scientific reports

Check for updates

OPEN

Identifying representative sequences of protein families using submodular optimization

Ha Nguyen[1], Hung Nguyen[1], Phuong Nguyen[1], Anh N. Luu[3], David C. Cantu[2] & Tin Nguyen[1]✉

Identifying representative sequences for groups of functionally similar proteins and enzymes poses significant computational challenges. In this study, we applied submodular optimization, a method effective in data summarization, to select representative sequences for thioesterase enzyme families. We introduced and validated two algorithms, Greedy and Bidirectional Greedy, using curated protein sequence data from the ThYme (Thioester-active enzYmes) database. Both algorithms generated sequence subsets that preserved completeness (inclusion of all known family sequences) and specificity (accurate family representation). The Greedy algorithm outperformed the Bidirectional Greedy algorithm and other methods, particularly in reducing redundancy. Our study offers an efficient approach for identifying representative protein sequences within families that have significant sequence similarity, likely to deliver results close to theoretical optima in polynomial time, with the potential to improve the selection and optimization of representative sequences in protein databases.

Keywords Protein analysis, Representative sequences, Submodular optimization

Organizing proteins into families based on shared structure, function and catalytic mechanism, is a fundamental aspect of comparative and evolutionary genomics[1]. This classification, primarily derived from amino acid sequences, enables researchers to predict tertiary structures, identify catalytic residues, and elucidate enzymatic mechanisms for particular sequences within entire protein families[2,3]. Additionally, classifying enzyme sequences in protein families enables the inference of structure and function for uncharacterized sequences in organisms of interest by leveraging knowledge from well-studied proteins within the same family[4,5]. A critical step in managing and analyzing protein and enzyme families is the identification of *representative sequences*. This process involves reducing a large set of protein sequences into a small subset that effectively captures the diversity and essential characteristics of all sequences in the entire family. The selection of representative sequences aims to strike a balance between maintaining completeness (ensuring all significant variations within the family are represented) and curtailing redundancy (minimizing overrepresentation of highly similar sequences). Such sequence subsets are crucial for various applications in biological and biomedical research, such as aiding in modeling protein structures in structural biology, which is essential for understanding protein function and interactions, and facilitating drug design[6]; enabling the identification and quantification of proteins in complex samples using mass spectrometry[7,8]; or being used in enzyme engineering and synthetic biology to develop novel biological systems and enhance industrial applications[9].

Representative sequences are particularly relevant in the context of specialized enzyme databases. Our group recently updated and renewed the ThYme (Thioester-active enzYmes) database, an open-access resource that categorizes thioesterase (TE) enzymes into 35 distinct families[4]. ThYme also includes sequences, classified into families, of other enzymes involved in the fatty acid synthesis cycle, and/or active with substrates that include thioesters such as acyl transferases[5]. We constructed each enzyme family in ThYme around a set of representative sequences, which we use as the foundation for populating the families. However, selecting the most appropriate representative sequences for each specific family presents a substantial challenge. For a dataset containing n sequences, one must theoretically evaluate all 2^n possible subsets to find the optimal representation. This becomes computationally infeasible as n increases. Many existing approaches introduce a specific threshold (e.g. % identity or similarity) to define sequence representation[10–17]. These approaches typically rely on heuristic algorithms to find the smallest subset of the ground set that represents all sequences. However, these methods

[1]Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, USA. [2]Department of Chemical and Materials Engineering, University of Nevada, Reno, Reno, NV 89557, USA. [3]Kenneth P. Dietrich School of Arts & Sciences, University of Pittsburgh, Pittsburgh, PA 15260, USA. ✉email: tinn@auburn.edu

have significant limitations: they often disregard all similarities below the predefined threshold, potentially overlooking important relationships, and there is no theoretical guarantee that these heuristic algorithms produce a subset close to the optimal representative set of sequences.

Researchers have proposed a framework using submodular optimization to address these challenges, an approach that has shown remarkable success in data summarization[18,19]. In a recent study, Libbrecht et al.[20] demonstrated that this framework can yield a concise yet comprehensive representation of data, offering particular benefits in handling the redundancy common in sequence datasets. However, the scientific community has not adopted this approach for selecting representative sequences. Several factors contribute to this limited adoption: (i) the NP-hard complexity of subset selection; (ii) the non-trivial task of defining submodular functions for sequencing data and developing algorithms to use these functions; and iii) the lack of straightforward methods to test algorithm efficiency or validate the completeness and non-redundancy of results.

In this article, we present a ranking-based submodular optimization framework to select representative sequences for protein families in the ThYme database where sequences within each family share high sequence similarity, and sequences in different families have low similarity. We aim to demonstrate how submodular optimization can facilitate the creation of sequence subsets that maintain the integrity of protein and enzyme families. Our approach ensures the inclusion of all known sequences within an enzyme family (completeness) while correctly identifying sequences in their appropriate families (specificity). We explore how this optimization framework excels in distributing known sequences across different enzyme families, providing a more accurate and informative data representation. We apply our framework to enhance the selection process for representative sequences of thioesterase families in the ThYme database, which currently relies on an ad hoc procedure with expert curation. By implementing this methodology, we seek to improve the accuracy, efficiency, and robustness of representative sequence selection. We also applied the submodular algorithms to two protein families in the MEROPS (peptidases) and ESTHER (esterases) databases.

Methods

Representative protein sequences selection

The selection of representative sequences is critical in protein sequence analysis, with significant implications for molecular biology and bioinformatics. Researchers choose representative sequences for each protein family, defining them as a subset of sequences within that family. These carefully selected sequences serve multiple crucial functions: populating protein families in databases[4], classifying newly discovered protein sequences[21], categorizing 3D protein structures[21], and guiding target selection in structural genomics initiatives[22]. The objective of selecting representative sequences is to identify the smallest subset that fulfills a predefined criterion, typically described as "maximum coverage with minimum redundancy"[9] from a finite set of protein sequences, referred to as the "ground set". Maximum coverage ensures that each sequence in the family is represented by at least one sequence in the representative set, while minimum redundancy guarantees that no two proteins in the representative set exceed a predefined sequence identity threshold.

The primary challenge in selecting a representative set from a large dataset lies in its combinatorial complexity. Even for a relatively small set of 100 sequences, evaluating all $2^{100} \approx 10^{30}$ possible combinations to find the optimal subset is computationally infeasible. Hobohm et al.[9] developed one of the earliest systematic approaches to construct a representative set of non-redundant protein sequences. This algorithm calculates similarity between sequences, using an alignment score or similar metrics, and sets a specific threshold (e.g. % similarity) to determine if two proteins are neighbors. Given a list of candidate proteins and a list of neighbors for each of the proteins, the algorithm removes one protein at a time until those remaining in the list have no neighbor. Subsequent algorithms, including CD-HIT[23], PISCES[24], MMSEQS[25] and UCLUST[26], have further developed this concept, typically sorting protein sequences by length and sequentially adding sequences to the representative set if no existing member exceeds a specified similarity threshold. However, these approaches have several limitations: (i) reliance on greedy strategies often leads to suboptimal solutions, (ii) ignoring all similarities below the specified cutoff may overlook important relationships, and iii) lack of control over the size of the representative set as it is possible to include all sequences or only one depending on the predefined threshold.

In this study, we apply classical submodular optimization to the challenge of selecting representative protein sequences. This approach has demonstrated remarkable success in diverse fields, including the selection of representative subsets in text document analysis[18,19,27], speech recognition[28–30], machine translation[31], and image analysis[32]. However, its application in sequence analysis remains limited. A recent study by Libbrecht et al.[20] applied this approach to choose non-redundant representative subsets of protein sequences, demonstrating that submodular optimization achieves the best possible results in polynomial time. In the following sections, we provide a detailed description of how submodular optimization can be effectively utilized for selecting representative protein sequences, exploring its potential to overcome the limitations of previous methods and enhance the accuracy and efficiency of protein sequence analysis.

Mathematical description of submodular optimization

Mathematical notations

Capital letters denote sets and lowercase letters to denote items in the sets (i.e., sequences).

- $\{a, b, c\}$ denotes a set with items a, b and c.
- $\emptyset$ is the empty set.
- $A \cup B$ is the union of A and B.
- $A \setminus B$ is the set of all items in A but not in B.
- $a \in A$ means a is an element of A.

- $A \subseteq B$ means A is a subset of B.

Submodular function

Figure 1 illustrates the application of submodular optimization in identifying a set of representative protein sequences. Let $S = \{s_1, s_2, \ldots, s_n\}$ denote a finite set of protein sequences, and $f : 2^S \to \mathbb{R}$ represent a function over subsets of S. The function f evaluates a set of elements and outputs a real value quantifying the quality of that set. We assume that f is monotonically increasing. For any subset $X \subseteq S$, we define the marginal improvement of adding an element $s \in S \setminus X$ to set X as $f(X \cup \{s\}) - f(X)$. The function f is considered submodular and normalized if and only if it satisfies the following three conditions:

$$f(\emptyset) = 0 \tag{1}$$

$$f(X \cup \{s\}) - f(X) \geq 0, \forall s \in S \setminus X, X \subseteq S \tag{2}$$

$$f(X \cup \{s\}) - f(X) \geq f(Y \cup \{s\}) - f(Y), \forall X \subseteq Y \subseteq S, s \notin Y \tag{3}$$

Equation (1) establishes that the function has a value of zero for an empty set. Equation (2) ensures the function is monotonically non-decreasing, meaning that adding a new sequence to the representative set can only maintain or increase its value. Equation (3) describes a crucial property of submodular functions: as we select more points from the ground set, the incremental gain decreases. In the context of selecting representative sequences, this property reflects that the benefit of adding a given protein sequence diminishes when the representative set already contains similar sequences. Our objective is to identify a subset $R \subseteq S$ that maximizes the value of f, thereby obtaining an optimal set of representative sequences that efficiently captures the diversity within the protein family.

Submodular functions form a broad class of functions with applications across various domains. Notable examples include the weighted coverage function[33], rank function of a matroid[34], entropy[35], mutual information[36], and cut capacity[37]. In this study, we employ a classical submodular function known as the facility location function[38] to select representative sequences for protein families. Facility location functions are versatile submodular functions that, when maximized, select examples that effectively represent the data space. These functions optimize the pairwise similarities between points in the dataset and their nearest neighbors, ensuring that the chosen subset accurately reflects the overall data distribution. The general form of the facility location function is:

$$f(R) = \frac{1}{|S|} \sum_{s \in S} \max_{r \in R} \phi(s, r) \tag{4}$$

where f denotes the facility location function, S is the ground set of all protein sequences, $R \subseteq S$ is the selected subset of representative sequences, s and r are individual sequences in the ground set, and $\phi(s, r)$ represents the similarity measure between sequences.

This function satisfies all properties defined in Eqs. (1)–(3): (1) normalization: $f(R) = 0$ when $R = \emptyset$; (2) monotonicity: f is monotonically non-decreasing; and (3) submodularity: adding sequences similar to those already in R yields diminishing returns in the value of the function. In our context, maximizing this facility location function will select a subset of sequences that represents the entire protein family. These chosen sequences serve as query sequences for retrieving related proteins from the database, efficiently capturing the diversity within the family while minimizing redundancy.

Similarity function

The similarity function must be non-negative, with higher values indicating greater similarity between sequences. In this study, we define the similarity between a pair of sequences, $\phi(s, r)$, as the fraction of matching residues.

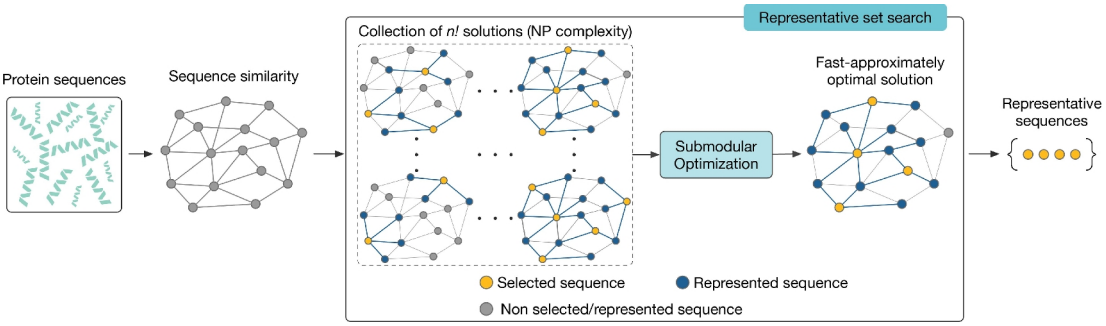

Fig. 1. Selecting representative sequences using submodular optimization. The objective is to identify a subset of sequences that effectively represent the diversity within a given protein family. While an exhaustive search would require evaluating 2^n possible subsets (where n is the number of sequences), rendering the problem NP-hard, submodular optimization offers an efficient approach to find a near-optimal solution in polynomial time.

This metric, widely used in protein sequence analysis, can be efficiently calculated using BLAST (Basic Local Alignment Search Tool) tools. For our analysis, we employed BLAST with its default settings optimized for alignment tasks. These settings include: 1) an expect value for saving hits set at 0.05 (*evalue*), 2) retaining a maximum of 100 aligned sequences (*max_target_seqs*), 3) a word size of 3 (*wordsize*), and 4) the BLOSUM62 substitution matrix. We observed that modifying these parameters did not significantly affect the similarity measures returned by BLAST. In cases where BLAST does not report a similarity between a pair of sequences, we assign a similarity value of 0.

Optimization algorithms

Submodular functions arise in numerous applications, making the study of submodular optimization both natural and crucial. While extensive research has been conducted on minimizing submodular functions[37,39], our focus lies on maximizing these functions in the context of representative sequence selection. Specifically, we aim to solve problems of the form:

$$\max f(R), \text{ subject to some contraints in } R \subseteq S \tag{5}$$

The simplest optimization approach involves *cardinality constraints*, where we require that $|R| \leq k$ for some positive integer k. In our application using the facility location function, this translates to finding the k best representative sequences for a given protein family. However, even this seemingly simple approach is computationally challenging and classified as NP-hard. Fortunately, efficient approximation algorithms for submodular functions exist, capable of finding solutions guaranteed to be close to the optimal[33,40]. These algorithms provide a balance between computational feasibility and solution quality. In this study, we employ two such algorithms, as described hereinafter.

The first algorithm we employ is the *Greedy algorithm*, which guarantees solutions that are at least $(1 - \frac{1}{e}) \approx 63.2\%$ of the optimal value[41], where e is the base of natural logarithm. Algorithm 1 presents the pseudocode for this approach. The algorithm starts by initializing an empty representative set and then repeatedly identifies new representative sequences through the following steps. Particularly, for each sequence in the remaining set, the algorithm calculates the marginal improvement in the submodular function (Equation 4) when adding the sequence to the current representative set. Next, it selects the sequence that provides the largest improvement and adds it to the representative set. The algorithm then removes the selected sequence from the remaining set and eliminates other sequences in the remaining set that are identical to the selected sequence (above a 90% threshold). The algorithm repeats this process until no sequences remain in the set.

$R_0 \leftarrow \emptyset, f(R_0) = 0, S_0 \leftarrow S$
while $length(S_i) > 0$ **do**
  $e^* = \arg\max_{e \in S_i \setminus R_i} (f(R_i \cup \{e_i\}) - f(R_i))$
  $R_{i+1} = R_i \cup \{e_i\}$
  $S_i = S_{i-1} \setminus \{e_i, e_j : \phi(e_i, e_j) \geq 90\%\}$
end while
Return R

Algorithm 1. Pseudocode of the Greedy algorithm.

The second algorithm we employ is the *Bidirectional Greedy algorithm*, which guarantees solutions that are at least $\frac{1}{2}$ of the optimal value[42]. Algorithm 2 presents the pseudocode for this approach. Unlike the standard Greedy algorithm, this method introduces randomization and maintains two sets: a "growing set" initialized as empty, and a "shrinking set" initialized as the complete ground set. The algorithm iteratively considers each sequence, deciding whether to add it to the growing set or remove it from the shrinking set based on which action yields the greater gain in the objective function (Eq. 4). If adding the sequence provides a greater gain, the algorithm adds it to the growing set; otherwise, it removes the sequence from the shrinking set. The algorithm terminates when the growing and shrinking sets converge to identical sets, either of which represents the final solution.

$A \leftarrow \emptyset, B \leftarrow S$
for $i \in \{1, \ldots, n\}$ **do**
  $\alpha_i = f(A_{i-1} \cup \{e_i\}) - f(A_{i-1})$
  $\beta_i = f(B_{i-1} \setminus \{e_i\}) - f(B_{i-1})$
  if $\alpha_i / (\alpha_i + \beta_i) > 0.5$ **then**
    $A_i = A_{i-1} \cup \{e_i\}, B_i = B_{i-1}$
  else
    $A_i = A_{i-1}, B_i = B_{i-1} \setminus \{e_i\}$
  end if
end for
Return $A_n (= B_n)$

Algorithm 2. Pseudocode of the bidirectional Greedy algorithm.

We significantly enhance the performance of these algorithms by executing them multiple times to obtain the optimal set with the highest score. We propose a ranking strategy that combines the results from multiple runs of each algorithm to select the optimal set. For each run, we shuffle the ground set, generate a representative set, and assign ranks to the sequences based on their inclusion order (lower rank indicates a better sequence). We then iteratively select the sequences that appear most frequently and hold the lowest ranks, adding them to the final set. We continue this addition process until a BLAST search using the selected sequences as query sequences successfully retrieves all members of the ground set. We configure the BLAST parameters *max_target_seqs* and *evalue* with default values of 999999 and $1e-07$, respectively.

Results

Protein sequences data

We evaluated our submodular optimization approach for selecting representative protein sequences using data from thioesterase (TE) enzyme families. We downloaded protein sequence data in FASTA format from the ThYme database that have been developed and maintained by our group (https://thyme.engr.unr.edu/v2.0/)[4]. In the ThYme database, we categorized TE enzymes into 35 families based on their structural similarity, function, and catalytic mechanism (Fig. 2). Enzymes with different structural folds catalyze thioesterase function. Most TEs have either a HotDog fold or an alpha/beta-Hydrolase fold; however, enzymes with the NagB, SGCH, Lactamase, Beta-hairpin/TIM barrel folds can also perform thioesterase function[4]. Enzymes families within each fold are more closely related to each other than to families with a different structural fold. Our recent review on TE enzymes, which describes in detail how the TE families were defined based on sequences similarity, shows phylogenetic trees of how the TE families within the HotDog and alpha/beta-Hydrolase folds are related to each

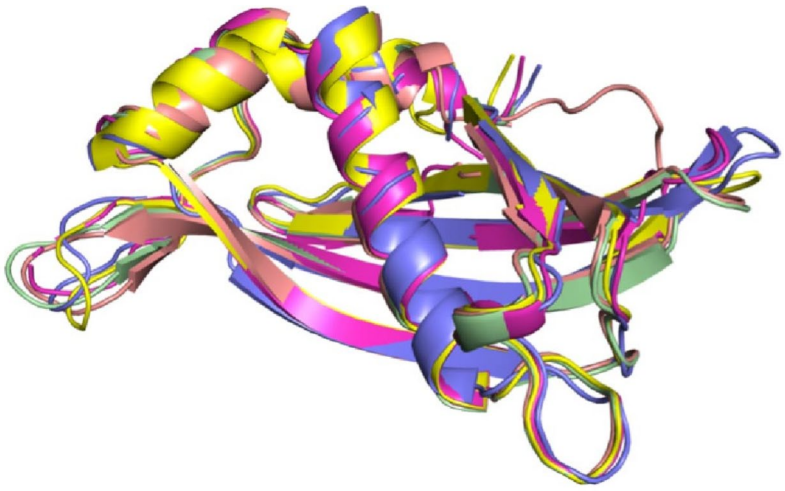

Fig. 2. Five superimposed enzymes in TE15: 5PVJ (*Homo sapiens*) - pink, 2W3X (*Micromonospora echiospora* - yellow), 2XEM (*Micromonospora chersina* - orange), 2XFL (*Micromonospora chersina* - green), and 4I4J (*Streptomyces globisorus* - blue) all have very similar tertiary structures even though they originate in different organisms.

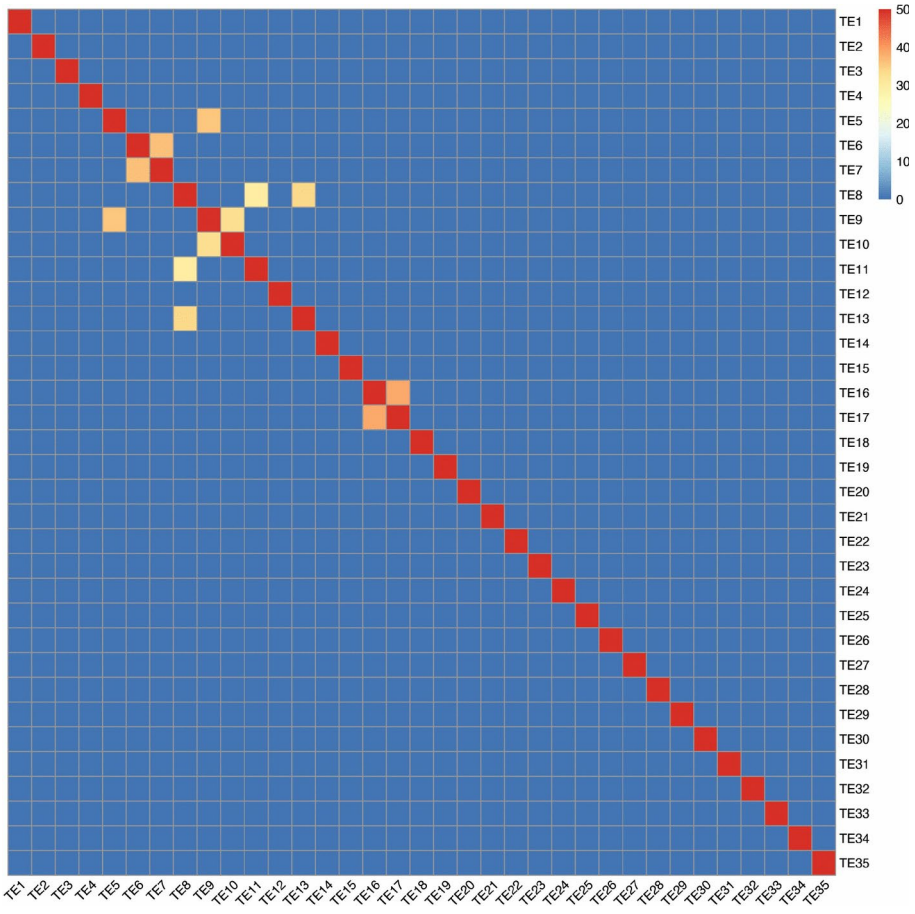

Fig. 3. Sequence similarity among the TE families. We performed pairwise sequence alignments on experimentally verified sequences from 35 TE families using blastp, with percentage identity used as a measure of sequence similarity. For each pair of families, we calculated inter-family similarity by averaging the percentage identities of the pairs consisting of one sequence from each family. Among different TEs, sequences come from different structural folds, so the TE families are very different to others. Within each family, sequence similarity is high.

other. The phylogenetic and structural diversity of the TEs ensures that the submodular optimization approach works for different enzymes/protein families (Fig. 3), which is a reason why TEs were chosen to develop the submodular optimization approach to identify representative sequences.

Each TE family is based on at least one experimentally verified sequence which serves as a representative sequence. Families are populated with protein BLAST using the catalytic domain of the representative sequences as a query. Families are verified to have nearly identical tertiary enzyme. This results in that ThYme families have approximately 15-30% sequence similarity, which corresponds to narrow subfamilies of larger protein families based on sequence profiles for example in the Pfam database[43]. Among different TEs, sequences come from different structural folds, so the TE families are very different to others, as shown in Fig. 3. Accordingly, to perform the representative sequences selection using submodular optimization, we created an initial ground set of sequences for each TE family, including only unique and experimentally characterized sequences. We identified these sequences using the "Evidence at Protein Level" indicator in UniProt, as clearly marked in the ThYme database. We excluded families (TE5, TE12, TE19, TE24, TE28, TE32, TE33, and TE35) with fewer than three sequences in their ground set from further analysis. In total, we obtained 737 sequences in the ground set across all selected families, with each family containing between four and one hundred sequences, as shown in Table 1.

The main goal of this article is to introduce the two greedy algorithms based on submodular optimization to replace the ad hoc procedure with submodular optimization in the ThYme database. We are optimistic that future algorithms based on submodular optimization can be developed for the purpose of finding representative sequences for many other protein families. To demonstrate the potential of this new direction, we evaluate the approach using sequence data from a carboxylesterase (CE) enzyme family and the peptidase family A1 (A1A). We retrieved 12,277 sequences of the CE family from the ESTHER database[44] and 13,847 sequences of the A1A family from the MEROPS database[45]. Unlike TE families in which know the sequences that have been experimentally characterized, we used all sequences for the CE and A1A families as available in MEROPS

Family	Ground set size	Greedy	Bidirectional Greedy	CD-HIT	MMseqs2	Ad hoc w/ manual curation
TE1	6	P32316	Q9HTC2, P83773, B3EY95, Q7MVN7, P32316	P83773, P32316	Q9HTC2, P83773, B3EY95, Q7MVN7	P32316
TE2	44	Q86TX2	G3V4F2, Q9QYR7, A0A287B8D4, A2AKK5, A0A287A758, A0A287BCT2, D3ZIQ1, A0A8I6A6H9, O55137, Q9QYR9, Q8BWN8, Q8BGG9, F1SSB1	Q9QYR7, A0A287A758, A0A287BCT2, A0A8I6A6H9, Q8BWN8	G3V4F2, A0A287BCT2, Q8BGG9	Q86TX2
TE3	8	A0A4P1LYH5	A0A1Z1F9L9, A0A4P1LYH7, A0A4P1LYH6, Q07792, Q9HZY8, P0ADA1, A0A4P1LYH5	A0A1Z1F9L9, A0A4P1LYH7, Q07792, Q9HZY8, P0ADA1	A0A1Z1F9L9, Q07792	P0ADA1, A0A4P1LYH5
TE4	29	Q95Q68, Q73Z74, A0A3E2MQQ7, O06135, P41903, Q9U1Q5, Q19781, P58137, B3H5Z2, A0A0M3KKU4	A0A7U8YAW7, Q73Z74, A0A3E2MQQ7, P41903, F4HU51, B3H5Z2, Q8VHK0, A0A0M3KKU4	Q73Z74, P41903, Q8VHK0	Q73Z74, A0A3E2MQQ7, P41903, F4HU51, Q8VHK0, A0A0M3KKU4	P41903, O14734, Q73Z74
TE6	33	Q814K4, A0A5P8YGN6, Q9DBK0	A0A0M0KCI4, A0A8J0V2L8, A0A2U0QTN1, Q8WXI4, A0A287A7V2	A0A0M0KCI4, Q6GM80, A0A8J0V2L8, A0A2U0QTN1, Q8WXI4	A0A0M0KCI4, Q6GM80, A0A2U0QTN1	P44886, Q8WXI4, Q6ZUV0, A1KUS8, A0A0H3K033, A0A5P8YGN6
TE7	28	Q7T175, Q23044, H7C5Q2	Q6AWX1, Q9V9W4, S4TF94, Q95TK5, Q94245	Q6AWX1, Q9V9W4, S4TF94, Q95TK5, Q94245	Q6AWX1, Q9V9W4, Q94245	Q9Y305
TE8	15	Q18187	Q9VZZ6, P34419, F1RUE0, A0A0P0WAD4, A0A178W7W7, A9ULW5	P34419, F1RUE0, A0A0P0WAD4, A0A178W7W7, A9ULW5	Q9VZZ6, P34419, F1RUE0, A0A0P0WAD4, A0A178W7W7	Q9NPJ3
TE9	7	P44679, A0A1P8AM78	A0A0H3M6V9, Q9C7I5, P44679, P0A8Z5, P94842	A0A0H3M6V9, Q9C7I5, P94842	A0A0H3M6V9, Q9C7I5, P44679, P0A8Z5, P94842	B5B3P5, P94842, A0A0H3M6V9
TE10	4	Q9KBC9	P56653, Q5SJV0, Q9KBC9, O67466	P56653, Q5SJV0, Q9KBC9	P56653, Q5SJV0, Q9KBC9, O67466	P56653, Q9KBC9
TE11	16	P45083	P0A8Y8, Q7SGA6, Q9I3A4, P77781, Q9SX65, Q04416	P0A8Y8, NCU02744.1, Q9I3A4, P77781, Q9SX65, BAB40578.1	P0A8Y8, NCU02744.1, Q9SX65, BAB40578.1	Q04416, P77781, Q9SX65, B4XYA6
TE13	7	P76084	Q5SJP3, A0A0M3KL08, A0A0M3KL07, Q8DUV0	A0A0M3KL08, Q8DUV0, P76084	A0A0M3KL08, P76084	P76084, A0A0H2URF0, Q5SJP3
TE14	12	A0A174JUF1, Q41635, Q0J0M2, G3ESV0	Q9SJE2, A0A837P8G3, G3ESU9, Q9SQI3	Q9SJE2, Q9SQI3	A0A837P8G3, G3ESU9	Q9SQI3
TE15	4	Q8KNG2	Q84HI7, Q8GME8, A0A2D0TCG5, Q8KNG2	Q84HI7, Q8GME8, A0A2D0TCG5, Q8KNG2	Q84HI7, Q8GME8	Q8KNG2
TE16	79	A0A0S2E7W7, B3FWS8, Q0UI00, P0DUV3	P25464, A0A0B4ESU9, B3FWT6, P0C064, A0A384XH94, I3LCW1, Q71SP7, P91871, O31784, A0A0X1KH98, O31827, A5YV76, Q45563, Q08787, I1RF58, P0DUV3, A0A125R003, P0DUV4, Q5AUX1	A0A384XH94, I3LCW1, Q71SP7, A5YV76, Q45563	1JMK, S0DZM7, Q12053, P25464, B3FWT6, P0C064, A0A384XH94, I3LCW1, P91871, O31784, A0A0X1KH98, O31827, I1RF58, P0DUV3, A0A125R003, A0A142C799	P12276, Q45563, Q03149, A5YV76
TE17	5	Q03133	A5TZD1, Q03133, F1CLA7, Q9ZGI2, A4KCE5	A5TZD1, Q03133	A5TZD1, Q03133, F1CLA7, Q9ZGI2, A4KCE5	Q03133
TE18	22	Q5VUB9, P9WQD5	A0A061LQM0, Q7BUF9, Q9I1H3, O54157, P08635	Q7BUF9, O54157, P08635	Q7BUF9, Q9I1H3, O54157, P08635	Q9NV23, Q7BUF9
TE20	65	A0A0G2JKR3, Q8L7H5, P50897, Q9LVS4, A0A1P8B5G7, A0A286YFL8	A0A654G8S8, Q10T53, A5A8Z8, Q336S3, Q9LY31, Q9W3C7, P50897, A0A5S9XTF3, A0A287AYH9, Q9LVS4, P45478, A0A5G2QB02, A0A0G2JLK6, E9PVM9, E9PIA8, O59747	K7GLB5, A0A0G2JLK6, O59747	Q10T53, K7GLB5, A0A287AYH9, O59747	O59747, O35448, Q9UMR5
TE21	72	Q5QPN5, E5RJ48, C6VYE5, Q94E46	E5RGR0, C6VYE5, Q9HXE7, Q8L9X1, Q53547, Q5VWZ2, Q9VGV9, Q3UFF7, D3Z269		E5RGR0, C6VYE5, Q8L9X1, Q53547, Q5VWZ2	O75608, Q3J2V1
TE22	44	B8Y562, B7F3S0, H3BL99	P51025, A0A7U9J4Q3, Q07XK4, Q8LAS8, D0VWZ4, A0A8J0UAE1	P51025, A0A7U9J4Q3	P51025, A0A7U9J4Q3, A0A8I6A8R5	P51025, P33018, Q2FUY3
TE23	74	Q10LW8, A0A0H3JL43, Q0CCY4, A0A8I5KVK5, A7Z4X7, Q53H82	O24496, A0A2I1C3U0, Q8ZRM2, E1ACR1, A7Z4X7, Q53H82, A0A1L9WLF1, D7PHZ8, F1RU12, Q16775, C8WS08, Q5AXB0	B1XD76, Q8ZRM2, C8WS08	O24496, B1XD76, Q53H82, D7PHZ8, C8WS08	P0AC84, P0CU68
TE26	26	A0A5G2R2F9	G8JVR4, A0A1E5RUL9, A0A2S1GUX0	G8JVR4, A0A1E5RUL9	G8JVR4, A0A2S1GUX0	A0A1E3P8S6
TE27	8	B9JYM4	A0A0R4ILM1	A0A0R4ILM1	A0A0R4ILM1	Q9NUJ1
TE29	46	A0A494BBA3, A0A7G2EI69, P9WLC7, A0A0R4IW94	Q9SB70, Q8VYT1, A0A654G156, A0A7G2EI69	Q9SB70, Q8VYT1, A0A654G156	Q9SB70	Q6PCB6
TE30	8	Q9Y7C9	Q0CF71, Q3S2U0	Q3S2U0	Q0CF71, Q3S2U0	A0A161CKG1
TE31	6	A0A8I5ZSX8, H0Y6P4	A0A8I5ZSX8, H0Y6P4, Q3UUI3, Q5T1C6, A0A0G2JEK7	H0Y6P4, Q5T1C6, A0A0G2JEK7	A0A8I5ZSX8, A0A0G2JEK7	Q8N1Q8

Continued

Family	Ground set size	Greedy	Bidirectional Greedy	CD-HIT	MMseqs2	Ad hoc w/ manual curation
TE34	23	A3PGR7, Q8N0X4	P17725, Q9ZC38, F1RP25, Q5JVC1, S5N020		P17725, Q9ZC38, F1RP25, S5N020	Q8R4N0
TE35	33	H7C3P5	H7C3P5, A0A1L8GNZ8	A0A1L8GNZ8, P97819	Q95YD2, A0A1L8GNZ8	A0A3L7I2I8

Table 1. Representative sets for TE families produced by Greedy algorithm, Bidirectional Greedy algorithm, CD-HIT, MMseqs2, and the ad hoc with manual curation currently used in the ThYme database. The Ground set size column indicates the number of experimentally verified protein sequences within the corresponding family, which serve as the ground set on which the algorithms operate.

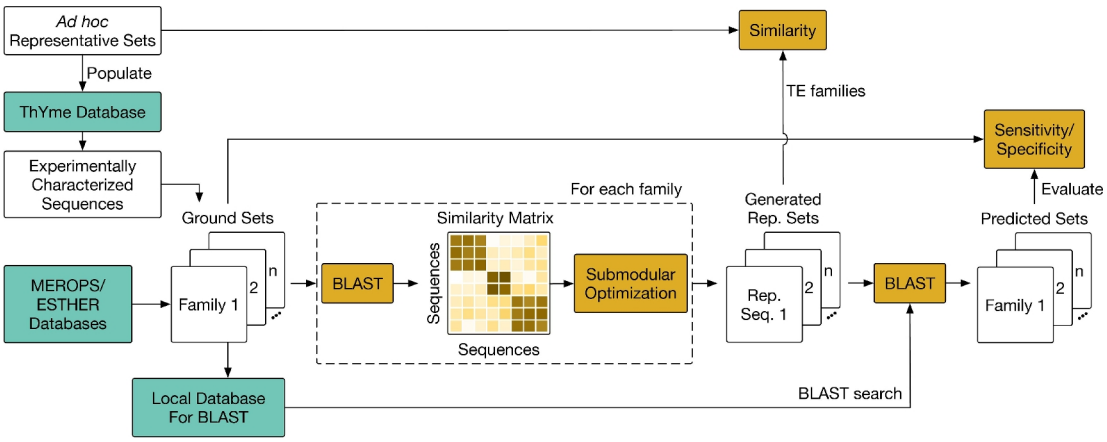

Fig. 4. For each protein family, we use BLAST to compute pairwise percentage identities within the ground set, generating an identity matrix that serves as input for the submodular function. For families from the ThYme database, we compare the representative sequences generated by submodular optimization algorithms with those from our ad hoc procedure as a benchmark. Our comparison focuses on correctly recognized protein IDs and residue identity. For all families, we evaluate the quality of the representative sets using sensitivity and/or specificity metrics. We perform a BLAST search against a local database of ground sets, using sensitivity to assess completeness (coverage) and specificity to measure accuracy.

and ESTHER. To facilitate reproducibility of our analysis results, we consolidated all ground sets for individual families into a single local database, which we provide as a .FASTA file (see Data availability).

Ad hoc procedure for identifying protein families

The TE enzyme families in the ThYme database are based from representative sequences that were chosen by an ad hoc procedure from a larger pool that includes all known TE sequences that have been experimentally verified[2]. The ad hoc method to identify enzyme families in ThYme prioritizes completeness to ensure the inclusion of all known sequences with a specific function, particularly those experimentally characterized or with known tertiary structures. We retrieved sequences with specific functions and experimental characterizations from the UniProt database[46,47]. We then used each retrieved sequence as a query in a Basic Local Alignment Search Tool (BLAST)[48,49] search against all known protein sequences (nr database). We compared the BLAST results against each other to identify the representative sequence of a family: which is the query sequence with the BLAST results that ensure completeness. We then populated the families by subjecting the catalytic domain of the representative sequence of a family through BLAST again and verifying sequence and structural similarity with multiple sequence alignments and tertiary structure superimposition. After identifying representative sequences using this ad hoc method, we performed expert manual curation to further refine the set and ensure the inclusion of the correct sequences if needed. While this procedure ensures completeness, it remains labor-intensive, particularly in verifying the quality of the representative set, which currently hampers the update and maintenance processes of the ThYme database.

Evaluation workflow

Figure 4 illustrates our overall analysis workflow for method evaluation. In this analysis, we will apply submodular optimization using the Greedy and Bidirectional Greedy algorithms and compare their performance to two widely-used clustering methods for identifying representative protein sequences: CD-HIT[23] and MMseqs2[25]. For each family, after establishing the ground set, we used BLAST alignment tool (blastp) to compute the percentage identity for every pair of sequences within this set. This step produces a pairwise percentage identity matrix that captures the similarity relationships among the sequences, serving as input for our submodular function. For

CD-HIT and MMseq2, we provided the required ground set .FASTA files as input and used the default settings for the evaluation analysis.

For TE families, we compared the representative sequences generated by these algorithms with those obtained through our ad hoc procedure, which includes manual curation as used in the current ThYme database. We focus our comparison on the correctly recognized protein sequences that are highly similar to these optimal sets of representative sequences. Additionally, we use two metrics, sensitivity and specificity, to assess the quality of the representative sets generated by the submodular optimization in terms of completeness (inclusion of all known sequences belonging to the family) and specificity (accurate family representation). To do this, we use the representative sequences generated by the submodular optimization algorithms as the query set for a BLAST search against a local database containing all experimentally characterized sequences from all families in the TE group. We define sensitivity for a specific family as the percentage of that family's sequences (ground set) correctly identified in the BLAST search results. Specificity is calculated as one minus the proportion of sequences from other families that are incorrectly identified as belonging to the family in question when using its representative sequences as the query set.

We do not know how the MEROPS and ESTHER databases populate the CE and A1A families. Unlike the TE families in ThYme which are based on blastp results of experimentally-verified sequences, we cannot compare our results for the CE and A1A against representative sequences with experimental validation. Instead, we assess the performance of the algorithms by comparing the coverage of the entire family, using sensitivity as the primary metric. For the CE and A1A families, sensitivity reflects the proportion of sequences in the family that are correctly identified by the representative sets returned by the different algorithms, providing a measure of how well the returned sets cover the entire family.

Comparing the optimization approaches with the ad hoc procedure

Table 1 presents a comparative analysis of the representative sequence selection results across five different methods: the Greedy algorithm, the Bidirectional Greedy algorithm, CD-HIT, MMseqs2, and the ad hoc procedure (which includes manual curation) used in the ThYme database. The "Ground Set Size" column indicates the number of experimentally verified protein sequences within each TE family, serving as the ground set on which the algorithms operate. We considered the sets returned from the ad hoc procedure as the benchmark in the analysis for TE families. In general, all algorithms produce representative sequence sets that include most of the sequences identified by the ad hoc procedure for the majority of TE families. However, there are notable differences in the size of the representative sets generated by each algorithm. The Greedy algorithm, which focuses on selecting a minimal yet diverse set of representatives, performs similarly to the ad hoc procedure in 12 out of the 27 TE families. It produces smaller sets in 6 families (e.g. TE1, TE6) and larger sets in 9 families (e.g. TE2, TE4). In terms of redundancy, measured by the size of the representative set, the Greedy algorithm performs similarly to our ad hoc procedure in 12 out of 27 TE families. It returns smaller sets in 6 families and larger sets in 9 families compared to the benchmark. Conversely, we observe that the Bidirectional Greedy algorithm generally returns larger sets than the ad hoc procedure, with exceptions in TE27 (similar to ad hoc) and TE6 (smaller than ad hoc). CD-HIT returns a larger set compared to the benchmark for 19 TE families, sets with equal size for 6 TE families and did not return any result for 2 families (TE21 and TE34). Similarly, MMseqs2 also returns less tighter representative sets for most of the families, with exeptions for TE3, TE27, TE29 (equal size) and TE13 (smaller set). We conclude that the Greedy algorithm tends to select a smaller number of representative sequences compared to the other algorithms. This approach effectively captures the essential diversity within each family while minimizing unnecessary duplications, resulting in a more concise and efficient representation.

We further compared the sequence identity of the sets returned by four algorithms-Greedy, Bidirectional Greedy, CD-HIT, and MMseqs2- with the current set in our database, which we created using an ad hoc procedure with careful manual curation. Figure 5 presents the distributions of percentage identity between the sequences in the sets returned by the two algorithms and those in the optimal sets from our ad hoc procedure. The results reveal that the Greedy and Bidirectional Greedy algorithms achieve sequence sets with high similarity to the ad hoc procedure in many families, with a similarity exceeding 80% in TE1, TE8, TE18, TE21, TE22, TE29, and TE35. However, in families with a larger number of sequences, such as TE4, TE16, TE26, TE27, and TE30, the Greedy algorithm exhibits lower similarity, with percentage identities below 50%. The Bidirectional Greedy algorithm also returns sets with lower similarity in families such as TE4, TE17, and TE30. For CD-HIT and MMseqs2, the performance is more variable across TE families. CD-HIT tends to generate sets with high similarity to the ad hoc sets in several families, including TE7, TE18, TE29, and TE31, but struggles with lower similarity (below 50%) in families like TE4, TE6, TE26, and TE30. Similarly, MMseqs2 performs well for TE17, TE18, TE22, and TE34 but shows lower similarity in TE4, TE26, and TE30.

Overall, the Greedy algorithm tends to select fewer representative sequences compared to the Bidirectional Greedy algorithm, CD-HIT, and MMseqs2. This approach effectively captures the essential diversity within each TE family while minimizing redundancy, resulting in a more concise and efficient representation of the sequence space. The Bidirectional Greedy algorithm, although generally returning larger sets, also struggles with certain families. Both CD-HIT and MMseqs2 show mixed results, indicating that further refinement is needed to improve their performance, particularly in complex families with many sequences.

Sensitivity and specificity analysis

We assess the accuracy of both algorithms by using their returned representative sequences to populate their corresponding TE families. Figure 6 provides insights into the sensitivity of the Greedy and Bidirectional Greedy algorithms in selecting representative sequences for TE families. We observe robust performance from both algorithms, which achieve a sensitivity of 100% across all TE families when considering sequences

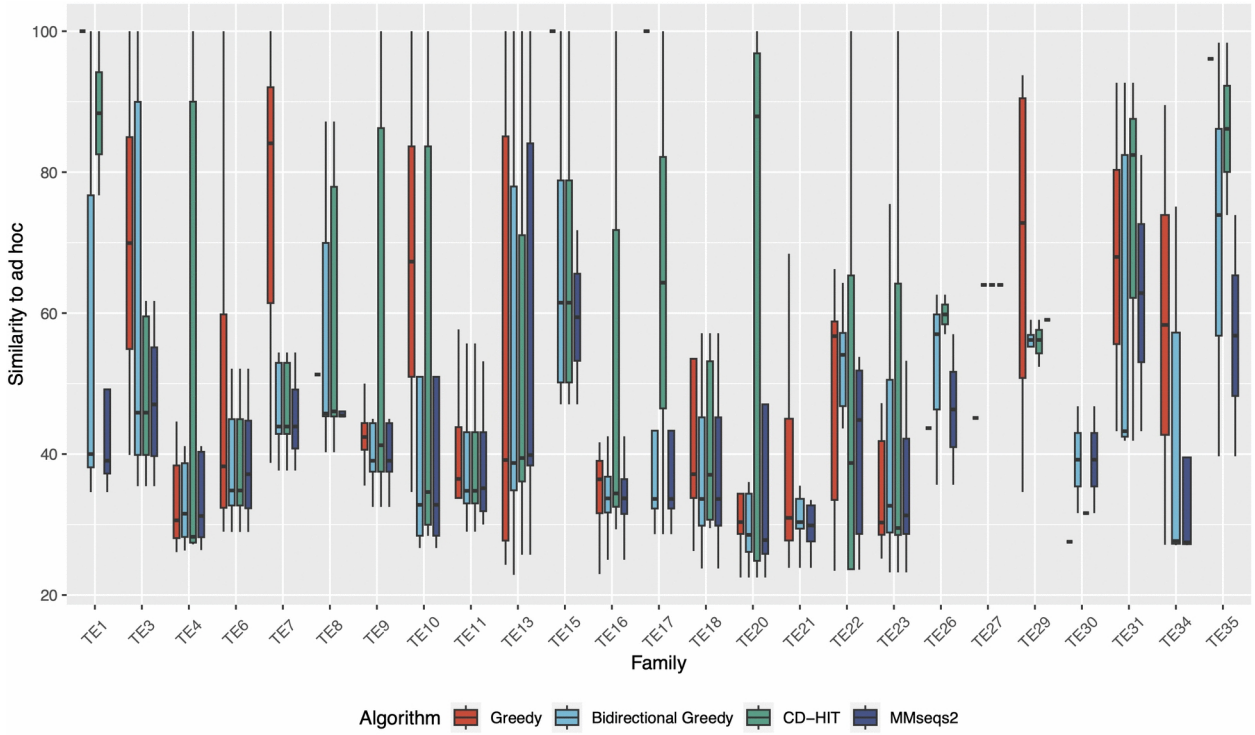

Fig. 5. Distributions of sequence identity between the benchmark representative sequences of each family and the sets returned by four optimization methods: two submodular optimization algorithms (Greedy and Bidirectional), CD-HIT, and MMseqs2. Each family is represented by four box plots showing the results for the Greedy algorithm, Bidirectional Greedy algorithm, CD-HIT, and MMseqs2, arranged from left (high transparency) to right (low transparency). A box plot with a single dot indicates that the algorithm returned a set containing only one sequence; otherwise, the returned set contains multiple sequences. Overall, the Greedy algorithm demonstrates the best performance, producing a tight set with high similarity to the ground truth. CD-HIT also performs well, though its representative sets generally contain more sequences than those from the Greedy algorithm. The Bidirectional Greedy algorithm and MMseqs2 show similar performance, often returning larger sets of representative sequences for each family.

with evidence at the protein level. The Greedy algorithm, on average, achieves an impressive 80% sensitivity with a representative set size of only 2 sequences. In contrast, the Bidirectional Greedy algorithm requires a larger representative set size of 4 sequences to reach a similar sensitivity level. This difference highlights the varying efficiency of the algorithms in capturing the diversity and coverage of sequences within each TE family. Furthermore, we find that both algorithms consistently produce representative sets that effectively retrieve all sequences from their respective TE families in BLAST searches, underscoring their utility in comprehensive sequence representation.

We assess the specificity of the representative sets generated by the Greedy and Bidirectional Greedy algorithms and show the results in Fig. 7. We observe a specificity of 1 across all TE families, indicating that both algorithms terminate without errors in sequence identification from other families. The maximum error rate, averaging around 0.02%, signifies the high precision and accuracy of these representative sets in distinguishing sequences belonging exclusively to their respective families. Although both algorithms perform exceptionally well, we find that the Greedy algorithm exhibits a slightly superior specificity compared to the Bidirectional Greedy algorithm, showcasing its ability to minimize cross-family sequence identifications more effectively.

We emphasize that the size of the representative set produced by these algorithms was not our primary focus in this application. We configure the Greedy algorithm to terminate based on a predefined threshold of sequence similarity, ensuring that the selected sequences adequately represent the diversity within each family. In contrast, we allow the Bidirectional Greedy algorithm to continue until it successfully identifies all sequences in the ground set via BLAST, potentially resulting in larger representative sets. Since the Bidirectional Greedy algorithm builds upon the Greedy algorithm, their sensitivity and specificity outcomes are generally comparable. This consistency in performance underscores the reliability of both algorithms in accurately identifying and representing sequences from their respective TE families.

Coverage analysis for carboxylesterase and peptidase families

Here we demonstrate that submodular functions can be applied to identify representative sequences of other protein families. Results show the potential of greedy algorithms in finding representative protein sequences of enzymes families with high sequence similarity; however, results are not complete without a thorough validation

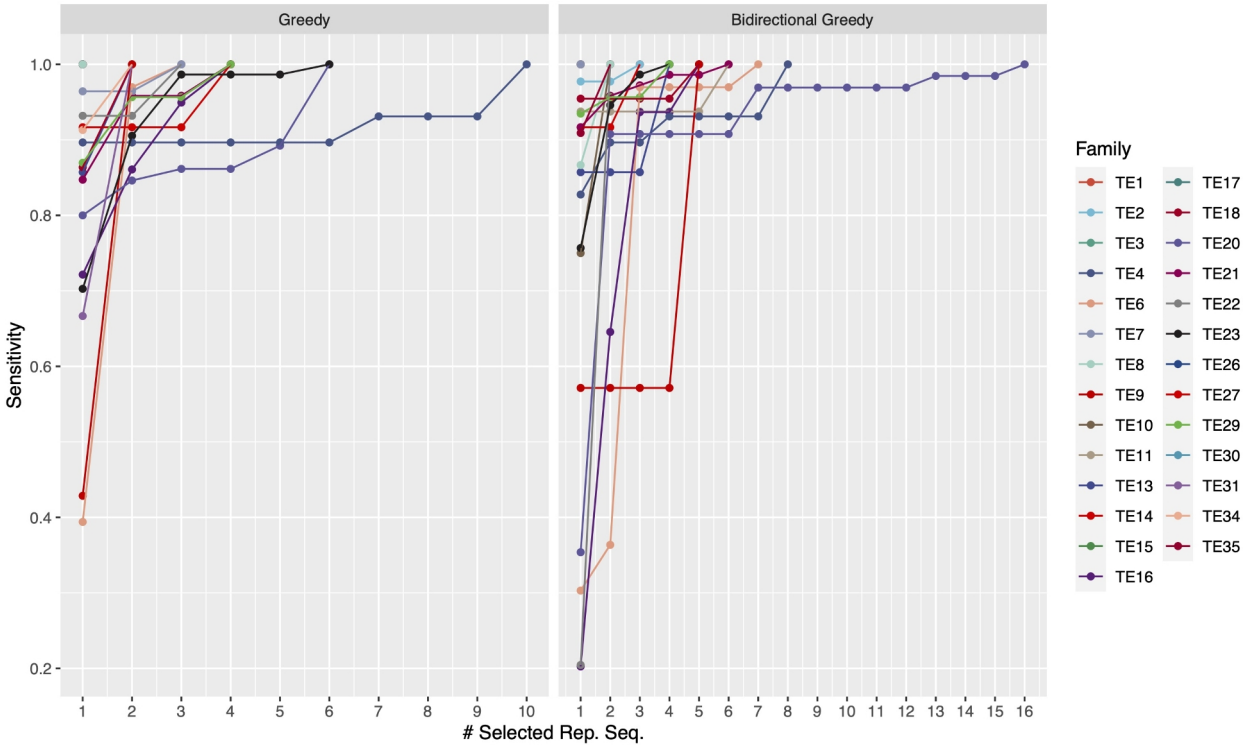

Fig. 6. Comparison of the sensitivity of two algorithms, Greedy and Bidirectional Greedy, in optimizing submodular functions for identifying representative protein sequences. The sensitivity values are plotted against the number of selected representative sequences across protein families in the TE group. Both algorithm demonstrates high sensitivity, but Greedy algorithm performs better in reducing the redundancy than Bidirectional Greedy algorithm does.

that greedy algorithms can identify representative sequences for any protein family. We applied the approach to two additional protein families, carboxylesterase (CE) enzyme family and the peptidase family A1 (A1A). We also compared the four methods (Greedy, Bidirectional Greedy, CD-HIT, MMseqs2) using the same data from CE and A1A families. For these two families, MMseqs2 returned many representative sequences. Specifically, MMseqs2 returned 2,323 representative sequences for the CE family and 1,701 for the A1A family. On the other hand, CD-HIT generated 10,564 representative sequences for the CE family and 8,774 for the A1A family. Note that CD-HIT and MMseqs2 are based on clustering, i.e., the methods group the sequences based on their similarity and then returns a representative sequence for each cluster[50]. There are two potential drawbacks of such approach: (1) it might return many clusters with many representative sequences, as shown above, and (2) it does not necessary provide high coverage because a cluster of sequences might not be covered 100% by a single representative sequence.

In contrast, the submodular optimization approaches offer more flexibility, allowing us to control the number of representative sequences returned Fig. 8 shows the coverage against on the number of representative sequences chosen by the two submodular algorithms. For CE family, both algorithms achieve 100% coverage with less than 10 representative sequences. For the A1A family, both algorithms achieve over 90% coverage with less than 10 representative sequences. The two algorithms achieve a 100% coverage with approximately 600 sequences, in which the Greedy algorithm has a slightly tighter set.

Discussion

In this study, we employed submodular optimization techniques to select representative sequences within thioesterase protein families in the ThYme database. This approach was motivated by the inherent redundancy in biological sequence datasets, which can obscure meaningful patterns and complicate genetic analysis. Our goal was to demonstrate the effectiveness of the facility location function, a widely used submodular function, in conjunction with optimization algorithms to enhance the selection of representative sequences. We specifically selected these algorithms because they minimize redundancy while ensuring high coverage of the protein family. Facility location functions have proven effective in modeling problems that involve selecting a representative subset from a larger set, particularly in clustering and summarization tasks. These functions balance the need to maximize coverage while minimizing redundancy, making them an ideal choice for selecting representative sequences from biological datasets. In contrast, information-theoretic functions like entropy or mutual information-though powerful-focus primarily on maximizing information gain. They are more suited to scenarios where capturing diverse information is essential, such as in cases involving uncertainty[51,52], rather than focusing on physical proximity or sequence similarity. Since our primary objective was to select a set of sequences

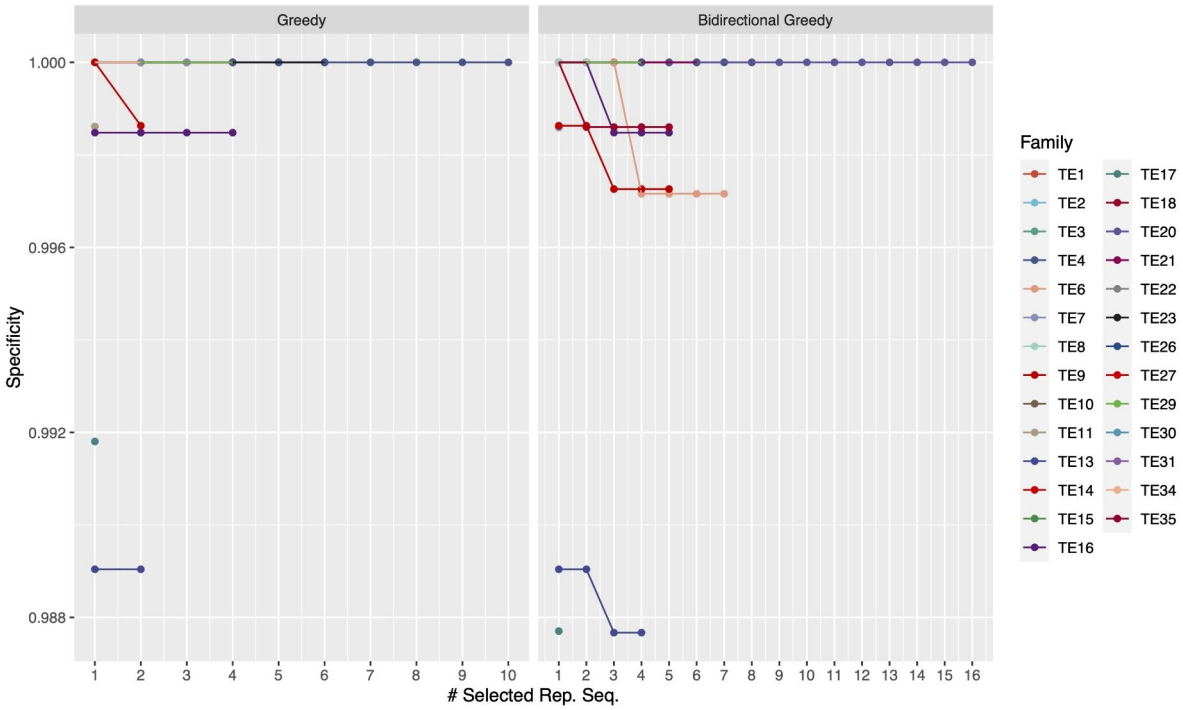

Fig. 7. Comparison of the specificity of two algorithms, Greedy and Bidirectional Greedy, in optimizing submodular functions for identifying representative protein sequences. The specificity values are plotted against the number of selected representative sequences across various protein families (TE1 to TE35). Both algorithms exhibit high specificity, with the maximum error rate is around 0.02%.

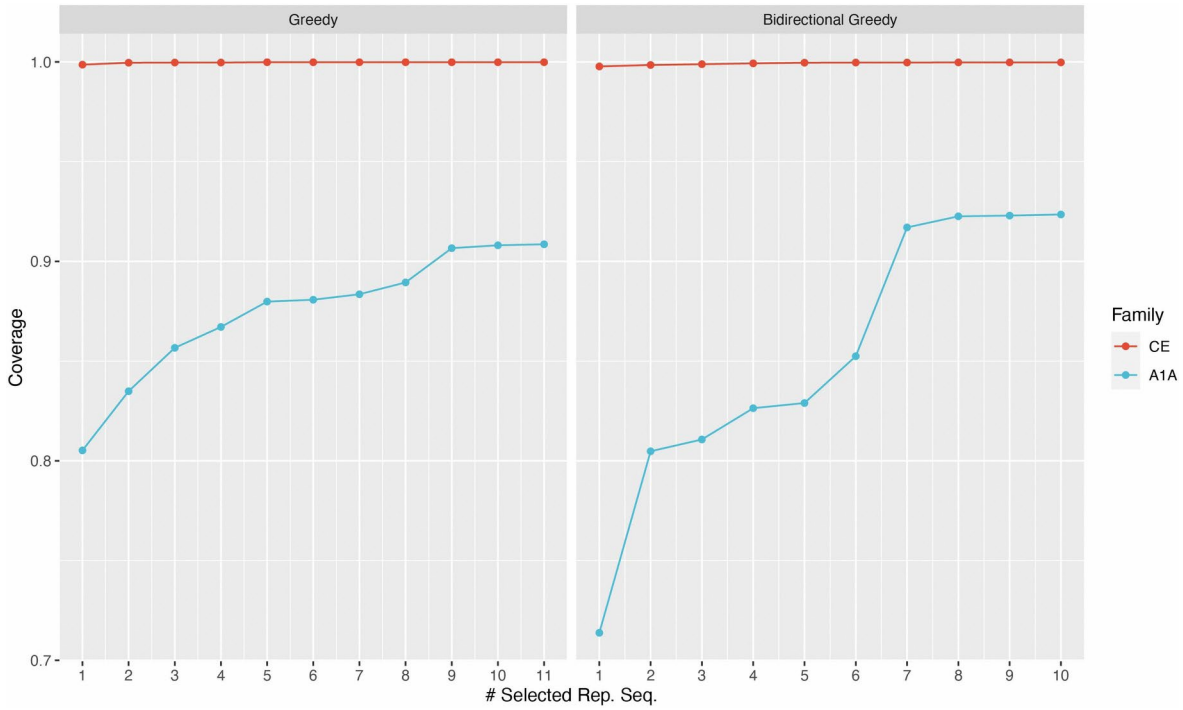

Fig. 8. Comparison of the coverage of two algorithms, Greedy and Bidirectional Greedy, in optimizing submodular functions for identifying representative protein sequences. The coverage values are plotted against the number of selected representative sequences across protein families in the two protein families: the carboxylesterase (CE) enzyme family and the peptidase family A1 (A1A). Both algorithm demonstrates to return the representative set with high coverage, but Greedy algorithm performs better in reducing the redundancy than Bidirectional Greedy algorithm does.

based on similarity and redundancy reduction, facility location functions were more directly applicable to our problem. However, information-theoretic approaches could be valuable for future extensions of this work.

Accordingly, we applied the approaches to real-world datasets to evaluate their performance. The results of our study highlight several key findings. Firstly, the submodular optimization approach consistently produced representative sets that effectively captured the entire spectrum of sequences within their respective families, achieving high sensitivity. This capability is crucial in ensuring that no significant sequence variations or genetic traits are overlooked during analysis. Furthermore, we meticulously evaluated the specificity of our approach. This involved assessing the extent to which sequences from unrelated families were mistakenly identified when using the representative sequences of a specific family as query sequences Our findings indicate that the representative sets generated through submodular optimization exhibited commendable specificity, minimizing the occurrence of false positive identifications and maintaining accurate family-level distinctions. In terms of optimization algorithm performance, the Greedy algorithm emerged as particularly effective in enhancing specificity and reducing redundancy compared to the Bidirectional Greedy algorithm. The Greedy algorithm systematically adds sequences based on their marginal contributions to the overall representation, thereby optimizing the balance between coverage and specificity, which explains its superior performance.

When applied to protein families with significant sequence similarity, our approach enables the selection of smaller, yet highly informative set of representative sequences compared to traditional clustering algorithms. It is important to note that representative subset selection differs fundamentally from clustering. While clustering results in groups of similar sequences, representative subset selection focuses on choosing individual sequences that best represent the entire dataset. Applying clustering methods to representative selection requires an additional step to identify an exemplar sequence for each cluster. In practice, clustering algorithms are rarely used for the purpose of selecting representative sequences from large datasets[20]. We note that when dealing with large datasets, unverified sequences are often present, and submodular optimization algorithms may frequently select these unverified sequences based solely on amino acid sequences, which can lead to inaccuracies. This means that these algorithms are not necessarily robust to errors occurring in generic-purpose databases, and are to be used for carefully curated datasets.

In summary, our study underscores the significant advantages of employing submodular optimization in the selection of representative sequences within protein families with significant sequence similarity in specialized, curated databases. By addressing redundancy and enhancing specificity, this methodology not only improves the accuracy of biological analyses but also contributes to a deeper understanding of genetic diversity and evolutionary relationships. These insights are pivotal for advancing research in various biological disciplines and informing strategies for precision medicine and biotechnological applications.

Data availability

The protein sequences data are available at ThYme database https://thyme.engr.unr.edu/v2.0/, MEROPS database https://www.ebi.ac.uk/merops/, and ESTHER database https://bioweb.supagro.inrae.fr/ ESTHER/. The code and data for reproducing the results are available at https://doi.org/10.5281/zenodo.14063070.

Received: 9 July 2024; Accepted: 1 January 2025
Published online: 07 January 2025

References

1. Yi, G., Thon, M. R. & Sze, S.-H. Supervised protein family classification and new family construction. *J. Comput. Biol.* **19**, 957–967 (2012).
2. Cantu, D. C., Chen, Y. & Reilly, P. J. Thioesterases: a new perspective based on their primary and tertiary structures. *Protein Sci.* **19**, 1281–1295 (2010).
3. Cantu, D. C., Chen, Y., Lemons, M. L. & Reilly, P. J. ThYme: a database for thioester-active enzymes. *Nucleic Acids Res.* **39**, D342–D346 (2010).
4. Caswell, B. T. et al. Thioesterase enzyme families: Functions, structures, and mechanisms. *Protein Sci.* **31**, 652–676 (2022).
5. de Carvalho, C. C., Murray, I. P., Nguyen, H., Nguyen, T. & Cantu, D. C. Acyltransferase families that act on thioesters: Sequences, structures, and mechanisms. *Proteins Struct. Funct. Bioinform.* **92**, 157–169 (2024).
6. Waterhouse, A. et al. SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic Acids Res.* **46**, W296–W303 (2018).
7. Aebersold, R. & Mann, M. Mass spectrometry-based proteomics. *Nature* **422**, 198–207 (2003).
8. Nguyen, Q.-H., Nguyen, H., Oh, E. C. & Nguyen, T. Current approaches and outstanding challenges of functional annotation of metabolites: a comprehensive review. *Briefings Bioinforma.* **25**, bbae498 (2024).
9. Khalil, A. S. & Collins, J. J. Synthetic biology: applications come of age. *Nat. Rev. Genet.* **11**, 367–379 (2010).
10. Hobohm, U., Scharf, M., Schneider, R. & Sander, C. Selection of representative protein data sets. *Protein Sci.* **1**, 409–417 (1992).
11. Holm, L. & Sander, C. Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics* **14**, 423–429 (1998).
12. Li, W., Jaroszewski, L. & Godzik, A. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics* **17**, 282–283 (2001).
13. Edgar, R. C. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* **26**, 2460–2461 (2010).
14. Parsons, J., Brenner, S. & Bishop, M. Clustering cDNA sequences. *Bioinformatics* **8**, 461–466 (1992).
15. Altschul, S. F. et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997).
16. Enright, A. J. & Ouzounis, C. A. GeneRAGE: a robust algorithm for sequence clustering and domain detection. *Bioinformatics* **16**, 451–457 (2000).
17. Rice, P., Longden, I. & Bleasby, A. EMBOSS: the European molecular biology open software suite. *Trends Genet.* **16**, 276–277 (2000).
18. Lin, G. et al. A multi-model approach to simultaneous segmentation and classification of heterogeneous populations of cell nuclei in 3D confocal microscope images. *Cytometry Part A J. Int. Soc. Anal. Cytol.* **71**, 724–736 (2007).

19. Lin, H. & Bilmes, J. A class of submodular functions for document summarization. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 510–520 (2011).
20. Libbrecht, M. W., Bilmes, J. A. & Noble, W. S. Choosing non-redundant representative subsets of protein sequence data sets using submodular optimization. *Proteins Struct. Funct. Bioinform.* **86**, 454–466 (2018).
21. Bateman, A. et al. Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins. *Nucleic Acids Res.* **27**, 260–262 (1999).
22. Burley, S. K. et al. Structural genomics: beyond the human genome project. *Nat. Genet.* **23**, 151–157 (1999).
23. Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–3152 (2012).
24. Wang, G. & Dunbrack, R. L. Jr. PISCES: a protein sequence culling server. *Bioinformatics* **19**, 1589–1591 (2003).
25. Hauser, M., Steinegger, M. & Söding, J. MMseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics* **32**, 1323–1330 (2016).
26. Steinegger, M. & Söding, J. Clustering huge protein sequence sets in linear time. *Nat. Commun.* **9**, 2542 (2018).
27. Lin, H. & Bilmes, J. Learning mixtures of submodular shells with application to document summarization. In *Proc. of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, 479–490 (AUAI Press, 2012).
28. Liu, Y., Wei, K., Kirchhoff, K., Song, Y. & Bilmes, J. Submodular feature selection for high-dimensional acoustic score spaces. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 7184–7188 (IEEE, 2013).
29. Wei, K., Liu, Y., Kirchhoff, K. & Bilmes, J. Using document summarization techniques for speech data subset selection. In *Proc. of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 721–726 (2013).
30. Wei, K., Liu, Y., Kirchhoff, K., Bartels, C. & Bilmes, J. Submodular subset selection for large-scale speech training data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3311–3315 (IEEE, 2014).
31. Kirchhoff, K. & Bilmes, J. Submodularity for data selection in machine translation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 131–141 (2014).
32. Tschiatschek, S., Iyer, R. K., Wei, H. & Bilmes, J. A. Learning mixtures of submodular functions for image collection summarization. *Adv. Neural Inf. Process. Syst.* **27** (2014).
33. Krause, A. & Golovin, D. Submodular function maximization. *Tractability* **3**, 3 (2014).
34. Birkhoff, G. On the combination of subalgebras. *Math. Proc. Cambridge Philos. Soc.* **29**, 441–464 (1933).
35. Fujishige, S. Polymatroidal dependence structure of a set of random variables. *Inf. Control* **39**, 55–72 (1978).
36. Krause, A. & Guestrin, C. Near-optimal nonmyopic value of information in graphical models. In *Proc. of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, 324–331 (2005).
37. Schrijver, A. *Combinatorial Optimization: Polyhedra and Efficiency* Vol. 24 (Springer, 2003).
38. Frieze, A. M. A cost function property for plant location problems. *Math. Program.* **7**, 245–248 (1974).
39. Fujishige, S. *Submodular Functions and Optimization* (Elsevier, 2005).
40. Schreiber, J., Bilmes, J. & Noble, W. S. apricot: Submodular selection for data summarization in Python. *J. Mach. Learn. Res.* **21**, 1–6 (2020).
41. Nemhauser, G. L., Wolsey, L. A. & Fisher, M. L. An analysis of approximations for maximizing submodular set functions–I. *Math. Program.* **14**, 265–294 (1978).
42. Buchbinder, N., Feldman, M., Seffi, J. & Schwartz, R. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM J. Comput.* **44**, 1384–1402 (2015).
43. Mistry, J. et al. Pfam: The protein families database in 2021. *Nucleic Acids Res.* **49**, D412–D419 (2021).
44. Lenfant, N. et al. ESTHER, the database of the α/β-hydrolase fold superfamily of proteins: tools to explore diversity of functions. *Nucleic Acids Res.* **41**, D423–D429 (2012).
45. Rawlings, N. D. et al. The MEROPS database of proteolytic enzymes, their substrates and inhibitors in 2017 and a comparison with peptidases in the PANTHER database. *Nucleic Acids Res.* **46**, D624–D632 (2018).
46. Bairoch, A. et al. The Universal Protein Resource (UniProt). *Nucleic Acids Res.* **33**, D154–D159 (2005).
47. Bairoch, A. et al. The Universal Protein Resource (UniProt) 2009. *Nucleic Acids Res.* **37**, D169-174 (2009).
48. Altschul, S. F., Gish, W., Miller, W., Myers, W. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990).
49. Altschul, S. et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997).
50. Tran, D. et al. Fast and precise single-cell data analysis using hierarchical autoencoder. *Nat. Commun.* **12**, 1029 (2021).
51. Nguyen, H., Tran, D., Tran, B., Pehlivan, B. & Nguyen, T. A comprehensive survey of regulatory network inference methods using single-cell RNA sequencing data. *Briefings Bioinforma.* **22**, bbaa190 (2021).
52. Nguyen, H., Nguyen, H., Tran, D., Draghici, S. & Nguyen, T. Fourteen years of cellular deconvolution: methodology, applications, technical evaluation and outstanding challenges. *Nucleic Acids Res.* **52**, 4761–4783 (2024).

Acknowledgements

This work was partially supported by NSF (2001385, 2343019, and 2203236), NCI (U01CA274573), and NIGMS (R44GM152152). Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the funding agencies.

Author contributions

Ha N. and T.N. conceived of and designed the approach. Ha N. performed the data analysis and all computational experiments. Hung N. and D.C. helped with data preparation and some data analysis. P.N. and A.L. helped with data curation and exploratory analysis. Ha N., Hung N., D.C., and T.N. wrote the manuscript. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to T.N.