \quad (1)$$

$$s_j = g_\phi(\mathcal{C}(x^{(j)}, f_\varphi(z))), \quad (2)$$

where $x^{(j)}$ is the class representative for class j ($x_i^{(j)}$, $z \in \mathbb{R}^N$, where $N = 1280$ for SP and $N = 2866$ for TM, $i = 1, \dots, K$).

4. Experimental Setup

4.1. Network Adaption

Given the nature of our input data as 1-dimensional vectors rather than images, we adapt the original RN architecture to suit our needs. The embedding module f_φ consists of a fully connected network comprising one layer of size 16 accompanied by a BatchNorm1d, a ReLU activation function, and a Dropout module. For the aggregation module \mathcal{A} , we use the sum of class embeddings, as done in (Sung et al., 2018).

For the concatenation module \mathcal{C} , our approach involves stacking vector embeddings together, which results in the effective doubling of the output size of the embedding module.

The relation module g_ϕ is composed of three layers of sizes 128, 64 and 32 for SP and one layer of size 128 for TM. Each linear layer is followed by a BatchNorm1d and a ReLU activation function. The final classification layer has a single output neuron passed through a sigmoid function.

4.2. Hyperparameter Tuning and Training Procedure

The hyperparameter tuning for RN is conducted through a comprehensive grid search strategy, exploring a wide range

of parameter combinations for the modules in a 5-way 5-shot setting serving as basis. It involves iterating over a predefined set of hyperparameters, including module layer sizes, learning rates, embedding module weight decays and dropout rates. This is executed separately for each dataset to tailor the model specifically to their respective characteristics.

We vary the number of layers for the embedding module to 4, 2, and 1, with dimensions of 128, 64, and 16 respectively. The relation module’s layer sizes is also adjusted to 6, 3, and 1, with dimensions decreasing progressively, similar to an encoder. Learning rates are tested at 10^{-2} , 10^{-3} , and 10^{-4} . Additionally, we experiment with different levels of regularization for the embedding module, including varying weight decays at 0 and 10^{-3} , and dropout rates at 0, 0.1, and 0.2 to balance model fit and generalization. This comprehensive tuning approach leads to the identification of the set of parameters described in 4.1 and below.

Following this tuning procedure, the model is trained end-to-end using episodic learning for 60 epochs. We employ the Mean Squared Error (MSE) loss function, as used in (Sung et al., 2018) and use the Adam optimizer with an initial learning rate of 10^{-3} and a batch size of 4. The embedding module f_φ is regularized with a weight decay of 10^{-3} and includes a dropout rate of 0, while the relation module g_ϕ has no dropout nor weight decay.

5. Results

RN’s performance is benchmarked against several key few-shot learning algorithms, specifically Prototypical Network, Matching Network, MAML, Baseline and Baseline++ methods, as cited in the related works section. Table 1 presents the results for various scenarios, including 5-way 1-shot, 5-way 5-shot, and 20-way 5-shot configurations, with accuracy serving as the principal metric for evaluation.

In our comparative study, RN exhibits a competitive stance among few-shot learning models, demonstrating reasonable accuracy in both 5-way and 20-way tasks. Notably, the Prototypical Network outperforms other models in most scenarios, particularly in the 5-way, 1-shot setting, suggesting its robustness in scenarios with less data availability per class. RN, while not leading, shows its merits in 5-way for both 1- and 5-shot settings on TM, indicating its potential for rapid learning from sparse data.

However, in the more complex 20-way, 5-shot classification on TM, RN’s performance trails, suggesting that the model might face challenges in differentiating among a larger set of classes. Yet, it performs better than the Matching Network on SP. The observed performance variability underscores the notion that model selection in few-shot learning should be contextually driven, tailored to the dataset’s intrinsic

properties and the specificities of the task.

The results also highlight the notable efficacy of simpler models like Baseline and Baseline++. Their effectiveness can be attributed to their straightforward training procedure using pre-trained weights, which can be an advantage in scenarios where the data does not require highly complex architectures to capture the underlying patterns.

5.1. Ablation Study

In order to gain deeper insights into the behavior and performance of RN, we conduct an ablation study. The goal is to identify which components are critical for achieving optimal results and which ones can be simplified or replaced without significantly affecting performance.

5.1.1. MODIFYING MODULES COMPLEXITY

We conduct experiments to assess the impact of varying the complexity of the embedding module f_φ and the relation module g_ϕ .

Two versions of f_φ are defined: a more complex f_φ^+ with four linear layers of size 128, and a simplified f_φ^- with a single layer of size 16. Similarly for g_ϕ , the module g_ϕ^+ consists of three linear layers of sizes 128, 64, and 32, followed by a classification layer. The less complex configuration g_ϕ^- is a single linear layer of size 16, also followed by a classification layer. These variations allow us to explore all combinations of embedding and relation modules.

As indicated in Table 2, enhancing the complexity of f_φ leads to a decrease in accuracy by approximately 3% in most scenarios. Altering g_ϕ 's complexity, however, does not significantly impact performance. This outcome suggests that the embedding module plays a pivotal role in class discrimination. It also indicates that the relation module may adapt to different embedding complexities.

In a further test, we replace the embedding module with an identity module f_φ^0 to examine its necessity. With TM, the removal of f_φ leads to a 3% accuracy decrease compared to the base model. Conversely, for SP, this change results in an approximate 4% accuracy increase. This contrast highlights the potential variability in the quality of initial embeddings across datasets. SP might inherently possess more discriminative embeddings, explaining this divergent trend.

5.1.2. MODIFYING AGGREGATION AND CONCATENATION MODULES

We first alter the aggregation module \mathcal{A} to compute the mean instead of the sum when obtaining class representatives. The primary motivation behind this change is to ensure that the magnitude of class representatives remains stable and

independent of the number of support examples K . When applied to SP, this adjustment resulted in a 1% increase in accuracy, and a 1% decrease for TM. This outcome suggests that mean aggregation, while stabilizing the magnitude of class representations, does not necessarily lead to a change in accuracy.

Building upon this insight, we further modified the concatenation module \mathcal{C} by changing the stacking of embeddings to computing the difference between the query embedding and the class representative. This approach still utilized mean aggregation to ensure consistent magnitudes. Remarkably, on SP, this modification leads to an accuracy increase from 51.38% to 55.65%, showcasing a noteworthy improvement. In contrast, on TM, the accuracy decreases from 88.95% to 84.51%. These results indicate that the modified concatenation approach may be a valuable addition.

5.1.3. REPLACING DEEP DISTANCE MODULE BY STANDARD DISTANCE METRICS

In another aspect of our ablation study, we explore the effects of replacing RN's deep distance module with standard distance metrics. Specifically, we implement cosine similarity, $L1$ and $L2$ norms to assess their efficacy compared to the original fully connected layers, leading to a model architecture closer to Prototypical Networks.

The results, as shown in Table 3, reveal interesting patterns. For TM, the use of $L1$ and $L2$ norms yield accuracy improvements to 87.7% and 88.8% respectively, higher than the base model's 86%. This indicates that in this context, simpler distance metrics cannot only replace the more complex deep distance module but also enhance the model's performance. In the case of SP, a similar trend is observed with $L1$ and $L2$ norms.

These findings emphasize the importance of synergy between the embedding and the relation modules. The enhanced performance with simpler metrics implies that a well-optimized embedding module, even without a deep distance metric, can be pivotal in making accurate predictions, underscoring the critical role of each component in RN's architecture.

5.2. Ways and Shots Analysis

We conduct a ways and shots analysis, varying these parameters to compare the performance of our RN with other models. This helped us evaluate each model's adaptability to different data quantities and class diversities. Figure 1 provides a visual comparison of this analysis. RN improves with more shots on SP but plateaus on TM, suggesting that it captures protein sequence features well but struggles with complex and high-dimensional gene expression profiles. The Prototypical Network, however, performs consistently

Model	Dataset	5-Way Acc. (%)		20-Way Acc. (%)
		1-shot	5-shot	5-shot
Prototypical Network	TM	81.15 \pm 11.62	89.20 \pm 7.53	71.80 \pm 4.55
Matching Network	TM	75.11 \pm 13.12	84.32 \pm 9.61	62.18 \pm 5.05
MAML	TM	70.64 \pm 13.95	86.76 \pm 8.66	—
Baseline	TM	79.80 \pm 11.73	89.65 \pm 7.39	74.60 \pm 4.35
Baseline++	TM	67.50 \pm 12.70	81.86 \pm 9.38	56.65 \pm 4.59
Relation Network	TM	77.55 \pm 11.57	85.95 \pm 8.88	64.92 \pm 4.92
Prototypical Network	SP	52.96 \pm 11.77	61.09 \pm 7.66	56.08 \pm 4.67
Matching Network	SP	53.43 \pm 11.29	64.43 \pm 7.47	27.81 \pm 4.70
MAML	SP	51.63 \pm 12.71	42.07 \pm 7.43	—
Baseline	SP	54.71 \pm 11.94	64.96 \pm 7.70	—
Baseline++	SP	54.80 \pm 11.29	59.35 \pm 7.68	—
Relation Network	SP	51.08 \pm 12.17	51.38 \pm 8.73	37.98 \pm 4.78

Table 1. Comparative results with standard deviations of the mean test accuracy between the various algorithms and RN.

Dataset	Combinations					
	$f_{\varphi}^{+}, g_{\varphi}^{+}$	$f_{\varphi}^{+}, g_{\varphi}^{-}$	$f_{\varphi}^{-}, g_{\varphi}^{+}$	$f_{\varphi}^{-}, g_{\varphi}^{-}$	$f_{\varphi}^0, g_{\varphi}^{+}$	$f_{\varphi}^0, g_{\varphi}^{-}$
TM	83.44 \pm 9.90	81.14 \pm 9.81	85.06 \pm 9.22	85.95 \pm 8.88*	78.81 \pm 11.46	82.58 \pm 11.04
SP	49.25 \pm 8.61	47.80 \pm 7.85	51.38 \pm 8.73*	51.40 \pm 7.43	54.50 \pm 7.50	56.63 \pm 7.72

Table 2. Ablation results on modules complexity reduction. Configuration with (*) represents the base model for each dataset.

Dataset	Distances		
	L1	L2	Cosine
TM	87.76 \pm 7.94	88.75 \pm 8.24	72.45 \pm 12.32
SP	53.70 \pm 7.75	54.17 \pm 7.12	53.98 \pm 8.27

Table 3. Ablation results on modules complexity reduction.

across both datasets.

In the ways analysis using TM, RN’s accuracy diminishes noticeably as the number of classes increases, more so than the Prototypical Network. This suggests that RN may have limitations in differentiating among a greater variety of classes, a challenge that the Prototypical Network seems to handle with more resilience, possibly due to its efficient metric space representation.

In light of the observed performance variations across different ways and shots, a pertinent consideration is the potential advantage of model tuning tailored to each specific way and shot configuration. Particularly for scenarios with a larger number of ways, customizing the model—such as employing a deeper embedding module—may yield improved results.

6. Conclusion

This report demonstrates RN’s potential and challenges in biomedical few-shot learning. Our analysis reveals RN’s competitive performance in certain scenarios, yet highlights

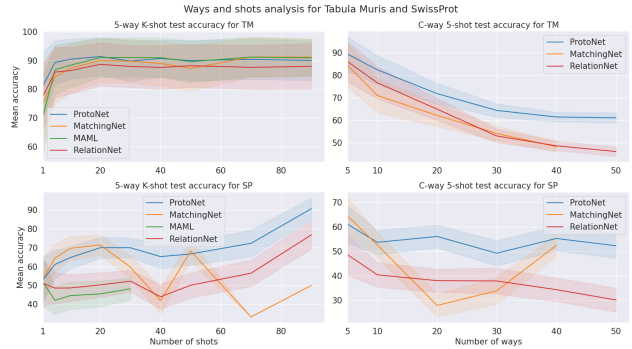


Figure 1. The effect of different ways and shots on test performance (with standard deviation).

its limitations in tasks with more classes. The ablation study underlines the RN’s sensitivity to architectural changes, where simplifications often lead to improved accuracy, indicating that less complex models can be effective in certain contexts. The model’s performance varies with different class diversities and data quantities, suggesting the need for tailored tuning for specific scenarios. These insights demonstrate RN’s potential in few-shot learning while underscoring the importance of adapting model architecture to the specific challenges of biomedical data.

References

- Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A closer look at few-shot classification, 2020.
- Consortium, T. U. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, 51(D1): D523–D531, 11 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac1052. URL <https://doi.org/10.1093/nar/gkac1052>.
- Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.
- Snell, J., Swersky, K., and Zemel, R. S. Prototypical networks for few-shot learning, 2017.
- Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning, 2018.
- Tabula Muris Consortium. A single-cell transcriptomic atlas characterizes ageing tissues in the mouse. *Nature*, 583 (7817):590–595, July 2020. URL <https://pubmed.ncbi.nlm.nih.gov/32669714/>.
- Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning, 2017.

Appendix A. Model

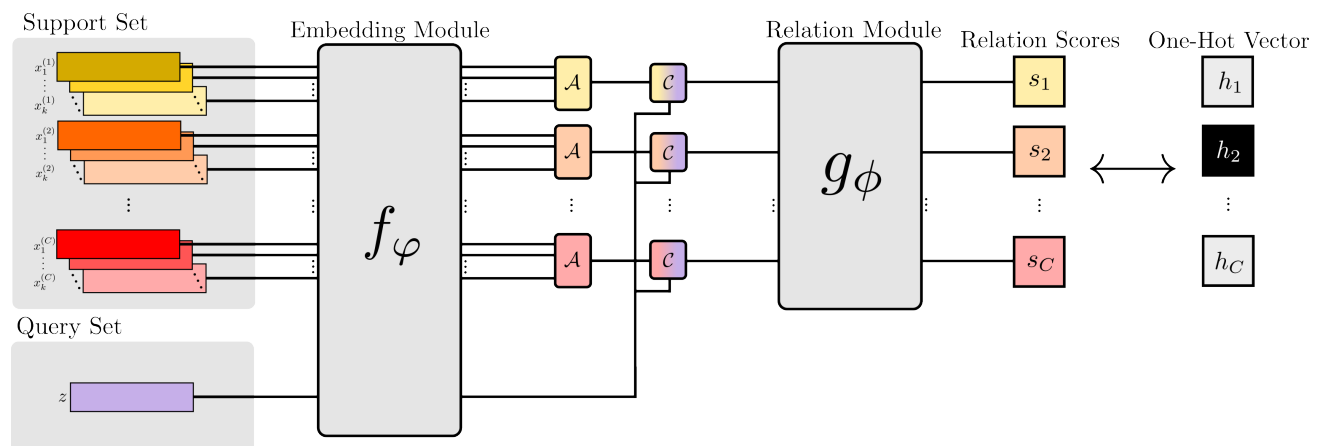


Figure A.1. Relation Network architecture for a C -way K -shot problem with 1 query example.