Deep Hierarchical Embedding for Simultaneous Modeling of GPCR Proteins in a Unified Metric Space

Taeheon Lee
Seoul National University
Sangseon Lee
Seoul National University
Minji Kang
Stanford University
Sun Kim (sunkim.bioinfo@snu.ac.kr)
Seoul National University

Research Article

Keywords: Deep Hierarchical Embedding, GPCR Proteins, Unified Metric Space, Simultaneous Modeling

Posted Date: January 29th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-154212/v1

Deep Hierarchical Embedding for Simultaneous Modeling of GPCR Proteins in a Unified Metric Space

Taeheon Lee[1], Sangseon Lee[2], Minji Kang[3], and Sun Kim[4,5,6,7,*]

[1]Looxid Labs, Seoul, 06099, Republic of Korea
[2]BK21 FOUR Intelligence Computing, Seoul National University, Seoul, 08826, Republic of Korea
[3]Department of Computer Science, Stanford University, Stanford, CA, 94305, United States of America
[4]Bioinformatics Institute, Seoul National University, Seoul, 08826, Republic of Korea
[5]Department of Computer Science and Engineering, Seoul National University, Seoul, 08826, Republic of Korea
[6]Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul, 08826, Republic of Korea
[7]Department of Computer Science and Engineering, Institute of Engineering Research, Seoul National University, Seoul, 08826, Republic of Korea
[*]sunkim.bioinfo@snu.ac.kr

ABSTRACT

GPCR proteins belong to diverse families of proteins that are defined at multiple hierarchical levels. Inspecting relationships between GPCR proteins on the hierarchical structure is important, since characteristics of the protein can be inferred from proteins in similar hierarchical information. However, modeling of GPCR families has been performed separately at each level. Relationships between GPCR proteins are ignored in these approaches as they process the information in the proteins with several disconnected models. In this study, we propose a deep learning model to simultaneously learn representations of GPCR family hierarchy from the protein sequences with a unified single model. Novel loss term based on metric learning is introduced to incorporate hierarchical relations between proteins. We tested our approach using a public GPCR sequence dataset. Metric distances in the deep feature space corresponded to the hierarchical family relation between GPCR proteins. Furthermore, we demonstrated that further downstream tasks, like phylogenetic reconstruction and motif discovery, are feasible in the constructed embedding space. These results show that hierarchical relations between sequences were successfully captured in both of technical and biological aspects.

Introduction

G protein-coupled receptor (GPCR) is the largest transmembrane protein family and one of the most extensively investigated drug targets[1,2]. GPCR is classified in a hierarchical class structure, represented by family, subfamily and sub-subfamily level classes. This structure was constructed following the sequence similarity and evolutionary histories of the proteins[3]. Analyzing the characteristics of the proteins with respect to this structure lies at the heart of GPCR studies[4] and inspecting relations between GPCR sequences regarding this structure is an important research task for two main reasons. First, properties of the protein are often inferred from the existing proteins that have already been experimentally validated[5]. Second, evolutionary history of the proteins can be revealed from the relations among proteins.

Approaches based on machine learning techniques, such as hierarchical classification and clustering, on the class structure have been widely explored[4,6,7]. These methods have shown successes in modeling GPCR fairly accurately. However, existing methods had to model GPCR at each of the family hierarchies separately[7,8] and inevitably employed series of separate steps to deal with the features in the class structure since the representations used in the methods can hardly reveal unified features across the class hierarchies. As a result, processing complex features throughout the hierarchy levels as a whole is nearly impossible with these representations[9]. In the sense that GPCR class hierarchy was constructed using complex features including phylogenetic traits, ligand types and their functions, these approaches may not provide research opportunities to inspect the relations between these traits throughout the GPCR proteins[10]. To be specific, robust inspections on proteins, for example comparison between sequence clusters at different hierarchy levels, cannot be done with existing methods since comparing models at different hierarchies is not feasible. In this regard, it is important to construct comprehensive representations of GPCR proteins with hierarchical features inclusively incorporated.

In this work, we present a novel method that simultaneously learns and represents complex protein features across the hierarchies. Our end-to-end deep learning network constructs a single embedding space of GPCR sequences where hierarchical sequence information is preserved in terms of metric distances. This embedding function incorporates significant features across all hierarchical levels into a single vector. On the constructed embedding space, distances in the embedding space can be

utilized in analyzing GPCR proteins for several downstream tasks.

In a series of experiments, we showed that the metric space generated by our method successfully reflected hierarchical class structure of GPCR proteins. First, we extensively investigated distances among sequences in the embedding space using cluster analysis techniques. In the distance-based cluster analysis on the embedding space, proteins are grouped well according to the hierarchical class labels of proteins. Second, we showed that phylogenetic trees constructed from the embedding vectors accurately reflected the phylogeny of the proteins. In addition, we showed that biologically significant motifs in the GPCR proteins are well-represented in each cluster. We further investigated how those motifs are generalized or narrowed along different hierarchical levels. Last but not least, experiments on the query search demonstrated that similarity search on proteins can be done fairly efficiently with the embedding vectors.

Related Work

Analyzing biological sequences with deep learning

Recently, deep learning technologies have brought unprecedented advances in analyzing biological sequences. For example, a previous deep learning study proposed deep convolutional layers to recognize protein folds from sequences with competitive accuracy and robustness[11]. Other works showed that biological sequences can be successfully modeled with a shallow architecture with one-layer convolutional layer[12,13]. In these works, each convolutional filter was trained to acquire features that correspond to significant sequence patterns. Especially, DeepFam[13] showed promising results in modeling proteins families with motif features. Although our work is built upon the architecture proposed in DeepFam, the proposed method differs from the previous work in that neural network architecture is significantly expanded to incorporate hierarchical class information into a single model. Furthermore, our work enables the embedding of hierarchical features, which are originated from the protein family hierarchy, onto a single metric space.

Metric learning on deep learning

With the success of deep learning models, more and more works have been proposed to utilize distances in the deep feature space. Training objective in these studies are to learn an effective mapping function of data where similarity between inputs can be directly inferred from the embedding vectors. Neural networks are trained to favor close distances between representations for data points of similar properties and distant representations for dissimilar data. For example, Siamese networks[14] and Triplet networks[15] were introduced in order to make feature vectors for similar data points to be located closely in distances in the proposed deep embedding spaces. Similarly, center loss was introduced to construct compact intra-class representations and separable inter-class representations[16]. In center loss, mean embedding vectors of data classes are employed as reference vectors to guide training. Our work adopted center loss to learn compact and separable embedding function according to protein family information.

In deep metric learning for sequence analyses, components of training phases are designed to grant biologically significant meanings to distances in the embedding space. For instances, a Siamese neural network for biological sequences was introduced, where alignment distances between sequences can be directly estimated from the embedding vectors of the given sequences[17]. Moreover, an embedding function based on long short-term memory was suggested to incorporate structural similarities of proteins during training[18]. While previous studies learned metrics for single level information of biological sequences, up to our knowledge, our work is a first work to model hierarchical relations between protein sequences into a single metric space.

Methods

Neural network architecture

The overall structure of the model is extended and modified from DeepFam[13]. Rather than merely adopting the previous work, however, we introduce novel components on the architecture, Embedding layer and Multi-branch classifier, to effectively build a metric space. Overall architecture of the neural network is illustrated in Figure 1.

Feature extractor with CNN. In DeepFam, variable length convolutional filters were used with 1-max pooling layer[13]. In such setting, output activation value from each convolutional filter encodes information on whether significant motif patterns are presented in the input sequences. This architecture was demonstrated to be successful in capturing motifs from biological sequences[12,13]. Especially, DeepFam was successful in detecting motifs of variable lengths. In this work, we also adopted this 1-max pooling architecture since GPCR families are characterized by highly preserved sequence regions[19]. After applying 1-max pooling, the resulting values in the convolutional layer is concatenated into an one-dimensional representation, which can be regarded as a list of existence values for learned motifs.

Embedding layer. A linear layer that comes after the convolutional layer is to project outputs from the convolutional layer onto a low-dimensional vector space. Projected vector will be used as an embedding vector of an input protein sequence. To simplify the explanation, we assume embedding vectors lie in a d-dimensional space. After the linear transformation follows an

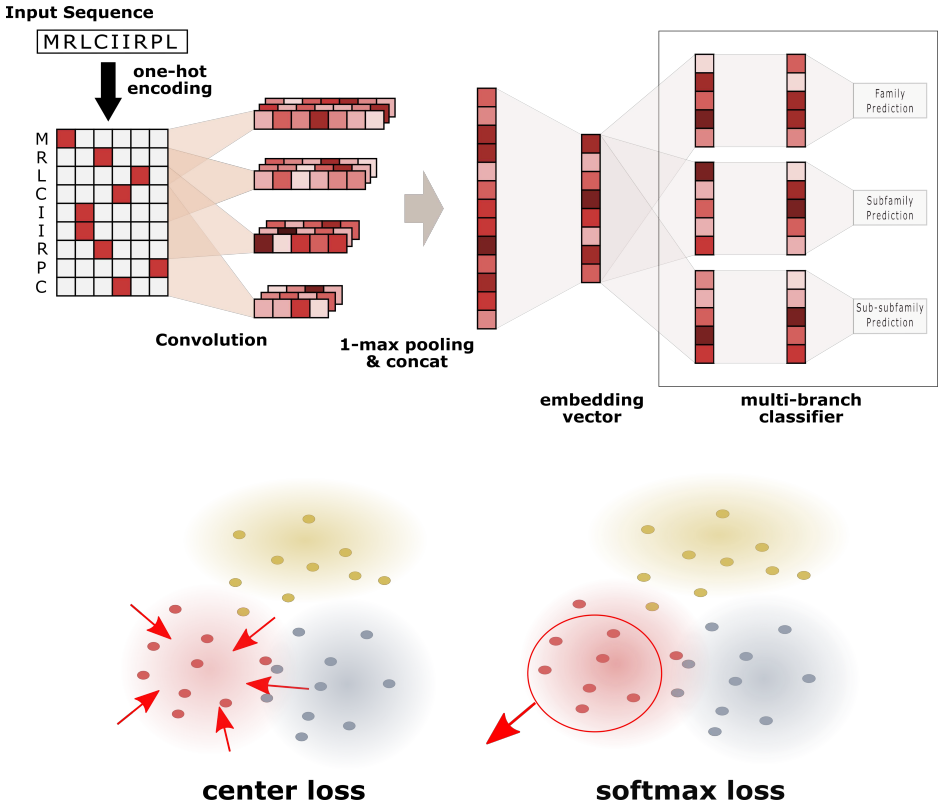


Figure 1. Neural network architecture and loss function.

L_2 normalization operator on resulting vectors to keep the distances between embeddings from exploding. Thus, we restricted the vector representation of each sequence to lie in a d-dimensional hyper-sphere.

Multi-branch classifier. Three branches of Multilayer Perceptron (MLP) classifier are attached to the embedding layer. Techniques of using branches in a single neural network, with each branch dedicated to a domain-specific task, were proposed in Multi-Domain Network (MDNet)[20] to construct shared features across the multiple domains. In our architecture, three branches correspond to each hierarchical level: class, family and subfamily level. Under this multi-branch architecture, gradients from these classifiers will guide embedding vectors to incorporate hierarchical information regarding GPCR protein family into a single representation.

Loss function

In our approach, a loss function with respect to a metric space is introduced to construct meaningful distance relations between protein sequences. We first list the notations used in defining the loss terms in Table 1. In the Table, μ_{y_i} is calculated as a mean vector of deep representations of sequences that corresponds to class y_i proteins[16] .

Softmax loss. Cross-entropy loss with a softmax function is frequently used as a loss function for training classifiers. Softmax function is used to generate probability distribution of candidate labels from the output layer of the network. In many cases, softmax function is combined with cross-entropy loss in supervised setting to enforce classifiers to output higher probability on desired labels. This is effective for neural network models in learning separable representations between different classes. Likewise, we employed this function as a part of the loss to empower neural networks to learn separable features in input sequences for each label.

Center loss. Center loss[16] was proposed to complement the softmax loss function. Although softmax loss is practical enough to separate features between classes, it lacks ability to learn compact representations between data that is in the same class[16,17]. To address this, this additional loss term minimizes the distances among data points within a same class. Center loss can be stated in following form:

$$L_C = \sum_{i=1}^{n} \left\| d(x_i) - \mu_{y_i} \right\|_2^2 \tag{1}$$

L	Total loss to be used in training
L_S	Summed softmax loss from three hierarchies
L_{S_i}	Softmax loss from hierarchy level i where $i \in \{cls, fam, sub\}$
L_C	Summed center loss from three hierarchies
L_{C_i}	Center loss from hierarchy level i
$d(x_i)$	Embedding vector in the hidden layer for input sequence x_i
μ_{y_i}	Class center of embedding vectors in deep feature space for class that input x_i belongs to

Table 1. Notations related to loss terms.

With loss from this metric, parameters are updated to make representation of the input data get closer to the mean vector during training.

In exploiting the above loss function, mean vectors should be updated simultaneously with parameters being updated. In the previous work[16], mean vectors are calculated based on the images in mini-batch basis since considering vector representations of the whole dataset, generally comprising of 50K to 200M images for computer vision, is computationally exhaustive. However, the number of sequences in the training data is relatively small, total of 8222 GPCR proteins for our study. Therefore, we updated mean vectors of classes based on the feature vectors from the whole dataset.

Overall loss. Combining cross-entropy loss L_S from the classifiers and center loss for each hierarchy, overall loss function can be stated in the following equation.

$$L = \sum_{i \in S} \omega_i L_{S_i} + \lambda_C \left(\sum_{i \in S} \omega_i L_{C_i} \right) \tag{2}$$

where S denotes the set of hierarchy levels (S = {class, family, sub-family}) and loss function is stated as a weighted sum of losses from each hierarchy. To balance between center loss and softmax loss, λ_C was introduced as a weight for center loss[16].

Training procedure

To effectively incorporate hierarchical information into a single embedding space, we designed a training procedure in three different phases. At each phase of training procedures, we devised our network to focus on loss values from one level, from family-level to sub-subfamily level. In our setting, training phases are ordered following the hierarchical order, from family level to sub-subfamily level. This order ensures neural networks to firstly learn general representations that correspond to family level properties of GPCR families and then acquire fine-grained representations for sub-subfamily level.

Analyzing distances in the embedding space

To assess the quality of distances in the proposed space, we evaluated how well the distances in the embedding space represent the hierarchical class structure of GPCR protein family. Embedding vectors from the proposed method were compared to following methods: (1) model with the same architecture as ours but without using the center loss function (w/o Center Loss), (2) model with the same loss function and feature extractor as in ours but without a multiple branches output layer (w/o Multiple Branch), (3) DeepFam, (4) Autoencoder, (5) MLP classifier, and (6) K-mer frequency vectors (3-mer and 4-mer). For models based on neural networks, activations of neurons at the last hidden layer were used as representations for the sequences.

To be specific, clusters were compared to the class labels at family, subfamily and sub-subfamily levels using adjusted mutual information (AMI) and silhouette scores. In calculating the silhouette scores, real class labels from three hierarchical levels were used as groundtruth. Since silhouette score is related with intra-cluster and inter-cluster distances, silhouette scores under our scheme represent the consistency of distances in embedding space to the real class labels. Correspondence between the hierarchical clustering results and GPCR family labels was evaluated using AMI score. In estimating the score, we used the real label information as ground-truth label information to measure the quality of clustering results. Agglomerative clustering based on ward linkage, a hierarchical clustering algorithm where pairs of clusters with closest distances are merged together, were applied to the embedding vectors until the target number of clusters are obtained. In our experiment, we increased the target number of clusters from 2 to 100 with a step size of 3. AMI scores were measured between class labels and cluster results from each of the target cluster number. We provide detailed explanations on AMI and silhouette scores in the supplementary material.

Analyzing the hierarchical structure in the embedding vectors

Relationship between proteins on the hierarchical class structure is an important aspect to be investigated in GPCR protein studies. We analyzed the overall distance relations between proteins in the embedding space compared to the real class labels.

Afterwards, we further constructed a phylogenetic tree from the embedding vectors in order to demonstrate that phylogenetic analysis is possible with embedding vectors. Here, a phylogenetic tree was constructed using a neighbor-joining clustering method[21].

Discovering motifs from the embedding vectors

As a way of demonstrating the biological significance of the vectors, we performed motif discovery experiments on the deep feature vectors. In these experiments, MUSCLE, a multiple sequence alignment tool[22], was used to detect conserved sequence patterns. Sets of reference proteins were selected from the hierarchical clustering results in the above section where target number of clusters were incremented. On the sequences in the designated clusters, we applied MUSCLE to figure out the highly preserved regions of that cluster. Afterwards, preserved regions from the alignment algorithm were compared to known motifs in the literatures to check if results are consistent with biological knowledge.

Query search on embedding spaces

We conducted a sequence similarity query search on the protein family database as one of the downstream tasks on the proposed embedding space. Distances in the low-dimensional embedding space have long been utilized for effective query search in the high dimensional data. In this setting, similarity between data points is defined in terms of distances in the low-dimensional space[23]. In a similar manner, since deep feature space was trained to express GPCR similarity in terms of metric distances, we directly utilize the euclidean distances between embedding vectors. Our work handles query search in terms of nearest points in the embedding space. In such setting, unlike conventional alignment-based algorithms for sequences, distance calculations on embedding space can be accomplished simply with numerical calculations. We compared the results with Basic Local Alignment Search Tool (BLAST), the *de facto* standard tool for sequence similarity search[24]. Since computation speed has been major limitations of alignment-based tools, we compared the execution time of each algorithm.

Experimental Setup

Dataset Preparation

The GPCR sequences were acquired from the BIAS-PROFS GPCR dataset[7]. In this dataset, sequence labels are annotated in three-level hierarchies, from the family level to the sub-subfamily level. In processing the dataset, classes with fewer than 10 sequences were removed, resulting in 86 sub-subfamilies and 8222 sequences. During the experiment for query search, sequences from the training dataset were utilized as a target database of approximately 6200 sequences. In contrast, query sequences were only retrieved from the test dataset.

Sequence representation

To enable computations on deep learning models, one-hot encoding scheme was adopted, where every amino acid position was represented as a one-hot vector[25,26]. In addition, each sequence was padded with zeros to a fixed length because CNN requires input vectors of the same size. Thus, after converting a sequence into one-hot vectors, we padded each sequence to the length of 1000[13]. As a result of encoding, each sequence was converted into a vector of size $R^{1000\times 20}$. Here, 20 is the alphabet size of amino acid characters.

Hyperparameter setting

Hyperparameters used in the experiments are listed in Table 2. Parameters regarding convolutional layer were identical to those of DeepFam. However, in our work, we did not employ dropout operator since the embedding vectors need to be learned consistently while training. In addition, the dimension of embedding vectors needs to be set in advance. We selected the dimension of embedding vectors using validation dataset according to accuracies on sub-subfamily level classification. Weights for softmax and center loss terms are also set, separately for the family, subfamily and sub-subfamily phase.

Implementation details

The neural network was implemented with Python 3.6.8 and PyTorch 1.1.0[27], an open-source deep learning library. Phylogenetic tree in the result section was drawn with GraPhlAn, an open-source tool for visualizing phylogenetic tree[28]. Dataset was split into train, validation and test datasets with ratios of 0.8, 0.1 and 0.1 respectively for each subfamily class. We selected the best performance model on validation set and performances were measured in 10-fold cross-validation scheme. Only the embedding vectors from the test dataset were used for evaluation and visualization during experiments.

Hyperparameter	Value		
Kernel sizes	8, 12, 16, 20, 24, 28, 32, 36		
Number of filters	256×8		
Dimension of Embedding Vectors	30		
Number of hidden units in classifier	15		
Learning rate	0.001		
L2-regularizer on classifier	0.0005		
Weight on center loss (λ_C)	0.01	0.3	0.5
Center loss weight for fam ($\omega_{C_{cls}}$)	0.8	0.1	0.1
Center loss weight for subfam ($\omega_{C_{fam}}$)	0.15	0.8	0.15
Center loss weight for sub-subfam ($\omega_{C_{sub}}$)	0.05	0.1	0.75
Softmax loss weight for fam ($\omega_{S_{cls}}$)	0.8	0.1	0.1
Softmax loss weight for subfam ($\omega_{S_{fam}}$)	0.15	0.8	0.25
Softmax loss weight for sub-subfam ($\omega_{S_{sub}}$)	0.05	0.1	0.65

Table 2. List of hyperparameters.

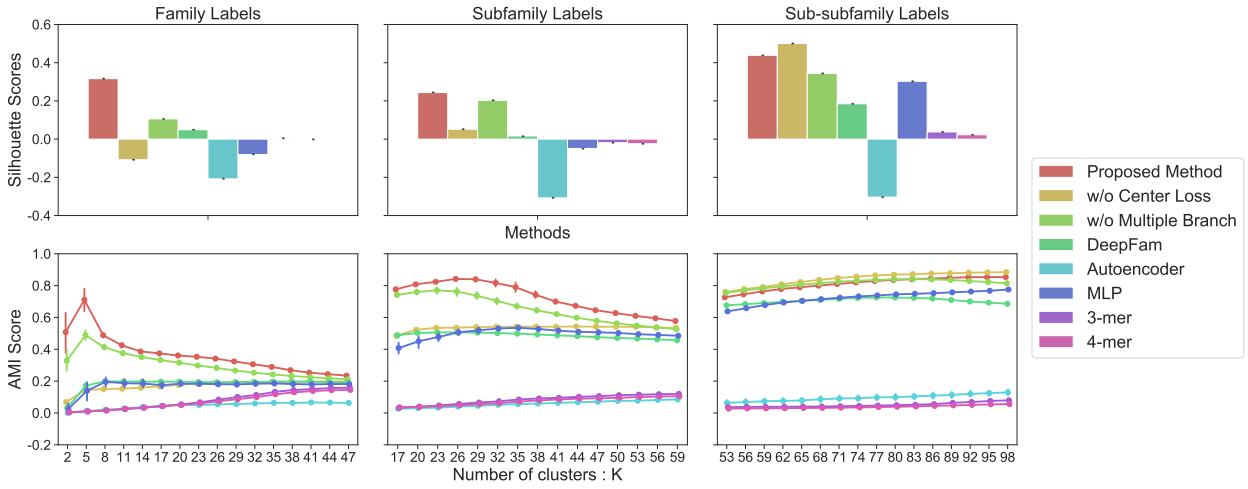


Figure 2. Top: silhouette score of representation vectors regarding true class labels on three hierarchical levels. **Bottom**: adjusted mutual information (AMI) score calculated for class label in each level versus k clusters yielded from hierarchical clustering algorithm.

Results

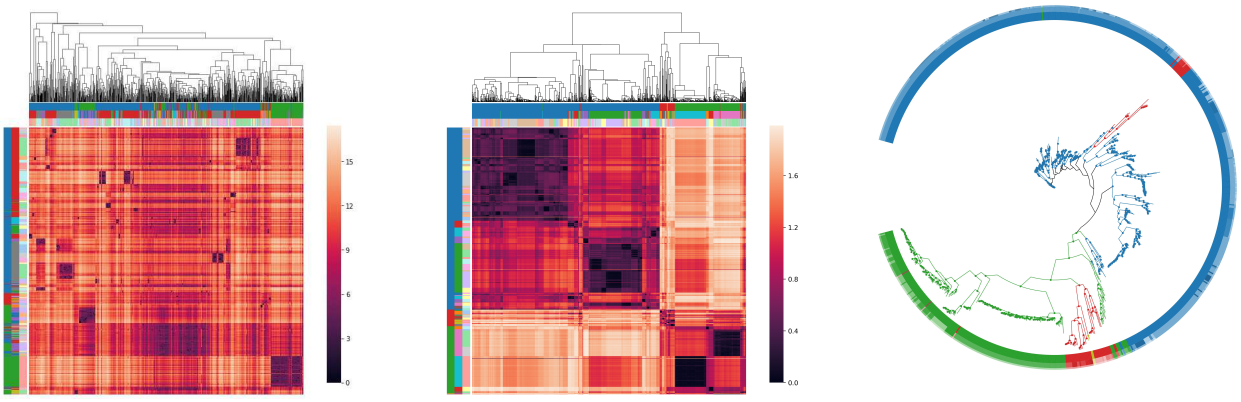
Evaluation on embedding distances

Inspection on silhouette scores

Figure 2 contains the calculated silhouette score from each method. In the figure, only *our model* and *our model w/o multiple branch* are the model equipped with a center loss that constructs metric distances in deep feature space during training. The distinctive result from these models is that they show non-negative silhouette scores in three class levels, whereas vectors from other models present non-negative values only in one level and negative or near-zero scores in other levels. These demonstrate that center loss regarding metric space helped neural network to build effective distance relations for GPCR families.

Vectors from the Autoencoder and k-mer frequencies, that does not use any class information during training, show low scores in all levels. This result shows that label information was a powerful supervisor in training a model. Competitive silhouette scores in all three levels show that the neural network architecture adopted in our model successfully incorporated the information from three labels into the distances in the embedding space.

We now focus on comparison of two models that adopted center loss during parameter updates. Although the model without multiple branch shows positive scores for all three levels, our model shows significantly higher scores. In summary, the proposed two components for metric learning, multiple branches and center loss, have been successful in learning compact representations at all levels.



Distance matrix from DeepFam

Distance matrix from the Proposed Method

Phlyogenetic Tree from the Proposed Method

Figure 3. Embedding vectors on hierarchical structure, distance matrix and phylogenetic tree. **Left**: distance matrix of embedding vectors from DeepFam. **Center**: distance matrix of embedding vectors from the proposed method. **Right**: phylogenetic tree reconstructed from the proposed method.

Inspection on AMI scores

Changes in AMI score for the number of clusters are illustrated in Figure 2. AMI score is maximized when the number of resulting clusters from the algorithm gets closer to the real number of labels in that class level, 5 for family and 37 for subfamily. Interestingly, this can be interpreted as: hierarchical clustering results show best correspondence when a target cluster number is close to the actual number of class labels. This demonstrates that distance relations between embedding vectors match well with hierarchical class structure. Similar to results from the experiments of silhouette scores, some algorithms achieved comparable AMI scores in the sub-subfamily level clustering evaluation. However, our model shows notably higher score in the family and subfamily levels. In fact, our model is the only method that generates the maximum AMI score higher than 0.7 for all class levels. This again supports that our model was successful in incorporating information from all three class levels into a single unified embedding vector. In addition, comparison between results from our model and our model without center loss demonstrates that center loss in our approach makes our embedding more compactly represented.

Analysis on the hierarchical structure

Overall distance structures in the embedding space

In Figure 3, distance matrix was generated using pairwise euclidean distances between the embedding vectors. Protein sequence vectors were clustered using a hierarchical clustering algorithm, Unweighted pair group method with arithmetic mean (UPGMA)[29]. To visualize the hierarchical clustering result, columns were ordered based on the clustering results whereas rows were sorted according to family, subfamily, sub-subfamily class labels. Clustering results are demonstrated with dendrogram in the figure. Each of the three-line color bands on the columns and rows corresponds to the hierarchical class label of each sequence. Thus, correspondence between clustering results and true class information can be revealed by comparing three-line color bands on columns and rows.

It can be observed that hierarchical relations between sequences are apparent in the embedding space from the proposed method. Overall pairwise distances between the sequences in the heat map show that embedding vectors clearly captured the three distinctive distance relations among GPCR sequences, the brightest one ($1.6 \sim$), the middle-range one ($1.2 \sim 1.6$) and the darkest one (~ 0.5). These distinctions in color level make separations between data points more clearly visible. Comparing boundary regions with true label information represented with color bands on columns and rows, color distinctions correctly correspond to the label information of sequences. For representations from DeepFam, although there exists local clusters of proteins, no hierarchical structure between distances is observed. Furthermore, hierarchical clustering results do not match with real class hierarchies in DeepFam. This demonstrates that even though DeepFam modeled sub-subfamily classes in GPCR protein families fairly well[13], hierarchical structure of the proteins was not properly incorporated. In contrast, through employing center loss and multiple branches, the proposed method constructed distinctive hierarchical relations successfully with distances in the embedding space. We further provide t-SNE visualizations[30] for the proposed embedding vectors in the supplementary material.

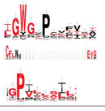
	Cluster 3_1	Cluster 5_4	Cluster 55_16	Cluster 55_48
Target GPCR class	Family A	Family B	Sub-subfamily Melanocortin	Sub-subfamily Traceamine
# Total sequences	519	46	43	14
# Sequences in target class	507	44	41	14
Demonstration of Extracted Motifs				

Figure 4. Discovered motif logos for each cluster on phylogenetic tree generated from hierarchical clustering results. Cluster information as well as information on the sequences included in the cluster is provided. Cluster 3-1 denotes that this cluster is first cluster from the hierarchical clustering when the target number of cluster is 3.

Phylogenetic tree reconstruction

Phylogenetic tree from the embedding vectors is drawn on Figure 3. In the tree, each family-level label is represented with a distinct color. A branch in the tree was assigned a color corresponding to the family when more than 70% of the leaves in the branch belong to a specific family. In addition, as we have done in generating the distance matrix, additional figure of overlaying three-layer color rings with colors corresponded to each class in the hierarchy. In the phylogenetic trees, sequences belonging to the same family labels were clustered in close positions. In the second and the third rings that represent subfamily and sub-subfamily respectively, sequences belonging to subfamily and sub-subfamily were also positioned in close locations in the tree. In sum, we created a single embedding space that can be used to construct a phylogenetic tree based on the single embedding space, grouping GPCR sequences closely at family, subfamily and sub-subfamily levels.

Motif discovery

To visualize the discovered motifs, WebLogo[31], a web-based visualization tool, was used in Figure 4. From the clustering results with a target cluster number of three, conserved sequences of *DRY* and *NSxxNPxxY* were found to be distinctive features in the first cluster. This cluster had 519 sequences, of which 507 sequences belonged to Family A of GPCR protein family. In fact, the discovered motifs in these clusters are the most characterizable sequence features shown in family A or Rhodopsin GPCR family[1,32,33]. Unlike the first cluster, however, alignments on the other clusters does not reveal the conserved sequence of *DRY*. This corresponds to our knowledge that *DRY* and *NSxxNPxxY* are distinctive features for Rhodopsin-like GPCR proteins. Likewise, other significant motifs were found from other clusters too. On the fourth cluster among five clusters in the results, conserved sequences of *LIGWG*, *GPVLASLL* and *CFLxxEVQ* were discovered. These sequences belong to the conserved regions in the transmembrane structures of family B or Secretin receptor family of GPCR family[34]. Indeed, this cluster consists of 46 sequences where 44 of them belonged to family B GPCR proteins. At deeper hierarchical levels, *RKAAKTLG* and *FKQLHXPTN* were found to be conserved in the 48th cluster among 55 clusters. These features are known to be the representative motifs in the Traceamine sub-subfamily that belongs to family A of GPCR proteins. This is consistent with the fact that all the sequences in the cluster are from Traceamine sub-subfamily.

Query search speed on embedding space

We measured execution time of similarity search algorithm. Firstly, preparing database for the proposed method requires embedding the whole sequences in the training dataset through deep learning model, which is composed of nearly 6300 sequences. Second, time spent for searching through the database should be measured. Thus we compared execution time on database construction and database search. For the proposed model, database construction took approximately 12.7 seconds. In contrast, for BLAST, constructing database using *makeblastdb* command took 368 ms. For similarity search on embedding space, it took approximately 19 ms on average for single sequence when database is comprised of 6300 sequences. This includes time spent for representing query sequence in terms of embedding vector from the neural network. In contrast, BLAST took 266 ms on average.

Constructing database in our approach is slower than that of conventional alignment-based approach, since the proposed method requires training of neural networks with whole training sequences. However, once embeddings of the sequences in the database is constructed, we don't need to repeat this process. Thus, we believe slow execution time in constructing database might not be a bottleneck in similarity search. For execution time spent on similarity search, the proposed method only requires

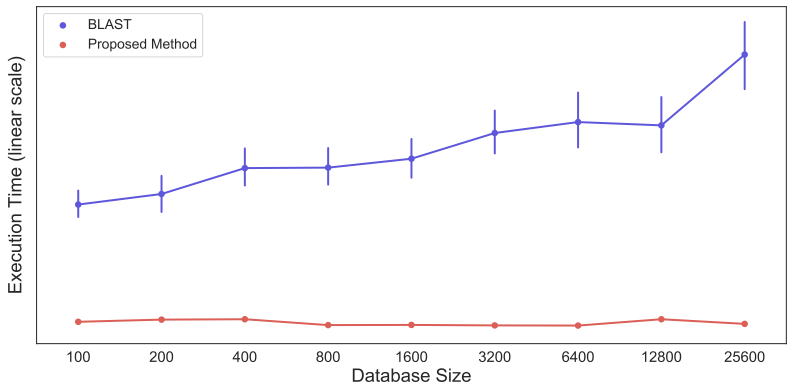


Figure 5. Execution time measured with varying the size of database.

approximately 7.1% of time compared to BLAST.

To assess the scalability of similarity search when the size of the database grows, we measured the execution time with varying the size of database. BLAST shows linearly increasing execution time with the growth of database (Figure 5). However, similarity search on the embedding vectors from the proposed shows consistent execution time. We believe that since the embedding vectors lie in a low-dimensional vector space, distance can be easily calculated with matrix operation. Thus, we obtained scalability in similarity search. These results suggest that similarity search with the proposed method can be a scalable alternative to BLAST for ever increasing size of biological sequences. In sum, our method enables similarity search on biological sequences with comparable accuracy even without leveraging alignment-based computations.

Discussion

We propose a deep learning based embedding function that simultaneously learns significant protein features at three hierarchy levels of GPCR families. Sequence features in GPCR proteins are learned from the proposed model, which consists of three-branch classifiers and a novel loss term. The main advantage of simultaneous embedding of the hierarchies is that distances in the embedding space are directly correlated with hierarchical relations between proteins, which lies at the heart of GPCR studies. This is significant in that phylogenetic relationships of GPCR protein families can be modeled with the embedding vectors learnt from our deep learning model. Throughout the experiments, we demonstrated that the embedding space correctly modeled GPCR family hierarchy into a single metric space. Furthermore, phylogenetic relations of the proteins were correctly inferred with the proposed method. In addition, embedding space from the proposed method was utilized in sequence similarity search tasks for ever increasing GPCR databases. Experimental results in our work indicate that several downstream analysis on the protein sequences can be successfully accomplished with the embedding vectors generated from the proposed method. In sum, we believe the proposed approach presented a novel and significant strategies for utilizing deep learning in analyzing GPCR protein sequences.

As a future work, information from the structural characteristics of GPCR receptor can be extensively incorporated into the training process in a similar manner to the proposed work. Since structural properties are crucial for identifying the functions of the protein, such investigation might give opportunity to obtain more accurate modeling of GPCR proteins.

References

1. Fredriksson, R., Lagerström, M. C., Lundin, L.-G. & Schiöth, H. B. The g-protein-coupled receptors in the human genome form five main families. phylogenetic analysis, paralogon groups, and fingerprints. *Mol. pharmacology* **63**, 1256–1272 (2003).

2. Bjarnadóttir, T. K. *et al.* Comprehensive repertoire and phylogenetic analysis of the g protein-coupled receptors in human and mouse. *Genomics* **88**, 263–273 (2006).

3. Mirzadegan, T., Benkö, G., Filipek, S. & Palczewski, K. Sequence analyses of g-protein-coupled receptors: similarities to rhodopsin. *Biochemistry* **42**, 2759–2767 (2003).

4. Hu, G.-M., Mai, T.-L. & Chen, C.-M. Visualizing the gpcr network: Classification and evolution. *Sci. reports* **7**, 15495 (2017).

5. Chan, W. K. *et al.* Glass: a comprehensive database for experimentally validated gpcr-ligand associations. *Bioinformatics* **31**, 3035–3042 (2015).

6. Hu, G.-M., Secario, M. & Chen, C.-M. Sequery: an interactive graph database for visualizing the gpcr superfamily. *Database* **2019** (2019).

7. Davies, M. N. *et al.* On the hierarchical classification of g protein-coupled receptors. *Bioinformatics* **23**, 3113–3118 (2007).

8. Peng, Z.-L., Yang, J.-Y. & Chen, X. An improved classification of g-protein-coupled receptors using sequence-derived features. *BMC bioinformatics* **11**, 420 (2010).

9. Davies, M. N. *et al.* Proteomic applications of automated gpcr classification. *Proteomics* **7**, 2800–2814 (2007).

10. Qian, B., Soyer, O. S., Neubig, R. R. & Goldstein, R. A. Depicting a protein's two faces: Gpcr classification by phylogenetic tree-based hmms. *FEBS letters* **554**, 95–99 (2003).

11. Hou, J., Adhikari, B. & Cheng, J. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics* **34**, 1295–1303 (2018).

12. Lanchantin, J., Singh, R., Lin, Z. & Qi, Y. Deep motif: Visualizing genomic sequence classifications. *arXiv preprint arXiv:1605.01133* (2016).

13. Seo, S., Oh, M., Park, Y. & Kim, S. Deepfam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics* **34**, i254–i262 (2018).

14. Chopra, S., Hadsell, R. & LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 539–546 (IEEE, 2005).

15. Hoffer, E. & Ailon, N. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 84–92 (Springer, 2015).

16. Wen, Y., Zhang, K., Li, Z. & Qiao, Y. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, 499–515 (Springer, 2016).

17. Zheng, W. *et al.* Sense: Siamese neural network for sequence embedding and alignment-free comparison. *Bioinformatics* (2018).

18. Bepler, T. & Berger, B. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661* (2019).

19. Cobanoglu, M. C., Saygin, Y. & Sezerman, U. Classification of gpcrs using family specific motifs. *IEEE/ACM Transactions on Comput. Biol. Bioinforma.* **8**, 1495–1508 (2010).

20. Nam, H. & Han, B. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4293–4302 (2016).

21. Saitou, N. & Nei, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. biology evolution* **4**, 406–425 (1987).

22. Edgar, R. C. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research* **32**, 1792–1797 (2004).

23. Hwang, Y., Han, B. & Ahn, H.-K. A fast nearest neighbor search algorithm by nonlinear embedding. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3053–3060 (IEEE, 2012).

24. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. molecular biology* **215**, 403–410 (1990).

25. Quang, D. & Xie, X. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research* **44**, e107–e107 (2016).

26. Pan, X., Rijnbeek, P., Yan, J. & Shen, H.-B. Prediction of rna-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks. *BMC genomics* **19**, 511 (2018).

27. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. In Wallach, H. *et al.* (eds.) *Advances in Neural Information Processing Systems 32*, 8024–8035 (Curran Associates, Inc., 2019).

28. Asnicar, F., Weingart, G., Tickle, T. L., Huttenhower, C. & Segata, N. Compact graphical representation of phylogenetic data and metadata with graphlan. *PeerJ* **3**, e1029 (2015).

29. Müllner, D. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378* (2011).

30. Maaten, L. v. d. & Hinton, G. Visualizing data using t-sne. *J. machine learning research* **9**, 2579–2605 (2008).

31. Crooks, G. E., Hon, G., Chandonia, J.-M. & Brenner, S. E. Weblogo: a sequence logo generator. *Genome research* **14**, 1188–1190 (2004).

32. Rosenbaum, D. M., Rasmussen, S. G. & Kobilka, B. K. The structure and function of g-protein-coupled receptors. *Nature* **459**, 356 (2009).

33. Rovati, G. E., Capra, V. & Neubig, R. R. The highly conserved dry motif of class ag protein-coupled receptors: beyond the ground state. *Mol. pharmacology* **71**, 959–964 (2007).

34. Harmar, A. J. Family-b g-protein-coupled receptors. *Genome biology* **2**, reviews3013–1 (2001).

Acknowledgements

This research is supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (No.NRF-2017M3C4A7065887), the Collaborative Genome Program for Fostering New Post-Genome Industry of the National Research Foundation (NRF) funded by the Ministry of Science and ICT (No.NRF2014M3C9A3063541), the Original Technology Research Program of the National Research Foundation (NRF) funded by the Ministry of Science and ICT (NRF-2019M3E5D4065965) ,and a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number : HI15C3224).

Author contributions statement

All authors conceived the experiment(s), T.L. and S.L. conducted the experiment(s), T.L., S.L. and M.K. analysed the results. All authors reviewed the manuscript.

Additional information

Supplementary Data are available at online. Code for this study is available at https://github.com/thlee93/DeepHierProtein.
Competing financial interests None declared. The corresponding author is responsible for submitting a competing financial interests statement on behalf of all authors of the paper.

Figures

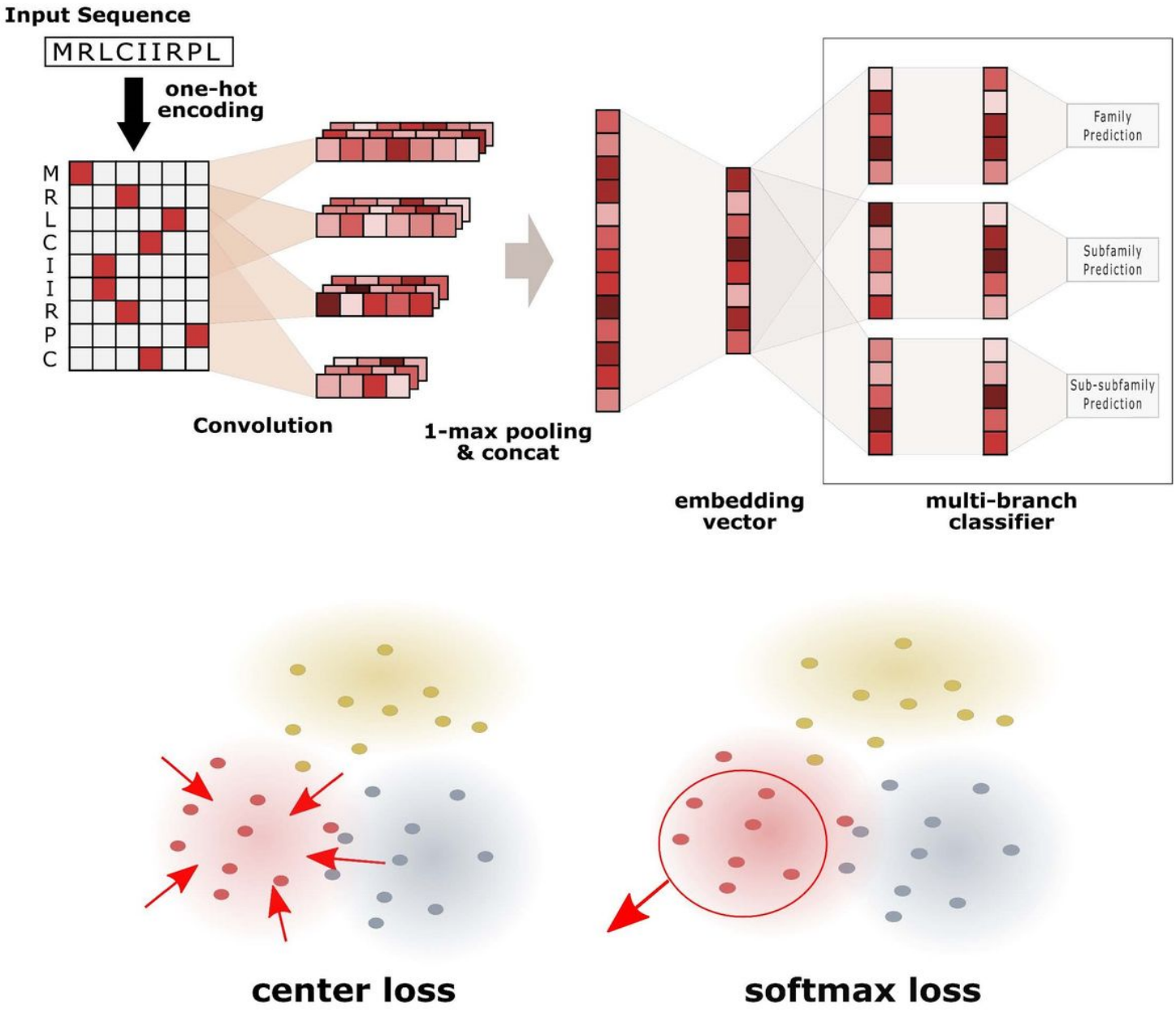


Figure 1

Neural network architecture and loss function.

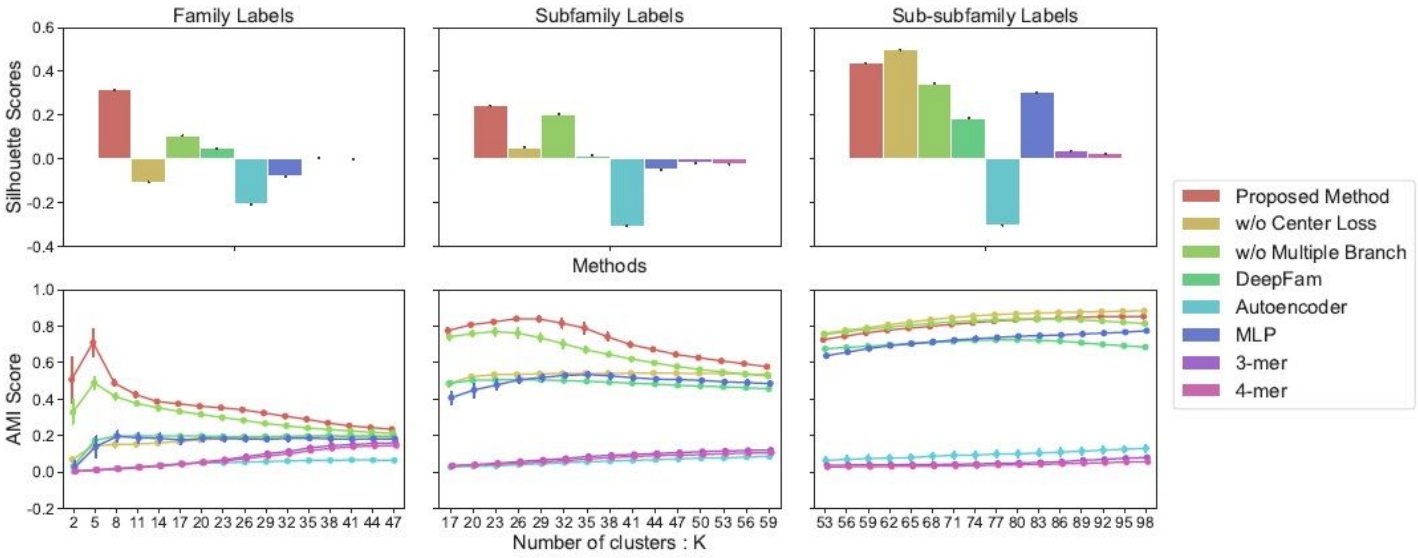
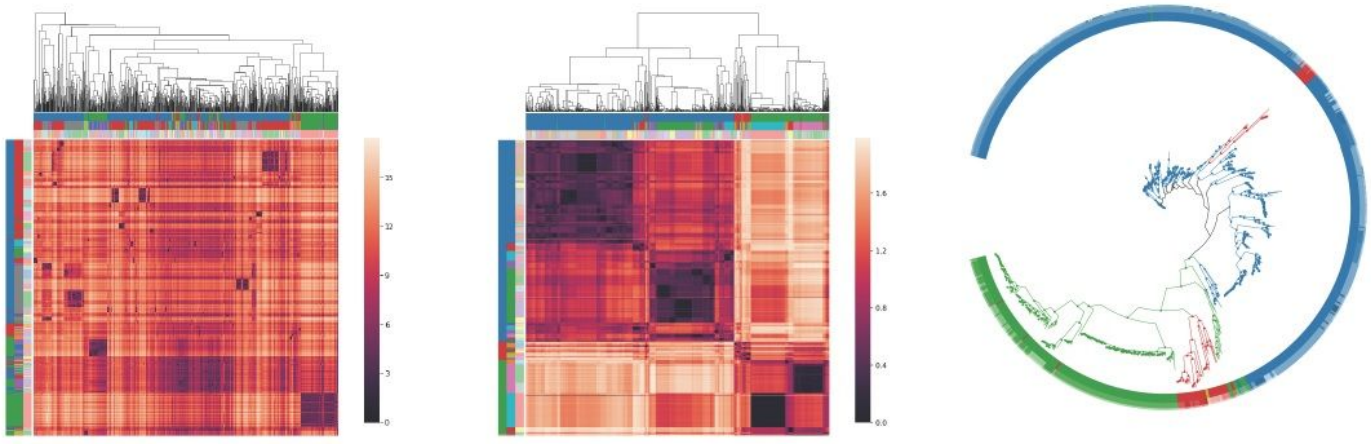


Figure 2

Top: silhouette score of representation vectors regarding true class labels on three hierarchical levels. Bottom: adjusted mutual information (AMI) score calculated for class label in each level versus k clusters yielded from hierarchical clustering algorithm.



Distance matrix from DeepFam

Distance matrix from the Proposed Method

Phlyogenetic Tree from the Proposed Method

Figure 3

Embedding vectors on hierarchical structure, distance matrix and phylogenetic tree. Left: distance matrix of embedding vectors from DeepFam. Center: distance matrix of embedding vectors from the proposed method. Right: phylogenetic tree reconstructed from the proposed method.

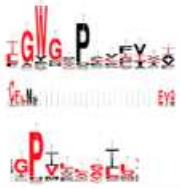
	Cluster 3_1	Cluster 5_4	Cluster 55_16	Cluster 55_48
Target GPCR class	Family A	Family B	Sub-subfamily Melanocortin	Sub-subfamily Traceamine
# Total sequences	519	46	43	14
# Sequences in target class	507	44	41	14
Demonstration of Extracted Motifs				

Figure 4

Discovered motif logos for each cluster on phylogenetic tree generated from hierarchical clustering results. Cluster information as well as information on the sequences included in the cluster is provided. Cluster 3-1 denotes that this cluster is first cluster from the hierarchical clustering when the target number of cluster is 3.

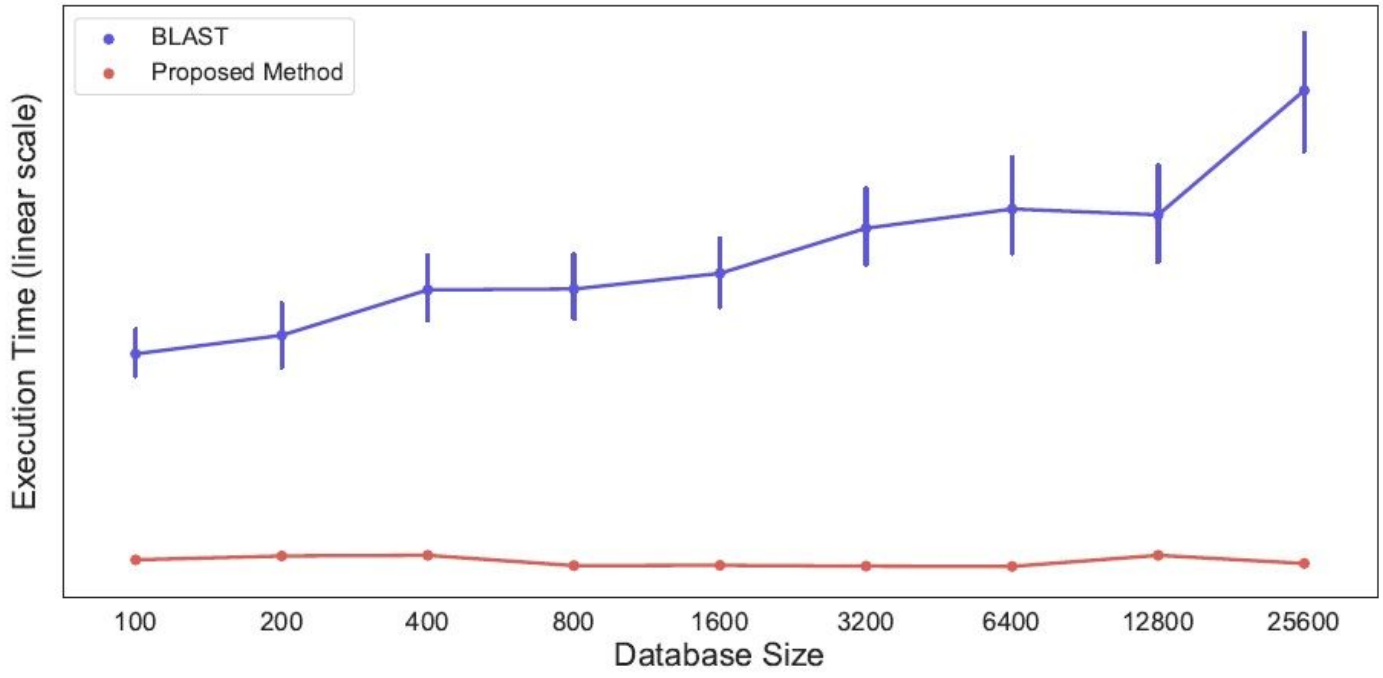


Figure 5

Execution time measured with varying the size of database.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- DeepMetricScientificReportsSupplemetaryFile.pdf