Journal Title Here, 2022, pp. 1–10

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

From Atoms to Fragments: A Coarse Representation for Functional and Efficient Protein Design

Leonardo V. Castorina,[1,] Christopher W. Wood[2] and Kartic Subr[1*]

[1]School of Informatics, The University of Edinburgh, 10 Crichton Street, Newington, Edinburgh, EH8 9AB, UK and [2]School of Biological Sciences, The University of Edinburgh, Roger Land Building, Edinburgh , EH9 3FF, UK
[*]Corresponding author. email-id.com
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Deep learning has made remarkable progress in protein design, yet current protein representations remain largely black-box and scale poorly with protein length, leading to high computational costs. We propose a fragment-based protein representation that balances interpretability and efficiency. Using a curated set of 40 evolutionarily conserved fragments, we represent proteins as fragment sets or fragment graphs, significantly reducing dimensionality while preserving functional information. Here, we show that fragment-based representations capture significantly more information at much lower dimensions compared to traditional methods. On a dataset of 215 functionally diverse proteins, our approach outperforms traditional sequence- and structure-based methods in clustering by protein function at $\leq$ 30% sequence identity. Additionally, fragment-based search achieves comparable accuracy while using 90% fewer tokens. It also runs $\sim$68.7$\times$ faster than RMSD-based methods and $\sim$1.64$\times$ faster than sequence-based methods, even when including fragment pre-processing overhead. Finally, we show that fragments can guide RFDiffusion backbone generation, with recovery rates higher than 40%. We propose fragment-based representations as a scalable and interpretable alternative for the next generation of protein design tools, spanning backbone and sequence design to functional searches in protein structure databases.

Key words: keyword1, keyword2, keyword3, keyword4

Introduction

Designing functional proteins could transform medicine, biotechnology, and sustainability. From enzymes that catalyze reactions, to vaccines against target diseases, proteins serve as precise molecular tools to our most pressing problems. However, designing proteins remains a computationally intractable problem due to the combinatorial complexity of the search space. With 20 possible amino acids at each position, the search space grows exponentially with protein length, making exhaustive explorations impossible.

To navigate this search space, Artificial Intelligence (AI) methods have enabled *de novo* design of protein binders [14], neutralizing antibodies against diseases [28], and enzymes [33, 23]. These models rely on different protein representations. Large Language Models (LLMs) (e.g., ProtGPT [11], ESM [26]), treat proteins as sequences, while diffusion models (e.g., RFDiffusion[31], EvoDiff[1]) represent protein structures as vector frames encoding the atomic coordinates. Other approaches represent structures as voxel grids (e.g., TIMED [8], DenseCPD[24]) or graphs (e.g., ProteinMPNN[10]).

While structure- and sequence-based representations have enabled breakthroughs, they impose significant computational burdens that scale non-linearly with protein size. This makes large-scale protein design prohibitively expensive and leads to increasingly complex models, highlighting the need for more efficient and interpretable representations.

To address these challenges, we propose **fragment-based representations** – an approach that represents proteins as combinations of evolutionarily conserved structural fragments instead of full sequences or atomic structures (see Figure 1). This idea is rooted in protein evolution, where structures and functions evolved from recombination, repetition, and accretion of small, functional peptides [2]. We show that this intermediate abstraction level significantly reduces dimensionality while preserving protein functional signatures, enabling faster and more interpretable protein search, analysis, and design.

Proteins inherently lend themselves to abstraction at multiple scales. Secondary structures such as α-helices and β-sheets provide simplified views of local folding patterns in ribbon diagrams [25]. At a higher level, tertiary structural motifs, such as β hairpins or helix-turn-helix domains are strongly associated with molecular functions and are widely used for design and analysis [21]. Our fragment-based representation follows this principle, focusing on functional building blocks rather than full atomic resolution.

© The Author 2022. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Previous studies have identified and leveraged recurring structural motifs for protein analysis and design. Alva et al. [3] identified a conserved set of ancient fragments associated with functions such as DNA and metal ion binding. Frappier et al. [13] introduced recurring Tertiary Structural Motifs (TERMs) to design protein binders. Kolodny et al. [20] compiled several sequence-based "THEMES" for functional protein analysis. Fragment-based representations have also been used beyond proteins, such as fragSMILES [22] for chemical representations and component-based approaches in computer vision [6].

Building on these ideas, we introduce fragment-based representations as an alternative to traditional sequence- and structure-based representations. Our method explicitly encodes the backbone geometry rather than amino acid sequence, decomposing proteins into functional fragments. Each fragment represents a recurring structural motif associated with specific functions. Using a library of 40 conserved fragments, we show that fragment-based representations effectively capture functional information at a much lower dimensionality than traditional methods.

Additionally, we provide a fully vectorized Python package (MIT License) for fragment detection and representation, adaptable to any fragment library.

We demonstrate three key applications of our fragment-based approach: (1) functional clustering to evaluate how well fragments capture protein function; (2) database searching to demonstrate effectiveness in retrieving functional proteins and computational efficiency; and (3) protein design using fragments as blueprints to guide RFdiffusion to generate backbones with functional signatures.

Because fragments encode functional units, fragment-based representations offer a computationally efficient and interpretable approach to protein representation, search, and design. By balancing efficiency and biological relevance, fragment-based representations provide a scalable foundation for the next generation of protein design tools.

Methods

We propose to represent proteins abstractly as being composed of evolutionarily conserved fragments. We choose 40 fragments identified by Alva et al. [3], representing ancient structural motifs associated with proteins that bind with DNA, RNA, metal ions, GTP, and ATP.

We first explain how to construct fragment-based representations protein structures and then evaluate them via three applications:

- functional clustering to assess how well fragments capture protein function;
- database searching to demonstrate effectiveness in retrieval and computational efficiency; and
- protein design to show how fragments can condition the backbone generation process.

In each case, we compare fragment-based representations against traditional sequence- and structure-based approaches.

Fragments as a Coarse Representation of Proteins

Given a protein structure from a Protein Data Bank (PDB) file, our representation decomposes the structure using evolutionarily conserved fragment motifs. We then propose two representations for the protein structure, without sequence information, as a **Fragment Graph** or as a **Fragment Set**. In the former, nodes in the graph represent fragments (with identification) and edges denote either peptide bonds between fragments or spatial proximity. Fragment Sets, on the other hand, only contain lists of unique fragments present in a structure, regardless of their arrangement (See Figure 1).

Our fragment library is based on the 40 fragments from Alva et al. [3]. To create a curated reference set, we extracted all instances of these fragments from their reported PDB structures using the AMPAL framework [32]. We then filtered these instances to ensure sequence consistency and correct residue lengths, resulting in a verified reference set of 219 instances across the 40 fragment types (see Supplementary Table 1).

Building Fragment Representations

Representing proteins as fragments involves three main steps: (1) detecting the fragments in the given structure, (2) classifying unmatched regions, and (3) converting the classified structure into a graph or a set representation (See Figure 1).

Fragment detection identifies segments of the target protein that match fragments in our library below a distance threshold. We implemented a sliding window algorithm (see Supplementary Algorithm 1) that computes distances between segments of the target protein and each fragment in the library. For this distance calculation, we evaluated several distance metrics both individually and in combination:

- **Sequence-based metrics** (sequence identity, BLOSUM distance) measure distance in amino acid sequence [17].
- **Angle-based metrics** (RMS, RamRMSD, LogPr) are sequence-independent and measure distance in backbone torsion angles (ϕ and ψ) [19].

This produces a **Fragment Distance Matrix** D quantifying the distance between each library fragment F_k and the segments of the target structure T. To classify regions as fragments, we normalize the Fragment Distance Matrix D to the $[0, 1]$ range. Regions with distances below the optimal threshold of 3.65% (determined through ROC analysis to maximize accuracy) are classified as matching fragments. If fragment matches overlap, we prioritize matches with lower distances. We allow up to two amino acids overlap between neighboring fragments.

After fragment detection, regions not matching any known fragments are classified according to their length: regions of 9–24 amino acids are classified as **unknown fragments**, while regions shorter than 9 amino acids are classified as **unknown connectors**.

Finally, we represent classified regions using two types of fragment-based representations:

- **Fragment Sets** record only presence or absence of fragment classes without considering connectivity information.
- **Fragment Graphs** represent structures as graphs, where nodes correspond to fragments (known or unknown) and connectors, and edges indicate peptide bonds or spatial proximity (<10 Å). Edge features are one-hot encoded to distinguish between connection types.

Fragment Sets are suitable for applications where speed is required, such as database searches. Fragment Graphs are better suited when the structural context is important, for example in protein design or functional clustering.

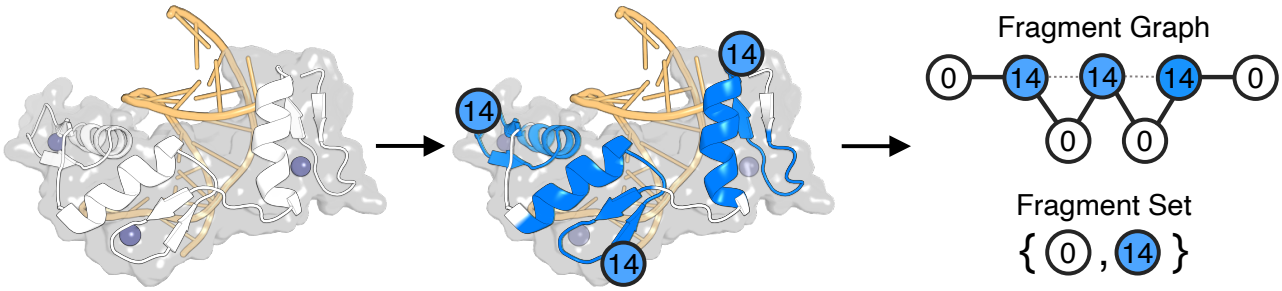

Fig. 1. Fragment-based protein representation. Conversion from protein structure to fragment representations using a ZIF268 Zinc Finger (PDB: 1AAY - DNA in Yellow and Zinc ions in Purple). The detection algorithm identifies regions matching known fragments, with Fragment 14 (blue), corresponding to DNA- and metal- binding functions essential for zinc fingers. Unclassified regions are labeled as "unknown" (white). The identified fragments are represented either as a Fragment Set, which counts unique fragment types or as a Fragment Graph, which preserves connectivity through peptide bonds (dark edges) and spatial proximity (dotted edges).

Next, we used the PDBench, a fold-balanced protein dataset, to analyze whether structural and chemical properties are preserved in fragment versus non-fragment regions [7].

Datasets for Validation

We used two datasets to validate our representation: PDBench to test the preservation of structural and chemical properties and Protein Function Dataset (PFD) to assess whether fragments capture functional properties in proteins.

To validate whether fragments preserve structural and chemical properties, we used PDBench [7], a fold-balanced dataset of protein structures. We tested whether properties such as hydrogen bonding and solvent accessibility are preserved in similar proportions between fragment and non-fragment regions. We hypothesized that important structural and chemical properties would be enriched in fragment regions relative to non-fragment regions. Specifically, we quantified the correlation between the percentage of each property in fragment regions and the fraction of the protein covered by fragments. A deviation from perfect correlation would indicate over- or under-represented properties.

To validate whether fragments capture functional relationships, we created PFD, a structurally diverse dataset of functional proteins. The dataset includes 215 protein monomers spanning 12 functional categories, covering binding of DNA, RNA, metal ions, GTP, ATP, and combinations thereof. We ensured structural diversity by filtering using Gene Ontology (GO) codes [4] and enforcing a sequence identity cutoff of $\leq$ 30% through the PDB Advanced Search interface [5]. This structural diversity is essential to assess whether functional relationships are captured, independent of high homology. Where possible, we selected 10 representative structures per functional category (detailed in Supplementary Table 1). We use this dataset to evaluate fragment-based functional clustering, then, to assess database search performance, and finally, for fragment-guided backbone generation.

Fragments for Functional Clustering

We tested the quality of our fragment-based representations by evaluating how well they capture functional relationships between proteins. Since fragments represent evolutionarily conserved functional motifs, we hypothesized that fragment-based representations should accurately capture functional similarities between proteins. Specifically, we evaluated whether proteins with similar functions cluster better when represented using fragments compared to traditional structure- and sequence-based representations.

To test this, we first computed pairwise distances for all protein pairs in the PFD using fragment-based, sequence-based, and structure-based metrics. Then, we projected each protein into a distance-preserving latent space of increasing dimensionality using Principal Coordinate Analysis (PCoA) and t-SNE. This allowed us to evaluate how effectively each representation captures functional relationships at various dimensions.

We selected these metrics to calculate the distances for clustering:

1. **RMSD (Root Mean Square Deviation)**: Measures traditional structural similarity using BioPython's CE-Aligner [29, 9].
2. **BLOSUM62**: Measures traditional sequence similarity using pairwise alignment scores based on amino acid substitution frequencies from BLOSUM[17].
3. **BagOfNodes**: Measures similarity based purely on the presence or absence of fragments (Fragment Sets), ignoring their spatial arrangement. This representation tests topology-independent functional information.
4. **GraphEditDistance**: Measures functional similarity by accounting for both fragment identity and spatial arrangement (Fragment Graphs), providing a more comprehensive fragment-based metric [12].

Next, we clustered the resulting embeddings using Gaussian Mixture Models (GMM) and K-Means. We set the number of clusters to 12 clusters, corresponding to the known functional categories within the Protein Function Dataset (PFD). To measure robustness and comprehensively evaluate clustering quality, we use Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Silhouette and Trustworthiness scores, F1-score, and the correlation between embedding distances and the original pairwise distances.

Fragments for Functional-based Searches

While the previous experiment evaluated the quality of the representation, here we assessed its feasibility for protein database searches. We tested how quickly fragment-based distances retrieved results and whether the retrieved proteins had similar functions as the query protein.

First, we benchmarked the initialization time, search speed, and memory requirements. Then, we tested the most relevant results matched the function of the protein query.

For benchmarking, we measured initialization and query times for our fragment-based methods (GraphEditDistance and BagOfNodes) against traditional approaches (RMSD and BLOSUM). We used 1, 10, and 100 queries on a database of 100 proteins to measure the scalability of the search, using 35 cores [1]. We also measured the memory requirement as the average number of data points required for each representation.

To assess the quality of the retrieved results, we queried each functional protein in the PFD against all other proteins and sorted by the lowest distance. We then evaluated whether the retrieved proteins shared functional similarity with the query using two complementary metrics: Normalized Discounted Cumulative Gain (NDCG) and Area Under the Receiver Operating Characteristic (AUROC), which measure whether functionally similar proteins rank higher in the results (see details in Supplementary Section 13).

From Fragments to Functional Proteins

Finally, we explored whether fragments could be used as blueprints to guide the generation of functional proteins by providing structural constraints to a generative model. We hypothesized that if fragments capture functional information, then proteins generated using fragment-derived templates should be structurally similar to known proteins with the same function.

To test this hypothesis, we used RFDiffusion[31], a state-of-the-art protein backbone generation model. For all the 215 PFD proteins, we generated partial backbone templates by masking non-fragment regions. After filling the missing regions, we used them as queries to assess whether the most similar results matched the function of the queries.

For each protein in the dataset, we first applied our fragment detection algorithm to identify functionally important regions. Then, we created partial backbone templates with only these regions, and non-fragment regions removed. RFDiffusion then filled in the missing regions, generating five candidate backbone structures per template.

To evaluate the functional recovery of the designs, we used FoldSeek to align each generated structure to known proteins in the PDB using sequence-independent shape matching [30]. We calculated the percentage recovery rate as the fraction of generated designs whose top 10 structural matches shared the exact Gene Ontology (GO) code(s) of the original protein. As a control, we performed the same evaluation with the original protein backbones to create a baseline for comparison.

Results

We evaluate our fragment-based representation across four key areas: fragment detection accuracy, physicochemical properties of fragments, effectiveness in capturing functional patterns, and applications in protein search and design.

Accurate Fragment Detection Using Combined Metrics

We validated the fragment detection algorithm using distance metrics based on sequence (BLOSUM and Sequence Identity) and angle-based distance metrics (LogPr, RamRMSD, and RMSD). We also tried RMSD (via both PyMol and BioPython), however it was much slower and we observed several silent failures during alignments. Supplementary Figure 1 shows the performance (F1 score) of individual metrics and ensembles. While individual metrics achieved modest F1 scores around 0.40, combining two complementary metrics significantly improved performance to approximately 0.85. The LogPr and RamRMSD combination consistently demonstrated the highest accuracy. Adding a third metric provided no significant improvement.

Using Receiver Operating Characteristic (ROC) analysis, we identified an optimal probability threshold of 3.65% or fragment classification, achieving an Area Under ROC (AUROC) of 87% (see Supplementary Figure 5).

Fragment Regions Show Distinct Structural and Chemical Properties

We evaluated physico-chemical properties of fragment and non-fragment regions using the PDBench benchmark. As shown in Supplementary Figure 6, the percentage of the protein covered by fragments was roughly 40% with consistent standard deviations. There were some outliers like Alpha Solenoid or Alpha-Beta Horseshoe at around 20%. The special folds had the highest standard deviation, larger than the coverage value itself. The fragment coverage was consistent across resolutions (Supplementary Figure 7).

Fragment regions showed a higher proportion of intra-fragment hydrogen bonds, especially in mainly β folds where we observed a $\sim$15% increase compared to non fragment regions. Conversely they showed lower inter-fragment hydrogen bonds, particularly in the mainly α folds with a $\sim$ 47% reduction. Surface accessibility was slightly reduced in fragment regions, showing a $\sim$5% decrease across most folds except special folds. Despite these structural differences, fragment and non-fragment regions maintained similar distributions of charge, polarity, and secondary structure elements (see Supplementary Section 10).

Fragment-Based Embeddings Efficiently Capture Functional Similarities

We evaluated how well fragment-based representations and traditional sequence- and shape-based methods capture functional similarities in reduced-dimensional embeddings. We compute a distance matrix for the dataset of 215 functional proteins using RMSD (shape), BLOSUM (sequence), BagOfNodes (fragment sets), and GraphEditDistance (fragment graphs).

We projected the data into lower dimensions using Principal Coordinate Analysis (PCoA) and calculated the cumulative explained variance across dimensions (Figure 2). Fragment-based representations significantly outperform traditional metrics, with BagOfNodes and GraphEditDistance preserving over 95% and 80% of cumulative variance within 20 dimensions, respectively. In contrast, traditional methods preserved significantly less information, with BLOSUM capturing less than 60% and RMSD less than 40% of the variance.

Interstingly, fragment-based distances showed strong correlation with sequence-based distances despite not directly using sequence information. GraphEditDistance achieved a Spearman correlation of 0.91 with BLOSUM distances, while BagOfNodes showed a moderate correlation of 0.57. RMSD,

[1] Intel(R) Core(TM) i9-10980XE CPU @ 3.00GHz

Distance	Based On	ARI	NMI	Silhouette	Trustworthiness	Distance Corr.	F1 Score
RMSD	Shape	*0.0357*	**0.3863**	-0.0334	0.9598	0.8647	*0.1957*
BLOSUM	Sequence	0.0027	0.2933	0.0248	*0.9923*	*0.9966*	0.1640
GraphEditDistance (ours)	Fragment Graph	**0.0458**	*0.3832*	*0.0766*	0.9915	0.9829	**0.1985**
BagOfNodes (ours)	Fragment Set	0.0050	0.3455	**0.8227**	**0.9991**	**0.9998**	0.1660

Table 1. Clustering performance comparison of different distance metrics using Gaussian Mixture Models (GMM) on Principal Coordinate Analysis (PCoA) embeddings of the functional protein dataset (215 proteins across 12 functional categories).

however, showed minimal or slightly negative correlation with other metrics.

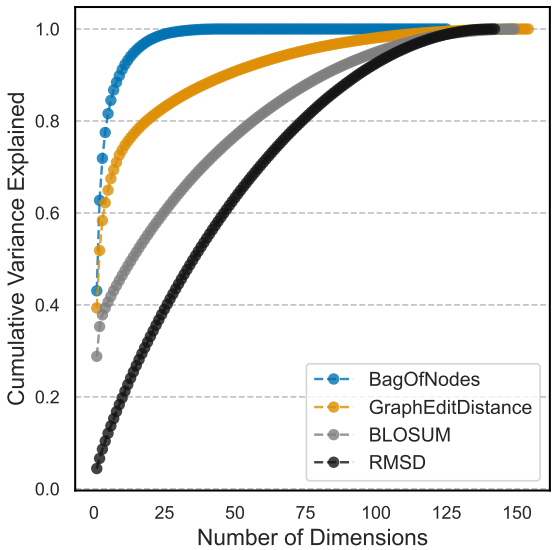

Fig. 2. Cumulative variance explained by different distance metrics after Principal Coordinate Analysis (PCoA) projection of a functional protein dataset containing 215 proteins across 12 functional categories.

We evaluated the clustering performance of Gaussian Mixture Models (GMMs) and K-Means using PCoA, t-SNE, and UMAP embeddings. Table 1 summarizes the results for GMMs on PCoA embeddings. Overall, fragment-based representations demonstrated better clustering performance across most metrics.

Notably, BagOfNodes achieved the highest Silhouette score (0.8227), indicating well-separated clusters, along with the best Silhouette and Trustworthiness scores, and Distance Correlation. GraphEditDistance performed best for for ARI (0.0458), indicating the highest agreement with the true functional clusters after adjusting for chance, and also the highest F1 score (0.1985). RMSD ranked highest in NMI score (0.3863), reflecting better mutual information between cluster assignments and true functions, and was second best for ARI and F1 Score. BLOSUM ranked second in Silhouette scores and Trustworthiness scores (see Supplementary Table 3).

Fragment-Based Search Combines Speed and Accuracy

We tested the fragment representation for functional proteins searches. We assessed both the quality of search retrieval and the computational efficiency of fragment distance methods (GraphEditDistance and BagOfNodes) against traditional sequence (BLOSUM) and shape (RMSD) distance methods.

Using the dataset of 215 functionally annotated proteins, we select individual proteins for each function as queries. Then, we calculate the pairwise distance to rank all other proteins. We assessed the quality of the retrieval using Normalized Discounted Cumulative Gain (NDCG) and Area Under the Receiver Operating Characteristic Curve (AUROC), to quantify how well the ranking of retrieved proteins matches the expected order based on shared functions.

As shown in Supplementary Figures 3 and 4, the AUROC and NDCG scores across all methods are generally within 1 standard deviations of one another. In terms of retrieval accuracy, fragment-based methods matched approaches for most functional categories, with particularly strong AUROC performance in identifying DNA+ATP+GTP-binding proteins (values >0.8 against 0.75 0.56 for RMSD and BLOSUM). RMSD showed an advantage in the NDCG for specific functions, especially in DNA+GTP, RNA+GTP, and RNA+GTP+Metal binding searches.

Then, we benchmarked the computational efficiency of each method, measuring query times for 1, 10, and 100 queries against a database of 100 proteins using 35 cores (Table 2).

Fragment-based representations substantially reduce data dimensionality compared to traditional methods. Relative to backbone atom representation (RMSD), our fragment approach achieves dimensionality reduction of 99.1% for fragment graphs and 99.7% for fragment sets. Even compared to sequence representations, we observe significant compression: 94.4% reduction for fragment graphs and 98.3% for fragment nodes.

Fragment-based representations use significantly less number of datapoints compared to traditional methods. Relative to backbone atom-based representations (RMSD), fragment graphs and fragment sets reduce dimensionality by approximately 99.1% and 99.7%, respectively. Compared to sequence-based representations, the reductions are 94.4% and 98.3%, respectively.

Overall, sequence search with BLOSUM distance is the fastest method considering initialization time and search time. BagOfNodes is the fastest search method overall, completing 100 queries in under 0.07 s, while other methods required substantially longer – RMSD took about 1717 s, GraphEditDistance about 573 s, and BLOSUM took 36.57 s.

Although fragment-based methods have a higher initial cost which involves converting protein structures to fragment graphs (around 6 s compared to 25 s), this cost is quickly offset by the faster search times.

Functional Design Recovery with Fragment-Based Diffusion

We evaluated the ability of fragment-based templates to guide the generation of functional proteins using RFDiffusion. For each of the 215 proteins, we generated a template backbone

Method	Init. Time (s)	1 Query Time (s)	10 Queries (s)	100 Queries (s)	Memory Req.
RMSD	*5.6796*	22.2474	155.3840	1717.0282	1744.36 ± 1306.90
BLOSUM	**4.5215**	*0.4238*	*3.4366*	*36.5742*	290.73 ± 217.82
GraphEditDistance (ours)	24.9708	5.6529	57.1725	573.0536	*16.39 ± 13.57*
BagOfNodes (ours)	24.9931	**0.0012**	**0.0075**	**0.0686**	**4.87 ± 2.48**

Table 2. Performance comparison of query methods across different protein representations. Initialization and query times (1, 10, and 100 queries) are measured on a database of 100 proteins using 35 cores. Memory requirement is reported as the average number of data points required to represent a protein: backbone atoms (RMSD), residues (BLOSUM), nodes (Fragment Graphs), or elements (Fragment Sets).

of fragments. We used RFDiffusion to fill in the gaps between the fragments and generate 5 different structures. Then, we use FoldSeek to search for the closest 10 backbones using sequence independent TMAlign. For each design and for the original backbone, we define recovery rate as the fraction of backbones annotated with the function of the original backbone (see Supplementary Figure 20). We also calculate the relative recovery rate as the recovery rate of the design over the recovery rate of the original backbone (see Figure 3 and boxplot in Supplementary Figure 21)

In general, there is a range of recovery rates, across different functional categories. Metal-binding proteins achieved perfect recovery rates, while ATP- and GTP-binding proteins also showed consistently high recovery rates. Multi-functional proteins demonstrated more variable outcomes, with DNA+ATP+GTP-binding showing the widest range, varying from 0% to 300% relative recovery rate and also the lowest recovery rate for the control.

Single-function designs generally demonstrated higher recovery rates compared to their multi-functional counterparts. Among dual-function proteins, metal-binding combinations proved most successful, with DNA+Metal, GTP+Metal, and RNA+Metal showing particularly high recovery rates. Interestingly, some triple-function combinations achieved surprisingly high recovery rates, particularly for DNA+ATP+GTP, DNA+RNA+Metal, and RNA+GTP+Metal binding.

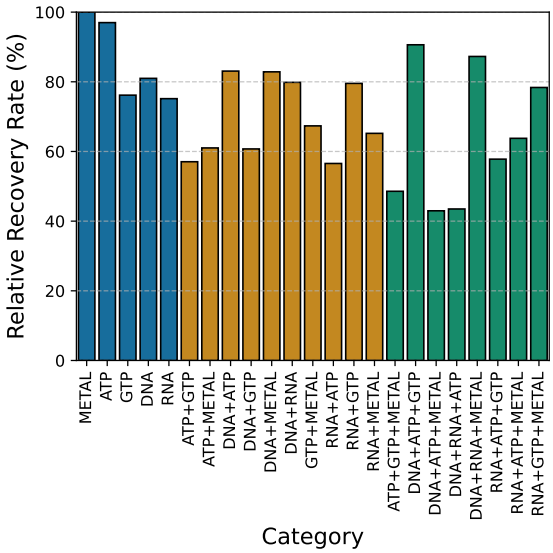

Fig. 3. Relative recovery rates for fragment-constrained backbone generation. The fragments from each protein in the Protein Function Dataset (PFD) were used to generate a template for RFDiffusion. The recovery rate is defined as the fraction of generated backbones whose top 10 structural matches in FoldSeek share the exact Gene Ontology (GO) function of the original protein. The relative recovery rate compares this to the recovery rate of the original protein backbone.

Fragments and Functional Similarity

We identified two DNA-binding proteins with high sequence and shape distances but low fragment distance (Figure 4). These proteins are the UvrABC system protein C, involved in DNA repair (PDB: 2NRR), and a viral DNA-dependent RNA polymerase (PDB: 6RIE). Despite their overall differences, they share fragments 17 (metal-binding), 23 (nucleotide-binding), and 35 (structural).

Fragment 17 is a small helix involved in metal binding, while fragment 23 is a helix-loop-sheet associated with nucleotide binding. Fragment 35, a sheet-loop-sheet motif, contributes to structural integrity. Notably, none of these fragments are explicitly classified as DNA-binding, yet their presence captures similarities in overall fold architecture. This is reflected in their low fragment distance scores compared to sequence (90% divergence) and shape (RMSD: 6.71 Å) distances.

Additionally, Graph Edit Distance (GED: 4%) accounts for the fragment neighborhood, considering factors such as the number of adjacent unknown fragments and peptide bonds.

Discussion

In this study, we demonstrate that fragment-based representations effectively coarsen protein structures while preserving essential functional information. Using just 40 evolutionarily conserved fragments, our approach captures important structural properties while reducing the dimensionality by up to 99% compared to traditional methods. Our fast and vectorised fragment-detection algorithm allows fast conversion to fragments and achieves an F1 score of 0.85. Furthermore, we successfully use fragments to guide backbone towards preserving protein functional signatures with recovery rates between 40-100%. These results make fragment-based representations a promising alternative to traditional sequence- and structure-based approaches for protein analysis, search, and design.

From Libraries to Detection: Building Fast and Robust Fragment Algorithms

We evaluated several metrics for fragment detection, focusing on shape and sequence. While individual metrics performed similarly, combining LogPr and RamRMSD nearly doubled the F1 score from 0.40 to 0.85.[19]. This highlights that torsion-angle alone can outperform sequence-based methods for robust fragment detection

While both metrics measure differences in backbone torsion angles ϕ and ψ, they process information differently. RamRMSD uses root-mean-square deviation, where squared differences are averaged, giving more weight to larger

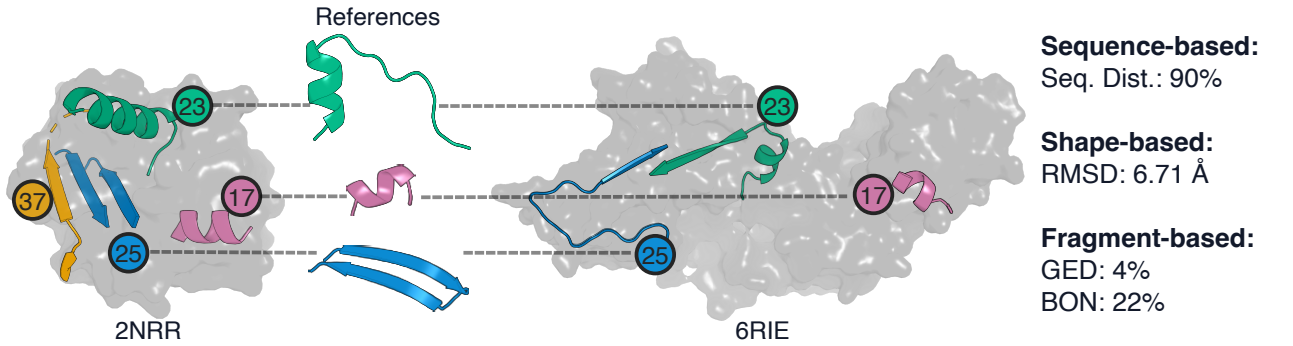

Fig. 4. Comparison of two structurally distinct DNA-binding proteins using sequence-, shape-, and fragment- based methods. The UvrABC system protein C, involved in DNA repair (PDB: 2NRR), is shown on the left, while a viral DNA-dependent RNA polymerase (PDB: 6RIE) is on the right. Despite significant differences in sequence and overall structure, both proteins share common functional fragments (17, 23, and 25, highlighted in color). Traditional sequence- and structure-based distances indicate high divergence between these proteins. In contrast, fragment-based metrics show a relatively low distance, suggesting a potentially shared functional role.

deviations. In contrast, LogPr applies a logarithmic transformation to normalized angle differences, emphasizing small deviations and converting them to a probability-like scale. This complementarity allows our algorithm to be sensitive to both, large and small deviations in torsion angles.

The fragment detection algorithm is written in Python and uses AMPAL [32] for parsing protein structures and NumPy's [16] vectorised convolutional operations. We deliberately avoided structural alignment methods based on incremental combinatorial extension (CE), which, despite potentially improving detection accuracy, proved computationally expensive and occasionally unstable during testing. [29, 9].

A key strength of our software is its flexibility. Users can easily swap our library with custom fragments by providing folders of PDB structures with the fragments of interest. This extends the software applications beyond the binding functions presented here, functions presented here, including enzyme design, antibody engineering, and *de novo* structural design. The software is written in Python and it is highly modular, meaning that users can expand it to integrate their own distance algorithms. Additionally, we use vectorized operations through NumPy, delivering fast performance while retaining the intuitive syntax that Python offers.

Fragment-based Representations Capture Functional Information

Using the fold-balanced PDBench dataset, we found that fragment regions capture distinct structural and chemical properties. These regions contained higher proportions of intra-fragment hydrogen bond, particularly in mainly β structures (+15%). This is consistent with β folds forming hydrogen bonds between adjacent strands[27]. On the other hand, inter-fragment hydrogen bonds were significantly lower in fragment regions, with a 47% reduction in mainly α structures. This observation is consistent with the characteristic hydrogen bonding pattern of α-helices, where hydrogen bonds stabilise the helical structure internally (i, $i+4$ pattern), reducing the potential for hydrogen bonds with adjacent fragments [27]. These results suggests that fragments may capture "self-contained" structural units. This is also supported by the reduced surface accessibility in fragment regions, such as the core of the protein, which is more likely to have folded regions, compared to surface exposed areas like loops [27].

Additionally, fragment-based representations outperform or match traditional methods in capturing functional similarities in embedding spaces. Both Fragment Graph (GraphEditDistance) and Set (BagOfNodes) metrics consistently achieved strong clustering scores, with BagOfNodes reaching a Silhouette score of 0.82 and GraphEditDistance showing the best overall performance for ARI (0.046) and F1 score (0.20). Fragment-based methods also preserved substantially more information at lower dimensions, achieving 95% and 80% cumulative variance compared to 60% and 40% for traditional sequence- and shape-based methods, respectively at 20 dimensions.

Notably, fragment-based representations capture these functional patterns without relying on the amino acid sequences. Instead they rely solely on backbone geometry. This effectiveness arises because fragments capture functional motifs regardless of their sequential arrangement, which is typical of other alignment-free analysis tools [34]. Sequence-alignment tools assume colinearity, meaning that they expect homologous residues to occur in the same order in both sequences [34]. Structural-alignment tools, such as combinatorial extension (CE), mitigate this by breaking the structure into smaller regions and reassemble them to complete the alignment [29]. However, these tools may struggle when there is little structural homology between proteins with the same function [15]. In contrast, fragment-based representations maintain performance by focusing on the presence of specific functional units. For example, Fragment Sets simply track the presence or absence of functional fragments, without their spatial precise arrangement.

Practical Implications of Fragments for Searching Protein Databases

Protein database searches are essential tools for finding structurally or functionally similar proteins. Traditional sequence- and shape-based methods can miss important relationships when functional motifs are arranged differently, for example when divided by other structural elements. Fragment-based representations overcome this limitation while delivering equal or better performance than traditional methods. Fragments require an initial processing cost to convert structures to graphs or sets. However, this one-time computation can be done for the entire dataset in advance

and it is quickly offset by the search speedups. In our benchmarks, Fragment Sets searches using BagOfNodes execute at fractions of a second (0.07s for 100 queries), over 500× faster than sequence-based BLOSUM searches (36.57s). Similarly, Fragment Graphs searches with GraphEditDistance are ∼3× faster that structure-based searches with RMSD (573s vs. 1717s for 100 queries), but are slower than sequence searches.

The major advantage, however, is memory efficiency. Fragment representations reduce the memory requirements by 90-99%, compared to traditional representations. For large-scale applications involving millions of proteins, this reduction could enable searches on hardware that would otherwise be insufficient for atom- or residue-level comparisons. These efficiency gains, coupled with comparable functional retrieval accuracy (as measured by AUROC and NDCG scores), make fragments an attractive alternative for the next-generation of protein search tools.

Fragment Constraints as Design Guides

The current generative tools for protein design are difficult to prompt for functional generation. For example, Ingraham et al. [18] highlight that there is currently no protein design system that can: (1) sample conditionally under diverse design constraints without retraining for new target functions, (2) with a sub-quadratic scaling computational efficiency, and (3) which integrates both sequence and structure modeling. For instance, RFDiffusion, a state-of-the-art diffusion model, lacks explicit mechanisms to enforce specific functional constraints in the generated structures.

Fragment-based constraints address this limitation by using evolutionarily conserved functional units to guide the generation process. Instead of retraining models with additional functional labels, our approach leverages evolutionarily conserved "building blocks" to steer generative models toward functionally relevant backbones.

We successfully generated functional-looking protein structures using fragments as RFDiffusion templates. Using our fragment-detection algorithm, we detected fragments in existing proteins and created partial backbones containing only these regions. We then used RFdiffusion to fill the connecting segments and used FoldSeek to retrieve the closest proteins available. On a dataset of various functional categories, we successfully generated structures that maintained the functional signatures of the original proteins. Recovery rates varied by functional category, with metal-binding and ATP-binding proteins achieving nearly perfect recovery (∼100%). Our approach was particularly effective for certain multi-functional proteins, with DNA+ATP+GTP and DNA+RNA+Metal combinations showing surprisingly high recovery rates despite their complexity.

These results suggest that diffusion models have implicitly learned about evolutionarily conserved fragments and are able to use them for design. Explicitly incorporating fragment representations in these models could help reduce the computational complexity while also providing more direct functional control to generate specific functional proteins.

Limitations and Future Work

Our current implementation uses a curated library of 40 fragments spanning functions of DNA, RNA, GTP, ATP, and Metal binding. Further studies could explore data-driven approaches to discover novel fragments with unsupervised learning, potentially expanding the representation capacity beyond the functions described here.

A major advantage of our approach is its inherent interpretability. Unlike traditional black-box methods, fragment-based representations provide clear functional insights as they are associated with specific structural motifs and known biological roles. This interpretability could improve generative models by making their outputs more functionally interpretable and allow more control during the design process. Additionally, for protein sequence design, classifying fragments sequences instead of individual amino acid in the backbone could be faster and take into account the sequence bias defined by the fragment function.

While the BagOfNodes approach is very fast, it is less effective when multiple instances of a fragment contribute to distinct functional roles. For example, Zinc finger proteins usually contain three instances of fragment 14, each binding a positively-charged Zinc ion, and all binding negatively-charged DNA (See Figure 1). More instances of fragment 14 may indicate binding to multiple DNA strands or different regions of the same strand. In these cases, Fragment Graphs with GraphEditDistance provide a more nuanced representation by capturing the connectivity and fragment context, despite their higher computational cost.

Our fragment detection algorithm achieves a good F1 score of 0.85, but is potentially sensitive to subtle variations in torsion angles which could lead to misclassification. For example, a large change in torsion angles of the middle amino acid of a fragment would change the backbone angles for one amino acid only, so it might still be classified similarly. Future work could integrate probabilistic models to quantify detection confidence and providing adjustable sensitivity, allowing users to choose the settings based on their design scenario.

Beyond protein design, fragment-based representations improve biosecurity applications by identifying potentially hazardous structural motifs for that might escape detection in sequence- or structure- based screening systems. By recognizing functional fragments, regardless of their arrangement in the proteins, our approach could provide an additional layer of safety for protein synthesis services.

Conclusion

We introduced a fragment-based protein representation that encodes structures using a curated library of 40 evolutionarily conserved functional fragments. This approach reduces dimensionality by up to 99% while preserving functional and structural information.

Our evaluations demonstrate that fragment-based representations capture functional relationships more effectively than traditional methods in clustering, enable significantly faster database searches with comparable accuracy, and successfully guide RFDiffusion to generate backbones with functional signatures. Unlike black-box representations, our method provides interpretability by linking fragments to biological functions.

Fragment-based representations offer a scalable and biologically relevant framework for protein design. By balancing efficiency and interpretability, this approach lays the foundation for the next generation of protein design tools.

Competing interests

No competing interest is declared.

Author contributions statement

All authors were involved in the conception of the project and the writing of the manuscript. CWW and KS supervised the work. LVC developed the code, ran the experiments, and produced the figures.

Acknowledgments

LVC thanks Lorenzo Pisani for his valuable guidance in developing the protein fragment library software.

References

1. Sarah Alamdari, Nitya Thakkar, Rianne Van Den Berg, Neil Tenenholtz, Robert Strome, Alan M. Moses, Alex X. Lu, Nicolò Fusi, Ava P. Amini, and Kevin K. Yang. Protein generation with evolutionary diffusion: Sequence is all you need, September 2023.
2. Vikram Alva and Andrei N Lupas. From ancestral peptides to designed proteins. *Current Opinion in Structural Biology*, 48:103–109, February 2018.
3. Vikram Alva, Johannes Söding, and Andrei N Lupas. A vocabulary of ancient peptides at the origin of folded proteins. *eLife*, 4:e09410, December 2015.
4. 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, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000.
5. H. M. Berman. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, January 2000.
6. Alice Bizeul, Thomas Sutter, Alain Ryser, Bernhard Schölkopf, Julius von Kügelgen, and Julia E. Vogt. From Pixels to Components: Eigenvector Masking for Visual Representation Learning, 2025.
7. Leonardo V Castorina, Rokas Petrenas, Kartic Subr, and Christopher W Wood. PDBench: Evaluating computational methods for protein-sequence design. *Bioinformatics*, 39(1):btad027, January 2023.
8. Leonardo V Castorina, Suleyman Mert Ünal, Kartic Subr, and Christopher W Wood. TIMED-Design: Flexible and accessible protein sequence design with convolutional neural networks. *Protein Engineering, Design and Selection*, 37:gzae002, January 2024.
9. Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, June 2009.
10. J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning based protein sequence design using ProteinMPNN, June 2022.
11. Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):4348, July 2022.
12. Andreas Fischer, Kaspar Riesen, and Horst Bunke. Improved quadratic time approximation of graph edit distance by combining Hausdorff matching and greedy assignment. *Pattern Recognition Letters*, 87:55–62, February 2017.
13. Vincent Frappier, Justin M. Jenson, Jianfu Zhou, Gevorg Grigoryan, and Amy E. Keating. Tertiary Structural Motif Sequence Statistics Enable Facile Prediction and Design of Peptides that Bind Anti-apoptotic Bfl-1 and Mcl-1. *Structure*, 27(4):606–617.e5, April 2019.
14. P. Gainza, F. Sverrisson, F. Monti, E. Rodolà, D. Boscaini, M. M. Bronstein, and B. E. Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, February 2020.
15. Tymor Hamamsy, James T. Morton, Robert Blackwell, Daniel Berenberg, Nicholas Carriero, Vladimir Gligorijevic, Charlie E. M. Strauss, Julia Koehler Leman, Kyunghyun Cho, and Richard Bonneau. Protein remote homology detection and structural alignment using deep learning. *Nature Biotechnology*, 42(6):975–985, June 2024.
16. Charles R. Harris, K. Jarrod Millman, Stéfan J. Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. Van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández Del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
17. S Henikoff and J G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, November 1992.
18. John B. Ingraham, Max Baranov, Zak Costello, Karl W. Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M. Lord, Christopher Ng-Thow-Hing, Erik R. Van Vlack, Shan Tie, Vincent Xue, Sarah C. Cowles, Alan Leung, João V. Rodrigues, Claudio L. Morales-Perez, Alex M. Ayoub, Robin Green, Katherine Puentes, Frank Oplinger, Nishant V. Panwar, Fritz Obermeyer, Adam R. Root, Andrew L. Beam, Frank J. Poelwijk, and Gevorg Grigoryan. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, November 2023.
19. Sunghoon Jung, Se-Eun Bae, Insung Ahn, and Hyeon S. Son. Protein Backbone Torsion Angle-Based Structure Comparison and Secondary Structure Database Web Server. *Genomics & Informatics*, 11(3):155, 2013.
20. Rachel Kolodny, Sergey Nepomnyachiy, Dan S Tawfik, and Nir Ben-Tal. Bridging Themes: Short Protein Segments Found in Different Architectures. *Molecular Biology and Evolution*, 38(6):2191–2208, May 2021.
21. Craig O Mackenzie and Gevorg Grigoryan. Protein structural motifs in prediction and design. *Current Opinion in Structural Biology*, 44:161–167, June 2017.
22. Fabrizio Mastrolorito, Fulvio Ciriaco, Maria Vittoria Togo, Nicola Gambacorta, Daniela Trisciuzzi, Cosimo Damiano Altomare, Nicola Amoroso, Francesca Grisoni, and Orazio Nicolotti. fragSMILES as a chemical string notation for advanced fragment and chirality representation. *Communications Chemistry*, 8(1):26, January 2025.

23. Geraldene Munsamy, Ramiro Illanes-Vicioso, Silvia Funcillo, Ioanna T. Nakou, Sebastian Lindner, Gavin Ayres, Lesley S. Sheehan, Steven Moss, Ulrich Eckhard, Philipp Lorenz, and Noelia Ferruz. Conditional language models enable the efficient design of proficient enzymes, May 2024.
24. Yifei Qi and John Z. H. Zhang. DenseCPD: Improving the Accuracy of Neural-Network-Based Computational Protein Sequence Design with DenseNet. *Journal of Chemical Information and Modeling*, 60(3):1245–1252, March 2020.
25. Jane S. Richardson. Early ribbon drawings of proteins. *Nature Structural Biology*, 7(8):624–625, August 2000.
26. Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences of the United States of America*, 2019.
27. Georg E. Schulz and R. Heiner Schirmer. *Principles of Protein Structure*. Springer Advanced Texts in Chemistry. Springer New York, New York, NY, 1979.
28. Fabian Sesterhenn, Che Yang, Jaume Bonet, Johannes T. Cramer, Xiaolin Wen, Yimeng Wang, Chi-I Chiang, Luciano A. Abriata, Iga Kucharska, Giacomo Castoro, Sabrina S. Vollers, Marie Galloux, Elie Dheilly, Stéphane Rosset, Patricia Corthésy, Sandrine Georgeon, Mélanie Villard, Charles-Adrien Richard, Delphyne Descamps, Teresa Delgado, Elisa Oricchio, Marie-Anne Rameix-Welti, Vicente Más, Sean Ervin, Jean-François Eléouët, Sabine Riffault, John T. Bates, Jean-Philippe Julien, Yuxing Li, Theodore Jardetzky, Thomas Krey, and Bruno E. Correia. De novo protein design enables the precise induction of RSV-neutralizing antibodies. *Science*, 368(6492):eaay5051, May 2020.
29. Ilya N Shindyalov and Philip E Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein engineering*, 11(9):739–747, 1998.
30. Michel Van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42(2):243–246, February 2024.
31. Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, August 2023.
32. Christopher W Wood, Jack W Heal, Andrew R Thomson, Gail J Bartlett, Amaurys Á Ibarra, R Leo Brady, Richard B Sessions, and Derek N Woolfson. ISAMBARD: An open-source computational environment for biomolecular analysis, modelling and design. *Bioinformatics*, 33(19):3043–3050, October 2017.
33. Andy Hsien-Wei Yeh, Christoffer Norn, Yakov Kipnis, Doug Tischer, Samuel J. Pellock, Declan Evans, Pengchen Ma, Gyu Rie Lee, Jason Z. Zhang, Ivan Anishchenko, Brian Coventry, Longxing Cao, Justas Dauparas, Samer Halabiya, Michelle DeWitt, Lauren Carter, K. N. Houk, and David Baker. De novo design of luciferases using deep learning. *Nature*, 614(7949):774–780, February 2023.
34. Andrzej Zielezinski, Susana Vinga, Jonas Almeida, and Wojciech M. Karlowski. Alignment-free sequence comparison: Benefits, applications, and tools. *Genome Biology*, 18(1):186, December 2017.