How suitable are clustering methods for functional annotation of proteins?

Rakesh Busi[1,*], Pranav Machingal[2], Nandyala Hemachandra[2], and Petety V. Balaji[1]

[1]Department of Biosciences and Bioengineering, Indian Institute of Technology Bombay, Mumbai 400076, India
[2]Department of Industrial Engineering and Operations Research, Indian Institute of Technology Bombay, Mumbai 400076, India
[*]rakesh.busi@iitb.ac.in

December 26, 2024

Abstract

The advent of affordable high-throughput genome sequencing has drastically expanded protein sequence databases, necessitating the development of computational tools to predict protein function from sequence data. Current methods, such as BLASTp and profile HMMs, while effective, are limited by difficulties in detecting remote homologs and uncertainties in multiple sequence alignments. To address this, we explore the use of clustering algorithms for unsupervised protein function annotation, using pseudo-amino acid composition (PAAC) as features.

In this study, we evaluated nine clustering algorithms for their ability to segregate protein sequences based on functional differences using the PAAC feature. Using intrinsic metrics, particularly the silhouette coefficient (SC), we determined the optimal number of clusters (k) for each algorithm. We observed that agglomerative clustering produced results resembling phylogenetic relationships; even k-means clustering, Gaussian mixture model(GMM), and spectral clustering do so but occasionally merge datapoints from distinct original clusters at higher k values.

Our findings reveal that k-means clustering, GMM, and agglomerative clustering effectively segregate distinct protein functional families, but effectiveness decreases when distinguishing fine-grained functional differences. Notably, spectral clustering underperformed relative to other methods. Affinity propagation clustering, while effective in some cases, generated more clusters than expected and is prone to false positives. Overall, we find that some of the clustering algorithms are suitable for functional annotation of protein sequences using PAAC as a feature set, even when the number of ground-truth sequences is limited.

The implementation of the clustering method for protein sequences is available in the GitHub repository linked below. It provides comprehensive steps for preprocessing, feature extraction, clustering, and evaluation. All results are presented in a Jupyter Notebook. https://github.com/RakeshBusi/Clustering

Introduction

Affordable high-throughput genome sequencing has transformed the way biologists ask questions to understand life at the molecular and cellular level. An offshoot of genome sequencing is the enormous increase in the size of protein sequence databases. The UniProtKB/TrEMBL release 2024_03 contains 244,910,918 sequences [8]. Knowledge of the functions of these proteins is necessary to fully realize the benefits of genome

sequencing. However, experimental characterization of the functions of all these proteins is impractical. This hurdle can be circumvented to a large extent by computational methods that can predict possible function(s) based primarily on amino acid sequence.

The "function" of a protein is a collection of descriptors or attributes. Currently available methods, whether experimental or computational, at best characterize only some of these attributes. Consequently, coverage of function annotation in databases is incomplete as well as non-uniform, i.e., some entries of the database are richly annotated whereas some have none, as shown in UniProtKB/TrEMBL. Currently, BLAST [2] (including its variants), profile HMMs [18], and position weight matrix-based motif search [7] are the computational methods of choice for function annotation. However, BLAST is not suitable for detecting remote homologs. Both PSI-BLAST and profile HMMs are critically dependent on high-quality multiple sequence alignments for assigning function with higher accuracy. However, proteins often have large indels relative to their homologs, leading to uncertainties in the placement of gaps in multiple sequence alignments. In addition, as sequences diverge, it becomes difficult to decide if two homologs are paralogs or orthologs.

The last few years have witnessed a surge in the use of machine learning methods to analyze biological data [3, 11]. We are broadly interested in using unsupervised methods for annotating the molecular function of sequences in the database. Clustering algorithms are widely used in very diverse domains [29, 28, 37] and, being unsupervised methods, they do not need labels for datapoints. The non-requirement of labels allows one to combine experimentally characterized (i.e., functionally annotated) and unannotated sequences for assigning functions to the latter. Working towards such a broad goal, our focus in this study is to assess the ability of clustering methods to segregate input protein sequences along the lines of their functional differences. Herein we report the clustering of functionally distinct protein sequences using pseudo amino acid composition (PAAC) [14] as features. PAAC was chosen because the amino acid sequence of a protein is an important determinant of its molecular function.

In the current study to evaluate the effectiveness of clustering methods in protein annotation, we selected datasets containing protein sequences labeled with fine-grained functional differences. Specifically, we focus on Glycoside Hydrolases (GH), which are crucial enzymes in carbohydrate metabolism. The GH dataset was chosen for several reasons: first, it consists of manually curated protein sequences from the widely recognized CAZy database [17]; second, the GH enzymes are categorized by EC numbers (EC 3.2.1.x) [24], where the first three digits represent glycosidases that hydrolyze O- and S-glycosyl compounds, and the final digit specifies substrate specificity, indicating fine-grained functional differences; third, GH enzymes are grouped into GH families in the CAZy database, which not only reflect structural features but also reveal evolutionary relationships [21]. In addition to GH datasets, we included datasets with broader functional differences to assess the robustness of clustering methods across a wider spectrum of protein functions.

Methods

Nomenclature

A protein sequence is treated as a 'datapoint' and a protein family is treated as a 'class' in the context of the clustering algorithm. Homologs are the proteins that evolved from the same ancestral protein. Among homologs, orthologs are those that retain molecular function, while paralogs diverge functionally; the extent of functional divergence varies [25]. A protein family is composed of a set of sequence or structural homologs. Proteins that perform the same molecular function are referred to as functional homologs. Functional homology can arise even in the absence of shared ancestry due to convergent evolution. In this research paper, "function" specifically refers to "molecular function", one of the three "aspects" used by Gene Ontology consortium to organize biological knowledge about proteins; the other two are the cellular component and the biological process. Molecular function covers reaction catalyzed (EC number), ligand/substrate specificity, etc. [5].

Feature engineering using pseudo amino acid composition (PAAC)

Sequences of proteins must be transformed into feature vectors of fixed dimensions for clustering algorithms to process them. Pseudo amino acid composition is a widely used and robust feature extraction method that captures both amino acid composition and sequence-order correlation [15]. In our study, we used the default values from the original paper [14] for the weightage factor (0.05) and lambda (30); the physicochemical properties used are the same as those suggested, i.e., hydrophobicity, hydrophilicity, and side chain mass. As a result, each feature vector has 50 dimensions, where the first 20 correspond to amino acid composition, and the next 30 correspond to sequence-order correlation. We used in-house scripts for all data processing steps except where mentioned to the contrary.

Clustering

Clustering is a type of unsupervised learning [23]. We have assessed nine clustering algorithms provided in the scikit-learn platform [30]. The number of clusters k to be formed by input data is user-defined for k-means clustering, spectral clustering, agglomerative clustering, and Gaussian mixture model (GMM). On the other hand, the number of clusters is determined by the algorithm itself in the case of DBSCAN clustering, OPTICS clustering, affinity propagation clustering (APC), and mean-shift clustering. In the case of BIRCH clustering, `n_clusters` was set to `X`, where `X` represents the number of classes in the dataset. BIRCH first creates a clustering feature tree (CFT) from input data and groups datapoints into subclusters. If the number of subclusters is greater than the user-defined k, then BIRCH uses a global clustering algorithm to reduce the number of subclusters to k. And if the number of subclusters is less than or equal to the user-defined k, then BIRCH retains the current subclusters as the final clusters [39]. All algorithms were used with their default values, except for two cases: (i) in spectral clustering, the default value of `affinity` parameter `rbf` was changed to `nearest_neighbors` to ensure scale invariance, (ii) in APC, the value for the `preference` parameter was changed for reasons given below.

The performance of APC is primarily dictated by the parameter `preference`. Choosing a `higher` value can result in more number of datapoints to be cluster centers (the so-called exemplars). Alternatively, one can assign `preference` values for each of the datapoints, and those with higher values are likely to be exemplars. By default, the `preference` is set to the median of the similarity matrix [36]. The number of clusters formed for various datasets ranged between 4 and 249. These numbers are very large when one considers the way datasets are curated; for example, at most 2 or 3 clusters were expected for the lysozyme CGCh dataset. Using the minimum value in the similarity matrix is an option, and this is expected to reduce the number of clusters. Hence, the datasets were clustered by setting `preference` to the minimum value. As expected, fewer clusters are obtained for all datasets.

Assessment of the quality of clustering

The quality of clustering was assessed using (i) intrinsic metrics and (ii) database-assigned labels [27].

Assessment using intrinsic metrics

Different intrinsic metrics have been used by various research groups. In the present study, three intrinsic evaluation metrics available in the scikit-learn [30] platform were computed. These are the silhouette coefficient (SC) [32], Calinski-Harabasz (CH) index [13], and Davies-Bouldin (DB) index [16]. These three metrics were part of a group of 6 metrics (out of 30) that were found to perform satisfactorily against 720 synthetic and 20 real datasets clustering using k-means, ward, and average-linkage algorithms [4].

The SC is calculated by comparing the mean distance between a sample and all other points within the same cluster to the mean distance between the sample and points in the nearest neighboring cluster. The SC ranges from -1 to +1, where -1 indicates incorrect clustering, +1 indicates highly dense clustering, and values around zero suggest overlapping clusters [32]. The DB index compares the distance between clusters with

the size of the clusters. Zero is the lowest possible score, which indicates a better partition. However, there is no defined upper bound for the DB index [16]. The CH index is the ratio of the sum of between-clusters dispersion and the sum of within-cluster dispersion for all clusters (here the dispersion is defined as the sum of squared distances) [13].

Assessment using protein sequence database-assigned labels

Even though clustering is an unsupervised learning method, database-assigned labels have also been used to assess the quality of clustering by means of contingency matrices and the Fowlkes-Mallows index (FMI) [19]. It may be noted that database-assigned labels are labels that have not been part of the input for clustering. For preparing contingency matrices, sequence database annotations of the input datapoints (i.e., protein sequences) were used as labels for comparison with the cluster labels assigned by clustering algorithms.

Extrinsic evaluation metrics are less commonly used than intrinsic metrics, mainly due to the requirement of database-assigned labels [4]. In this study, we considered four extrinsic metrics: FMI, F1, MCC, and AMI. To check for redundancy among these metrics, we computed the correlation coefficient between every pair of metrics on all datasets and among all algorithms with k= 2-10. The correlation coefficients were found to be 0.99 between FMI and F1, 0.95 between AMI and MCC, and 0.77 between FMI and AMI. Hence, we only considered FMI for further analysis.

FMI [19] is the geometric mean of pairwise precision and recall. It is calculated by considering two datapoints at a time. This is illustrated in table 1 where circles represent datapoints and their colors represent assigned labels. The class is calculated for all pairs of datapoints based on database-assigned labels and cluster labels. The cluster(s) to which these datapoints belong and the labels assigned from the database are compared (table 1). FMI is 1 when every pair of datapoints that belong to the same cluster has the same labels. FMI is 0 when every pair belonging to the same cluster has non-identical labels.

$$FMI = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}} \tag{1}$$

Table 1: Schematic illustration of classes and corresponding labels.

Class	**Database-assigned labels**$^{\phi}$	**Cluster labels**†
True Positive	🟢 🟢	🟡 🟡
True Negative	🟢 🔵	🟠 🟡
False Negative	🟢 🟢	🟠 🟡
False Positive	🟢 🔵	🟠 🟠

$^{\phi}$ Green and blue circles denote distinct labels.
† Yellow and orange circles denote distinct labels. Green and yellow circles may or may not denote the same labels for a given pair of datapoints.

Creation of datasets

We manually curated the datasets, as described in table 2. It was observed that the sequence lengths of proteins within a protein family show large variation. Therefore, outliers identified using a box-whisker plot were omitted. The following sequences were also omitted: (i) contain letters B, J, O, U, X, and Z, since these letters do not uniquely denote any of the 20 standard amino acids, (ii) have the term "fragment" or "partial" in their annotation, and (iii) have two or more EC numbers in their annotation. Additionally, within a protein family, only one copy of identical sequences or accession numbers was retained, while identical sequences or accession numbers were removed across protein families within the dataset. Finally, sequences with a length

< 30 were omitted because the default value of lambda was set to 30 for calculating the PAAC of sequences, resulting in a uniform 50-dimensional feature vector.

Table 2: Summary of 15 datasets and their associated protein families.

Group[†]	**Dataset**	**Protein family**	**EC no.**	**Fold**	**# seq.**
Group 1	Lysozyme CGCh[‡]	Lysozyme C	3.2.1.17	lysozyme fold	52
		Lysozyme G	3.2.1.17	lysozyme fold	17
		Lysozyme Ch	3.2.1.17	$(\beta/\alpha)_5(\beta)_3$ Tim-like	54
	Xylanase	GH10	3.2.1.8	$(\beta/\alpha)_8$ barrel	352
		GH11	3.2.1.8	β-jelly roll	347
	Chitinase	GH18	3.2.1.14	$(\beta/\alpha)_8$ barrel	462
		GH19	3.2.1.14	lysozyme fold	194
Group 2	Lysozyme CaLA	Lysozyme C	3.2.1.17	lysozyme fold	52
		α-Lactalbumin	None	lysozyme fold [22]	22
	Protease	Trypsin	3.4.21.4	chymotrypsin/ trypsin fold [20]	67
		Chymotrypsin	3.4.21.1	chymotrypsin/ trypsin fold [20]	16
	Ferredoxin	FDX1	None	ferredoxin fold [33]	9
		FDX2	None	ferredoxin fold	7
Group 3	β-Glucosidase	GH1	3.2.1.21	$(\beta/\alpha)_8$ barrel	348
		GH3	3.2.1.21	$(\beta/\alpha)_8$ barrel	204
	Lysozyme	GH22	3.2.1.17	lysozyme fold	135
		GH23	3.2.1.17	lysozyme fold	47
		GH24	3.2.1.17	lysozyme fold	31
	β-Galactosidase[¶]	GH2	3.2.1.23	$(\beta/\alpha)_8$ barrel	117
		GH35	3.2.1.23	$(\beta/\alpha)_8$ barrel	295
		GH42	3.2.1.23	$(\beta/\alpha)_8$ barrel	83
Group 4	GH2	β-Galactosidase	3.2.1.23	$(\beta/\alpha)_8$ barrel	117
		β-Glucuronidase	3.2.1.31	$(\beta/\alpha)_8$ barrel	85
	GH3	β-Glucosidase	3.2.1.21	$(\beta/\alpha)_8$ barrel	204
		1,4-β-Xylosidase	3.2.1.37	$(\beta/\alpha)_8$ barrel	99
	GH5	Cellulase	3.2.1.4	$(\beta/\alpha)_8$ barrel	342
		Endo-β-Mannanase	3.2.1.78	$(\beta/\alpha)_8$ barrel	127
Group 5	SNARE	SNARE	None	No specific fold	57
		non-SNARE	None	No specific fold	61
	GPCR1	GPCR	None	7TM helix fold [38]	1697
		non-GPCR (TM)	None	No specific fold	1907
	GPCR2	GPCR	None	7TM helix fold	1697
		non-GPCR (no TM)	None	No specific fold	1861

[¶] Protein families in this dataset also belong to the same GH clan
[†] Groups will be discussed in later subsection
[‡] GH families related information is taken from CAZy database (http://www.cazy.org/Glycoside-Hydrolases.html)

CAZy is a database of carbohydrate-active enzymes wherein protein sequences have been grouped into different families based on sequence similarity. The protein families in Xylanase, Chitinase, β-Glucosidase, Lysozyme, β-Galactosidase, and Lysozyme CGCh datasets were fetched from the CAZy database [17] using their corresponding EC numbers as queries. Additionally, protein names were used for the protein families in the Lysozyme CGCh dataset. For other families, protein names could not be used because of the multiple and ambiguous nomenclatures. These datasets were chosen due to their relatively large size. The GH2, GH3, and GH5 datasets were also taken from the CAZy database due to their large size and the presence of two functionally distinct protein families. Protein families in the Protease dataset were fetched from BRENDA

[24] using EC numbers, and those in the Ferredoxin dataset from Swiss-Prot [12] using protein family names. α-Lactalbumin sequences were collected from [34] and these constitute the Lysozyme CaLA dataset along with lysozyme C sequences.

The SNARE dataset is identical to the D128 dataset described by [6] and was obtained from their electronic supplementary data. The authors of this work downloaded SNARE sequences from UniProt using the filter GO:0005484, which corresponds to the molecular function "SNAP receptor activity," while non-SNARE sequences were fetched from the PDB database [10]. They further used CD-HIT [26] with a 25% sequence similarity threshold to obtain a non-redundant dataset.

The GPCRs in GPCR1 and GPCR2 datasets were obtained from Swiss-Prot using the advanced search option. The "protein family" field under "Family and Domains" was queried with the keywords 'G protein-coupled receptor n family' (where $n = 1,\ldots,5$), 'G protein-coupled receptor fz smo family' and 'G protein-coupled receptor T2R family' which correspond to IUPHAR classes A-F and Taste 2, respectively [1],[9]. The non-GPCRs were also obtained from Swiss-Prot using the NOT operator for the keywords given above. They were then split into two groups: (i) sequences with at least one transmembrane domain (non-GPCR (TM)) and (ii) sequences with no transmembrane domain (non-GPCR (no TM)). A 90% sequence identity cut-off was applied to each of the three protein families using CD-HIT to reduce redundancy and size. From these, 2000 entries were chosen randomly from each of the non-GPCR (TM) and non-GPCR (no TM) families in order to maintain class balance with the GPCR family in the GPCR1 and GPCR2 datasets, respectively.

Sequence similarity across protein families of a dataset

We performed an all-against-all sequence comparison using BLASTp. Two proteins were considered to be similar (or homologous) if the percentage sequence identity (PSI) $\geq$ 30% [31] and the alignment covers at least 70% (empirically) of one of the sequences. This was used to infer whether protein families shared sequence similarity or not (data not shown).

Grouping of dataset based on characteristic properties

As mentioned earlier, we have considered a diverse range of 15 datasets, each of which contains $\geq$ 2 protein families. Depending on the presence (or lack of) sequence, 3D structural, and/or functional similarity between protein families constituting a dataset, the 15 datasets have been divided into 5 groups table 3.

Table 3: Grouping of 15 datasets based on their similar/dissimilar characteristics.

Group	**Characteristics**	**Datasets**
Group 1	Sequence - Dissimilar	Lysozyme CGCh (3 families)
	3D structure - Different fold	Xylanase (2 families)
	Function - Similar	Chitinase (2 families)
Group 2	Sequence - Similar	Lysozyme CaLA (2 families)
	3D structure - Same fold	Protease (2 families)
	Function - Dissimilar	Ferredoxin (2 families)
Group 3	Sequence - Dissimilar	β-Glucosidase (2 families)
	3D structure - Same fold	Lysozyme (3 families)
	Function - Similar	β-Galactosidase (3 families)
Group 4	Sequence - Dissimilar	GH2 (2 families)
	3D structure - Same fold	GH3 (2 families)
	Function - Dissimilar	GH5 (2 families)
Group 5	Sequence - Dissimilar	SNARE (2 families)
	3D structure - Different fold	GPCR1 (2 families)
	Function - Dissimilar	GPCR2 (2 families)

Group 1: Lysozyme CGCh, Xylanase, and Chitinase datasets belong to this group. Protein families in these datasets perform the same function, i.e., sequences have the same EC number. But there is no sequence or 3D structure similarity (adopt different folds).

Group 2: Lysozyme CaLA, Protease, and Ferredoxin datasets belong to this group. Protein families in these datasets share sequence and 3D structure similarity but differ in their function. The difference is fine-grained in the case of the Protease dataset because chymotrypsin and trypsin catalyze the same reaction (hydrolases of the peptide bond) but differ in their substrate specificity. Lysozyme C and α-Lactalbumin of the Lysozyme CaLA dataset, and FDX1 and FDX2 of the Ferredoxin dataset perform mutually exclusive functions [33].

Group 3: β-Glucosidase, Lysozyme, and β-Galactosidase datasets belong to this group. Protein families in these datasets perform the same function, i.e., sequences have the same EC number and have the same fold. But there is no sequence similarity (based on all-against-all pairwise sequence alignment; data not shown).

Group 4: GH2, GH3, and GH5 datasets belong to this group. These datasets are distinct from those in other groups: Protein sequences from all three datasets have the same first 3 digits of EC number, i.e., 3.2.1.-. The description for EC 3.2.1.- is *Glycosidases, i.e., enzymes hydrolyzing O- and S-glycosyl compounds*. They differ in their substrate specificities, leading to differences in the 4th digit of the EC number. Also, there is no or very little sequence similarity (based on all-against-all pairwise sequence alignment; data not shown). All datapoints have the same fold, i.e., $(\beta/\alpha)_8$ barrel.

Group 5: This group consists of SNARE, GPCR1, and GPCR2 datasets. These datasets comprise proteins with no catalytic activity (no EC number). The SNARE family within the SNARE dataset consists of proteins that perform a generic receptor function but differ from each other in various aspects: localization, key functional residue, presence of transmembrane domains, coiled-coil domain, downstream pathway, etc. [35]; there is no discernible sequence identity with each other. The SNARE and non-SNARE families don't share any sequence, structure, and function similarities. The GPCR1 and GPCR2 datasets are similar to the SNARE dataset, as the GPCR family does not share any sequence, structure, and function similarities with non-GPCR (TM) and non-GPCR (no TM) families, respectively.

Results

Suitability of intrinsic metrics and possible thresholds

As mentioned in the Methods section, silhouette coefficient (SC), Calinski-Harabasz (CH) index, and Davies-Bouldin (DB) index were used to analyze clustering output. For a given dataset + algorithm combination, each of the three metrics was plotted as a function of k.

By definition, SC is bounded between -1 for incorrect clustering and +1 for highly dense clustering [32]. Scores around zero indicate overlapping clusters. SC was computed for all combinations of datasets, clustering algorithms (k-means, GMM, agglomerative, and spectral clustering), and k (varied from 2 to 10). It was observed that SC is not close to 1 for any of the clusters. It is not clear if this is because all the clusters are of poor quality or if this is the best separation one can get for the type of datasets + features we have used. This was resolved in the following two ways:

(i) We clustered a set of datapoints that consists of two completely unrelated protein families, i.e., SNARE and trypsin. It is seen that SC is between 0.31 and 0.32.

(ii) We assigned database-assigned labels to every datapoint and treated these as the cluster labels for computing the SC value. Taking the Xylanase dataset as an example, the computed SC value was 0.35, closely matches the SC values obtained using cluster labels from the four algorithms, which ranged from 0.35 to 0.36 for $k = 2$. A similar pattern was observed for other datasets also.

From these, we infer that SC is unlikely to reach the highest value (1.00) irrespective of the value of k and the clustering algorithm for the type of data and feature engineering we are using in this study. In view of

this, for a given algorithm, we chose the k value for which SC is highest and positive as optimal. This k value happens to be the same if we were to use the lowest DB index instead of the highest SC, except in 8 out of 15 cases. The choice of SC over the DB index is because the DB index does not have an upper limit.

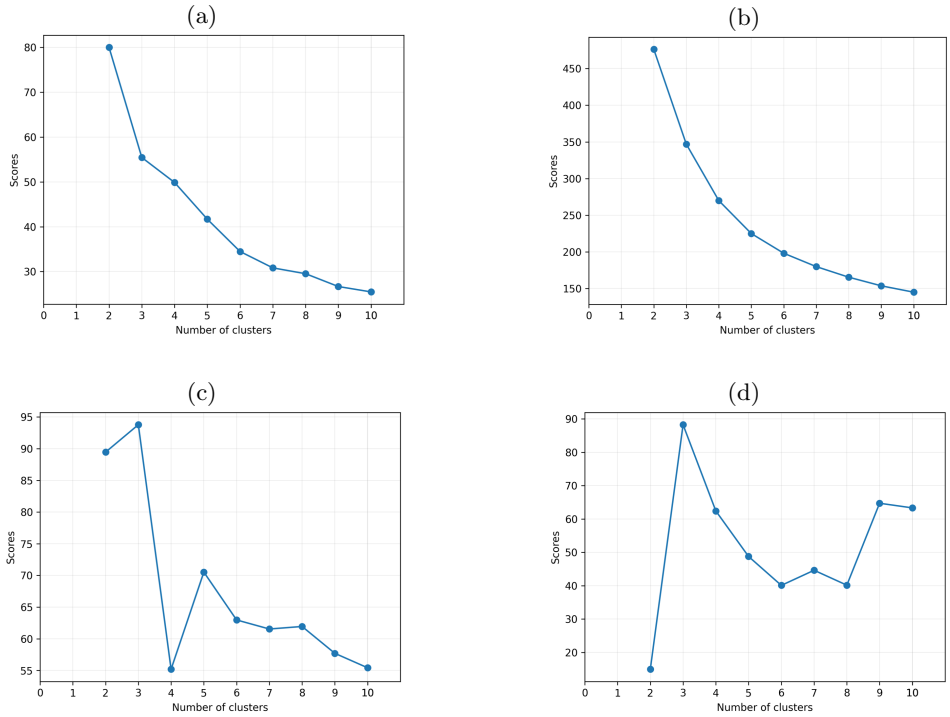

Figure 1: The CH index (scores) decreases monotonically as k increases as shown in (a) k-means clustering of the Lysozyme CGCh dataset and (b) agglomerative clustering of the Xylanase dataset with a few exceptions as shown in (c) spectral clustering of the Chitinase dataset and (d) spectral clustering of the β-Glucosidase dataset (Data not shown for other datasets and algorithms).

The CH index decreases monotonically as k increases, with a few exceptions, as shown in fig. 1. Thus, $k = 2$ happens to have the best score irrespective of the input dataset and the clustering algorithm. Also, by definition, there is no upper bound for the CH index. The highest CH index for the various datasets (among all algorithms) ranged from 6.82 (Ferredoxin, $k = 10$, agglomerative) to 1001.27 (GPCR2, $k = 2$, k-means) as shown in Supplementary_figure1. It is not clear how this index can be used to ascertain the quality of clustering when one is working with multiple input datasets and clustering algorithms.

Overall, we infer that the CH index and DB index are not suitable in the current study. SC was used without reference to the absolute value as long as it is positive.

Analysis of the number and sizes of clusters formed by mean-shift, DBSCAN, OPTICS, and BIRCH

Mean-shift, DBSCAN, OPTICS, and BIRCH determine the number of clusters to be formed based on input data. The number of clusters formed is shown in table 4. BIRCH gave a single cluster for all the datasets. Mean-shift groups most of the datapoints in a single cluster even when the number of clusters formed is >1. DBSCAN and OPTICS identify most of the datapoints as noise points irrespective of the number of datapoints. This output cannot be reconciled considering domain knowledge as explained in detail below.

The performance of mean-shift is primarily dictated by the parameter `bandwith` which determines the size

Table 4: Number clusters (k) predicted and the cluster with maximum datapoints by mean-shift, DBSCAN, and OPTICS .

Group	Dataset	# seq.	mean-shift		DBSCAN		OPTICS	
			k	Cluster size$^{\Psi}$	k	Noise¶	k	Noise
Group 1	Lysozyme CGCh	123	3	121	1	123	6	72
	Xylanase	699	5	365 & 331$^{\sigma}$	14	610	38	418
	Chitinase	656	5	633	24	504	48	301
Group 2	Lysozyme CaLA	74	5	46 & 24$^{\sigma}$	1	74	5	50
	Protease	83	1	83	1	83	5	51
	Ferredoxin	16	4	12	1	16	1	16
Group 3	β-Glucosidase	552	1	552	19	425	32	322
	Lyszoyme	213	1	213	8	170	15	77
	β-Galactosidase	495	5	418	28	250	49	143
Group 4	GH2	202	1	202	7	147	12	108
	GH3	303	1	303	10	243	18	182
	GH5	469	2	467	6	437	20	345
Group 5	SNARE	118	5	113	1	118	1	118
	GPCR1	3604	84	3351	1	3604	32	3376
	GPCR2	3558	92	3278	1	3558	29	3353

$^{\Phi}$ BIRCH results were not shown as it gave single cluster for all datasets.
$^{\Psi}$ Cluster size is the cluster with maximum number of datapoints.
¶ In case of DBSCAN and OPTICS, the cluster with maximum number of datapoints were noise points.
$^{\sigma}$ In this case, we showcased two cluster with maximum datapoints.

of the window in which datapoints are considered for calculating the mean. By default, mean-shift uses an estimator to determine the value of the `bandwidth`. It appears that the algorithm has estimated a high `bandwidth` value, resulting in larger neighborhoods leading to a single cluster.

The performance of DBSCAN is primarily dictated by the parameters `eps` and `min_samples`. The former sets the maximum distance between two samples for one to be considered as in the neighborhood of the other. The latter determines the number of samples in a neighborhood for a point to be considered as a core point. There is no basis on which values for these parameters can be chosen for the datasets considered in this study. Hence, default values, i.e., 0.5 and 5, respectively, were chosen. With these values, while a single cluster was obtained for some datasets, 6-28 clusters were obtained for the remaining datasets.

The performance of OPTICS is primarily determined by the parameters `max_eps` and `min_samples`. Being a variant of DBSCAN, `max_eps` has the same meaning as `eps`. The default value for this parameter is ∞. The use of default values led to a single cluster for two datasets (Ferredoxin and SNARE) and 5–49 clusters for other datasets.

The performance of BIRCH is primarily determined by the parameters `threshold` and `branching_factor`. The former parameter (default value = 0.5) defines the maximum radius within which a new sample can merge with the closest existing subcluster. The latter parameter (default value = 50) determines the maximum number of subclusters that can branch from a parent cluster in the hierarchical structure. A single cluster was obtained for all 15 datasets, suggesting that the default value for `threshold` is so high that even though `branching_factor` is set to 50, only a single cluster is formed.

It can be inferred that the default values for the various parameters of these four algorithms are not optimal for the feature engineering that we have chosen. Given the nature of the input considered in the present study, i.e., unlabeled datapoints whose relative spread in the 50-D space is not known a priori, there is no straightforward way to determine optimal values for the various parameters. Limited efforts to fine-tune the

values were infructuous because values that may seem optimal for one dataset are not necessarily optimal for others. Hence, results from these four algorithms were not considered to arrive at the conclusions.

Overall, it is concluded that the output from these four algorithms will not be considered for further analysis.

Analysis of the number and sizes of clusters formed by affinity propagation clustering

The number of clusters into which datapoints are to be segregated is determined by the algorithm itself in the case of affinity propagation clustering (APC). The number of clusters ranges from 2-12 considering all 15 datasets. However, we expected 2 or 3 clusters based on database-assigned labels. SC values range from 0.15 to 0.30 except as noted here. SC values range from 0.05 to 0.11 for GH5, SNARE, GPCR1, and GPCR2, which are comparatively closer to 0, indicating overlapping clusters. The clustering results were analyzed using database-assigned labels also; few examples are shown in table 5 and Supplementary_table1 may be referred to for complete data. The FMI values for some of the datasets are relatively lower because the number of database-assigned labels is fewer than the number of clusters (k) formed, resulting in an increase in the number of false negatives.

Visual analysis of contingency matrices shows that most of the clusters formed by APC have datapoints with the same database-assigned labels in the case of Lysozyme CGCh, Xylanase, Chitinase, Lysozyme CaLA, Ferredoxin, β-Glucosidase, and GPCR2 datasets. This suggests that datapoints are divergent (vis-à-vis the values used for various parameters of the algorithm) in the 50-D space despite having the same database-assigned labels.

The scenario is different in the case of the remaining 8 datasets. Let us take the β-Galactosidase dataset as an example. There are three protein families in the β-Galactosidase dataset, viz., GH2, GH42, and GH35. Sequences belonging to GH35 form many clusters with hardly any datapoints from the other two families. Sequences belonging to the GH2 and GH42 families together form multiple clusters. This is suggestive of the proximity of GH2 and GH42 datapoints to each other in the 50-D space. From this, one can envisage three possibilities: (i) These sequences have fine-grained functional differences despite sequence similarity. (ii) PAAC is not capturing the sequence features that determine function. (iii) The database-assigned labels need revision. A similar scenario is found in the case of the Lysozyme dataset as shown in table 5(b). However, in the case of the Protease, GH2, GH3, GH5, SNARE, and GPCR1 datasets, the database-assigned labels and cluster labels show large-scale disagreement. It appears that the database-assigned labels need revision in all these cases.

Pattern of cluster formation by k-means, GMM, agglomerative, and spectral clustering

All 15 datasets were used as input separately for each of these four algorithms and by varying k from 2 to 10. The results were first analyzed by comparing the membership of datapoints in different clusters for two consecutive values of k in fig. 2. It was observed that with each increment of k, agglomerative splits one, and only one, cluster into two. This is to be expected since agglomerative is a hierarchical method and, by design, iteratively clusters datapoints based on *mutual similarity* in the feature space. The process of clustering is depicted as a dendrogram and mimics a phylogenetic tree. In the present study, protein families that constitute datasets are indeed phylogenetically related. Hence, splitting of clusters with increments in k can be thought of as depicting evolutionary divergence.

The results from the other three algorithms show that the pattern of splitting is somewhat different, i.e., when k increases by 1, some of the newly formed clusters have datapoints from ≥ 2 clusters. This shows that the association of any two datapoints that are part of the same cluster is not guaranteed to remain when k changes. However, in most cases (i.e., for a given combination of dataset + algorithm + k), only $\leq 10\%$ of datapoints show this behavior; in 4-5 cases, we found up to 30% of datapoints change cluster membership, and this often happened for higher values of k. These observations suggest that the pattern of cluster formation mimics phylogeny, especially for lower values of k.

Table 5: Contingency matrix used to show affinity propagation clustering (APC).

Protein family	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**	**Cluster 6**	**Total**
(a) APC of the Lysozyme CGCh dataset (SC=0.30)								
Lysozyme C	17	33	2	0	Not applicable (only 4 clusters)			52
Lysozyme G	0	1	0	16				17
Lysozyme Ch	0	0	51	3				54
Total	17	34	53	19				123
(b) APC of the Lysozyme dataset (SC=0.19)								
GH22	19	36	26	42	12	0	Not applicable (only 6 clusters)	135
GH24	0	0	0	1	0	30		31
GH23	0	0	0	0	2	45		47
Total	19	36	26	43	14	75		213
(c) APC of the GH5 dataset (SC=0.10)								
Cellulase	49	55	68	54	37	11	68	342
Endo-β-Mannanase	0	7	35	29	14	3	39	127
Total	49	62	103	83	51	14	107	469

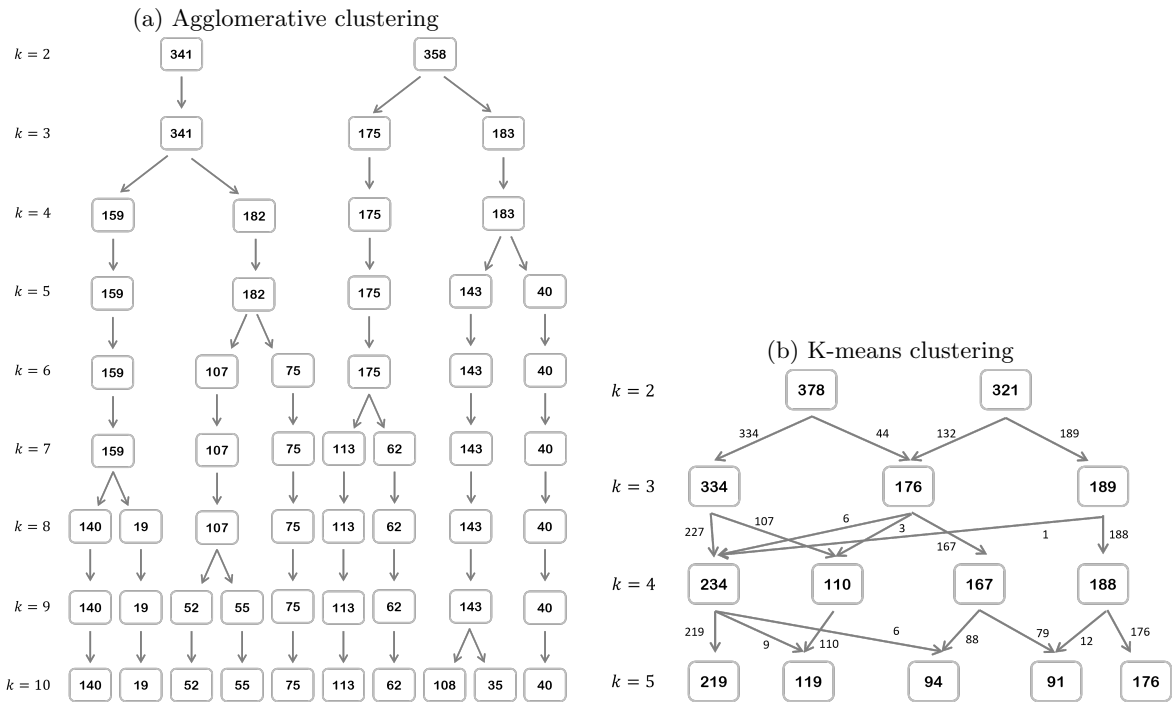

Figure 2: Split of the clusters when k value changes from 2 to 10 with examples demonstrated on a) Agglomerative clustering b) K-means clustering

Determining the optimal number of clusters for a given dataset

The number of clusters(k) to be formed by input data is user-defined for k-means, GMM, agglomerative, and spectral clustering. The optimal number of clusters for each clustering algorithm was determined by evaluating clustering quality using the SC value, with k ranging from 2-10. A higher SC value indicates better separation between clusters. Therefore, for each algorithm, we selected the k value corresponding to the highest SC, as shown in table 6. For each dataset, among the four optimal k values (one for each of the four algorithms), the clustering result corresponding to the k with the highest FMI was chosen as the final result. An analysis of these clustering results for each group is explained below and shown as contingency matrices. The contingency matrices of Lysozyme CGCh, β-Galactosidase, and GH3 datasets are shown in table 7 while other contingency matrices are shown in Supplementary_table2.

Group 1

Clusters formed by agglomerative/spectral clustering achieve the highest FMI for the three datasets of this group. The labels assigned by the clustering algorithm and the database agree very well in the case of the Xylanase and Chitinase datasets. In the case of the Lysozyme CGCh dataset, lysozyme C and lysozyme G sequences have been grouped into a single cluster. Hence, even though the number of families (as determined by database-assigned labels) is 3, we find only two clusters. It has been found that lysozyme C and lysozyme G share a common ancestor and hence retain the same fold. This suggests that, in this case, clustering is able to identify remote homologs.

Group 2

All three datasets in this group have two families, and even the clustering algorithm identified two clusters. The database-assigned and cluster labels agree completely for all sequences in the Lysozyme CaLA dataset,

Table 6: Predicted number of clusters k that corresponds to the highest\maximum (max) SC for each algorithm.

Group	Dataset	K-means		GMM		Spectral$^{\Phi}$		Agglomerative	
		k	**max SC**	**k**	**max SC**	**k**	**max SC**	**k**	**max SC**
Group 1	Lysozyme CGCh	2	0.35	2	0.35	2	0.31	2	0.34
	Xylanase	2	0.36	2	0.35	2	0.35	2	0.35
	Chitinase	3	0.18	3	0.18	2	0.12	3	0.18
Group 2	Lysozyme CaLA	2	0.37	2	0.37	2	0.37	2	0.37
	Protease	2	0.35	2	0.35	2	0.38	2	0.38
	Ferredoxin	6	0.30	6	0.30	2	0.22	5	0.32
Group 3	β-Glucosidase	2	0.24	2	0.21	10	0.14	3	0.24
	Lysozymes	10	0.25	10	0.25	10	0.21	10	0.27
	β-Galactosidase	3	0.33	3	0.33	8	0.15	3	0.33
Group 4	GH2	10	0.35	10	0.35	10	0.36	10	0.38
	GH3	3	0.25	3	0.25	2	0.26	3	0.24
	GH5	2	0.25	2	0.24	2	0.23	2	0.21
Group 5	SNARE	2	0.14	2	0.14	2	0.13	2	0.09
	GPCR1	2	0.15	2	0.34	2	0.14	2	0.12
	GPCR2	2	0.22	2	0.28	2	0.21	2	0.20

$^{\Phi}$ Spectral stands for Spectral clustering

for the majority of the sequences in the Ferredoxin dataset, but for only some sequences in the Protease dataset. In the Protease dataset, chymotrypsin and trypsin share very high similarity in their 3-D structure and function, differing only in substrate specificity. This shows that clustering algorithms do not necessarily distinguish proteins with fine-grained functional differences. In the Ferredoxin dataset, functional differences arise because of a loop region; the rest of the protein sequence and 3-D structure are the same. Clustering algorithms are able to detect this difference, and it is tempting to attribute this ability to the sequence correlation effect captured by feature engineering.

Group 3

The agreement between the database-assigned and cluster labels is not as good as it is for the datasets in Group 1 and Group 2 (except the Protease dataset). The β-Glucosidase dataset consists of two families; clustering gives three clusters. Two of the three clusters have sequences predominantly belonging to either one of the two families; the third cluster has representation from both families. For the β-Galactosidase dataset, GH35 sequences form a distinct cluster, whereas GH2 and GH42 sequences do not separate well. The Lysozyme dataset gives 10 clusters, while it has only three families. GH22 sequences are split into multiple clusters; only four sequences from the GH23 family are part of these clusters. The separation of GH23 and GH24 sequences is not quite distinct. We have not been able to assign any possible reason for this pattern of clustering.

Group 4

Datasets in this group are distinct from those in other groups: Sequences from all three datasets have the same first three digits of the EC number, i.e., 3.2.1.-. Sequences of the GH2 dataset form 10 clusters even though they have two families. GH3 and GH5 datasets also have two families, and in each of these cases, only two clusters are formed; yet there is a mismatch between cluster labels and database-assigned labels. Overall, the performance of the clustering algorithms from the viewpoint of matching with the database-assigned labels is not good.

Group 5

Datasets in this group comprise proteins without catalytic activity (no EC number). Each of the three datasets has two categories of sequences: one category is a functional family of proteins, whereas the other consists of sequences chosen using the NOT Boolean operator to ensure that none of the sequences belong to the functional family. Because of this reason, labels are given as a *functional family* and *non-functional family*. These labels were compared with cluster labels. The two labels show good agreement with each other for the SNARE and GPCR2 datasets, but not for the GPCR1 dataset. This may be attributed to the presence of transmembrane domains in both categories, i.e., GPCR and non-GPCR (TM) families. This discrepancy may be attributed to the presence of transmembrane domains that share a similar amino acid composition, i.e., rich in apolar amino acids.

Table 7: Contingency matrix corresponding to the clustering algorithm with highest FMI.

Protein family	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Total**
(a) Agglomerative clustering of the Lysozyme CGCh dataset (FMI=0.86)				
Lysozyme C	52	0	Not applicable (only 2 clusters)	52
Lysozyme G	17	0		17
Lysozyme Ch	0	54		54
Total	69	54		123
(b) GMM of the β-Galactosidase dataset (FMI=0.86)				
GH2	74	0	43	117
GH35	5	283	7	295
GH42	39	0	44	83
Total	118	283	94	495
(c) Spectral clustering of the GH3 dataset (FMI=0.62)				
β-Glucosidase	162	42	Not applicable (only 2 clusters)	204
1,4-β-Xylosidase	89	10		99
Total	251	52		303

Discussion

In this study, we considered nine available algorithms available in the scikit-learn platform [30] to evaluate their suitability in segregating protein families reflecting functional differences. K-means, Gaussian mixture model (GMM), agglomerative, and spectral clustering were chosen because of their flexibility to allow a user-defined number of clusters, denoted as k. We also gave special consideration to affinity propagation clustering (APC) despite its non-user-defined k, as it demonstrated the ability to segregate proteins effectively.

The remaining four algorithms mean-shift, DBSCAN, OPTICS, and BIRCH were excluded from consideration primarily because they do not allow user-defined k. These algorithms either produced a single cluster or an excessive number of clusters, where most data points from different classes fell into a single cluster. This outcome can be attributed to the default parameters, which may not align with the characteristics of our datasets. Furthermore, the lack of a widely accepted metric to assess intrinsic performance, especially when labels are not provided a priori, limits their applicability in this study.

To further assess clustering performance, we studied the pattern of cluster formation for values of k ranging from 2 to 10 for k-means, GMM, agglomerative, and spectral algorithms. Notably, agglomerative produced dendrograms resembling phylogenetic trees, which align with the phylogenetic relationships of the protein families in our dataset. Therefore, the splitting of clusters with increasing k values can be seen as a depiction of evolutionary divergence. However, in the case of k-means, GMM, and spectral clustering, the pattern differed; when k increased, some newly formed clusters contained data points from two or more original clusters. This behavior, although rare ($\leq 10\%$ of the data points), was occasionally observed at higher rates in 4-5 cases, with up to 30% of the data points affected.

We employed three intrinsic metrics: silhouette coefficient (SC), Calinski-Harabasz (CH) index, and Davies-Bouldin (DB) index to determine the optimal k for each algorithm. However, only SC proved suitable for this study, and we discarded the CH and DB indices due to their previously discussed limitations. Using SC, we identified the optimal k for each algorithm on each dataset, selecting the value with the highest SC score as the optimal solution for further analysis.

When comparing the optimal k and SC scores across the datasets, we found that the majority of cases exhibited consensus across the four clustering algorithms. Notably, in cases where consensus was absent, the discrepancy was largely due to the underperformance of spectral clustering, which often produced k values that differed when the SC value showed a ratio of approximately 2:3 difference from the other three algorithms for each dataset. Additionally, k-means, GMM, and agglomerative generally produced identical \similar SC values, whereas spectral clustering consistently underperformed. This raises the possibility that SC may be biased against spectral clustering or that spectral clustering is not well-suited for our datasets and features.

The suspicion of SC being algorithm-dependent prevented us from selecting a single optimal algorithm and k for each dataset. Instead, we carried out all four algorithms with their respective optimal k forward for further analysis. For the final evaluation, we utilized the highest Fowlkes-Mallows index (FMI) from these four algorithms for each dataset and plotted contingency matrices to visually assess performance.

Clustering methods (i.e., a combination of pseudo amino acid composition (PAAC) and clustering algorithms) successfully segregated the datasets in Group 1, Group 2 (except for the protease dataset), and Group 5 (except for GPCR1). Group 3 exhibited partial segregation, while Group 4 posed significant challenges. In the few exceptions where clustering methods underperformed, the discrepancies were attributed to specific characteristics of the datasets. For instance, the failure to fully segregate the protease dataset in Group 2 and all the datasets in Group 4 may be due to fine-grained functional differences within the datasets. Similarly, the challenge with the GPCR1 dataset in Group 5 may arise from the presence of transmembrane domains in both classes, which share a similar amino acid composition dominated by apolar residues.

These observations suggest that clustering methods are less effective in distinguishing fine-grained functional differences within protein families. However, the ability of the methods to differentiate Group 1 datasets, where each class was confined to a single glycoside hydrolase (GH) family, suggests that clustering is effective in distinguishing GH families despite functional similarities. This conclusion was further reinforced by similar results in Group 3, although the separation was only partial, potentially due to the presence of similar protein folds, a factor not observed in Group 1. Additionally, clustering methods are failing to segregate GH families in Group 4. Based on these results, clustering methods show promise as a tool for labeling unknown proteins, particularly for distinguishing GH families with fine-grained functional similarities.

Lastly, for affinity propagation clustering (APC), we did not apply either intrinsic or extrinsic evaluation metrics. The intrinsic metric was not used since the optimal number of clusters k is determined by the algorithm itself, and the extrinsic metric was not applied because APC produced more clusters than expected. This resulted in a high number of false negatives and a lower FMI score. Nonetheless, the results obtained from APC were largely consistent with the trends observed in the combination of other algorithms.

Conclusion

This study aimed to assess the ability of clustering algorithms to segregate functionally distinct protein sequences using pseudo amino acid composition (PAAC) as the feature set. Our results demonstrate that PAAC is an effective feature for this task, particularly in distinguishing protein families with shared functional similarities.

We find that the outcome of clustering methods is critically dependent on four factors: (i) type, nature, and size of data, (ii) feature vector, (iii) clustering algorithms, and (iv) parameters used by algorithms. For a given task, the type of data is fixed: in this case, the data are protein sequences. In the present study,

the nature of the data refers to sequence variability and the relationship between sequence and function. The latter is quite subtle and is not fully understood, especially when the focus is on fine-grained functional differences (e.g., substrate specificity). In an ideal situation, unsupervised learning algorithms ought to be assessed by intrinsic metrics, but these can be biased by the spread of data in the feature vector space. Thus, there is no universally applicable metric, i.e., a metric that is suitable for all combinations of types of data, feature vector, and clustering algorithm. Hence, extrinsic metrics are often used, and these also have limitations. Consequently, there is no way to find the most appropriate combination of feature vector, algorithm, and parameter values for a given dataset. To address these challenges, in the present study, we opt for a feedback-driven iterative approach to find the optimal combination, and we used both intrinsic (silhouette coefficient) and extrinsic (Fowlkes-Mallows index) metrics.

Among the clustering algorithms tested, agglomerative performed slightly better than k-means and Gaussian mixture model (GMM) as assessed by the silhouette coefficient and Fowlkes-Mallows index. This could be due to the ability of agglomerative to mimic phylogenetic relationships. However, k-means and GMM also showed a good capacity to reflect phylogeny to a large extent, and all three algorithms performed comparably well. While one may outperform the others in specific cases, we recommend using all three algorithms (k-means, GMM, and agglomerative) across a range of k values, choosing the optimal k based on the SC. For datasets containing proteins with known functions, the FMI can be used to select the best clusters and transfer annotations to unannotated proteins. However, if all data points are of unknown function, SC will be the sole metric for guiding cluster selection.

In the case of affinity propagation clustering (APC), while it performed well overall, its tendency to split homologous proteins into multiple clusters indicates higher sensitivity. For a more comprehensive analysis, it is advisable to use a combination of k-means, GMM, agglomerative, and APC and compare the results for consistency. This approach provides a robust framework for the clustering and functional annotation of protein sequences, especially when dealing with unannotated datasets.

Lastly, the implementation of the clustering method for protein sequences is available at the GitHub repository linked below. It includes detailed steps for preprocessing, feature extraction, clustering, and evaluation, with all results presented in a Jupyter Notebook. https://github.com/RakeshBusi/Clustering

Supplementary data

1. Supplemantary_figure1
2. Supplementary_table1
3. Supplementary_table2

Conflicts of Interests

None declared.

References

[1] Stephen PH Alexander, Arthur Christopoulos, Anthony P Davenport, Eamonn Kelly, Alistair A Mathie, John A Peters, Emma L Veale, Jane F Armstrong, Elena Faccenda, Simon D Harding, et al. The concise guide to pharmacology 2023/24: G protein-coupled receptors. *British journal of pharmacology*, 180:S23–S144, 2023.

[2] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[3] Chunyan Ao, Shihu Jiao, Yansu Wang, Liang Yu, and Quan Zou. Biological sequence classification: A review on data and general methods. *Research*, 2022:0011, 2022.

[4] Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M Pérez, and Iñigo Perona. An extensive comparative study of cluster validity indices. *Pattern recognition*, 46(1):243–256, 2013.

[5] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[6] Alessia Auriemma Citarella, Luigi Di Biasi, Michele Risi, and Genoveffa Tortora. Snarer: new molecular descriptors for snare proteins classification. *BMC bioinformatics*, 23(1):1–20, 2022.

[7] Timothy L Bailey, James Johnson, Charles E Grant, and William S Noble. The meme suite. *Nucleic acids research*, 43(W1):W39–W49, 2015.

[8] Alex Bateman, Maria-Jesus Martin, Sandra Orchard, Michele Magrane, Shadab Ahmad, Emanuele Alpi, Emily H Bowler-Barnett, Ramona Britto, Austra Cukura, Paul Denny, et al. Uniprot: the universal protein knowledgebase in 2023. *Nucleic acids research*, 51(D 1):D523–D531, 2023.

[9] Khodeza Begum, Jonathon E Mohl, Fredrick Ayivor, Eder E Perez, and Ming-Ying Leung. Gpcr-pendb: a database of protein sequences and derived features to facilitate prediction and classification of g protein-coupled receptors. *Database*, 2020, 2020.

[10] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[11] Rosalin Bonetta and Gianluca Valentino. Machine learning techniques for protein function prediction. *Proteins: Structure, Function, and Bioinformatics*, 88(3):397–413, 2020.

[12] Emmanuel Boutet, Damien Lieberherr, Michael Tognolli, Michel Schneider, and Amos Bairoch. Uniprotkb/swiss-prot: the manually annotated section of the uniprot knowledgebase. In *Plant bioinformatics: methods and protocols*, pages 89–112. Springer, 2007.

[13] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[14] Kuo-Chen Chou. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, 43(3):246–255, 2001.

[15] Kuo-Chen Chou et al. The significant and profound impacts of chou's pseudo amino acid composition or pseaac. *Natural Science*, 12(09):647, 2020.

[16] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[17] Elodie Drula, Marie-Line Garron, Suzan Dogan, Vincent Lombard, Bernard Henrissat, and Nicolas Terrapon. The carbohydrate-active enzyme database: functions and literature. *Nucleic acids research*, 50(D1):D571–D577, 2022.

[18] Sean R. Eddy. Profile hidden markov models. *Bioinformatics (Oxford, England)*, 14(9):755–763, 1998.

[19] Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569, 1983.

[20] Peter Goettig, Hans Brandstetter, and Viktor Magdolen. Surface loops of trypsin-like serine proteases as determinants of function. *Biochimie*, 166:52–76, 2019.

[21] Bernard Henrissat. A classification of glycosyl hydrolases based on amino acid sequence similarities. *Biochemical journal*, 280(2):309–316, 1991.

[22] Lakshmanan K Iyer and Pradman K Qasba. Molecular dynamics simulation of α-lactalbumin and calcium binding c-type lysozyme. *Protein engineering*, 12(2):129–139, 1999.

[23] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[24] Lisa Jeske, Sandra Placzek, Ida Schomburg, Antje Chang, and Dietmar Schomburg. Brenda in 2019: a european elixir core data resource. *Nucleic acids research*, 47(D1):D542–D549, 2019.

[25] Eugene V Koonin. Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.*, 39(1):309–338, 2005.

[26] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[27] Catriona Miller, Theo Portlock, Denis M Nyaga, and Justin M O'Sullivan. A review of model evaluation metrics for machine learning in genetics and genomics. *Frontiers in Bioinformatics*, 4, 2024.

[28] Jelili Oyelade, Itunuoluwa Isewon, Funke Oladipupo, Olufemi Aromolaran, Efosa Uwoghiren, Faridah Ameh, Moses Achas, and Ezekiel Adebiyi. Clustering algorithms: their application to gene expression data. *Bioinformatics and Biology insights*, 10:BBI–S38316, 2016.

[29] Gbeminiyi John Oyewole and George Alex Thopil. Data clustering: application and trends. *Artificial Intelligence Review*, 56(7):6439–6475, 2023.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[31] Burkhard Rost. Twilight zone of protein sequence alignments. *Protein engineering*, 12(2):85–94, 1999.

[32] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[33] Vinzent Schulz, Somsuvro Basu, Sven-A Freibert, Holger Webert, Linda Boss, Ulrich Mühlenhoff, Fabien Pierrel, Lars-O Essen, Douglas M Warui, Squire J Booker, et al. Functional spectrum and specificity of mitochondrial ferredoxins fdx1 and fdx2. *Nature Chemical Biology*, 19(2):206–217, 2023.

[34] Jaya Srivastava, Ritu Hembrom, Ankita Kumawat, and Petety V Balaji. Searching sequence databases for functional homologs using profile hmms: how to set bit score thresholds? *bioRxiv*, 2021.

[35] Daniel Ungar and Frederick M Hughson. Snare protein structure and function. *Annual review of cell and developmental biology*, 19(1):493–517, 2003.

[36] Kai-Chao Yang, Chang-Hsin Yu, and Jia-Shung Wang. Robust affinity propagation using preference estimation. 2012.

[37] Hui Yin, Amir Aryani, Stephen Petrie, Aishwarya Nambissan, Aland Astudillo, and Shengyuan Cao. A rapid review of clustering algorithms. *arXiv preprint arXiv:2401.07389*, 2024.

[38] Mingyang Zhang, Ting Chen, Xun Lu, Xiaobing Lan, Ziqiang Chen, and Shaoyong Lu. G protein-coupled receptors (gpcrs): advances in structures, mechanisms, and drug discovery. *Signal Transduction and Targeted Therapy*, 9(1):88, 2024.

[39] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.