Published on Sep 28, 2023 by Arcadia Science **DOI**: 10.57844/arcadia-a5a6-1068

ProteinCartography: Comparing proteins with structure-based maps for interactive exploration

The ProteinCartography pipeline identifies proteins related to a query protein using sequence- and structure-based searches, compares all protein structures, and creates a navigable map that can be used to look at protein relationships and make hypotheses about function.

Contributors (A-Z)

Prachee Avasthi, Brae M. Bigge, Adair L. Borges, Feridun Mert Celebi, Keith Cheveralls, Rachel J. Dutton, Jase Gehring, Megan L. Hochstrasser, Galo Garcia III, Cameron Dale MacQuarrie, David Q. Matus, Elizabeth A. McDaniel, Erin McGeever, Gilad Mishne, Manon Morin, Atanas Radkov, Taylor Reiter, Michael E. Reitman, Dennis A. Sun, Emily C.P. Weiss

Version 2 · Mar 31, 2025

Purpose

In the ProteinCartography pipeline, we use protein structural comparisons to generate interactive maps of protein families for exploration and discovery. This kind of analysis can be useful for provoking hypotheses about what properties could be driving functional differences within protein families and identifying outlier proteins where innovations might be found.

We're presenting our initial version of the pipeline, which contains the core functionality, but we intend to continue improving the pipeline itself and adding features in future versions. For additional information about what's coming, jump to the Next Steps section. Check back for new releases and updates!

- This pub is part of the **platform effort**, "Functional annotation: mapping the functional landscape of protein families across biology." Visit the platform narrative for more background and context.
- The **ProteinCartography pipeline** is available in this GitHub repository. Try it yourself and let us know what you think!
- We've included several examples throughout the pub. The **code** for that analysis and the resulting figures are available in the same GitHub repository and the associated **data** are on Zenodo.

The strategy

All organisms, from single-celled bacteria to multicellular animals, share common types of basic building blocks, including proteins. Comparing proteins across the tree of life can help us understand how different organisms have evolved distinct traits and discover novel biology. Recent tools that enable searches based on structural similarity, including Foldseek, have made it possible to compare proteins from diverse organisms in new, and perhaps more informative, ways **[1]**. We developed a pipeline that facilitates comparative protein biology by leveraging these emerging tools to enable users to interactively explore protein families.

The problem

Comparative protein biology is an important and rapidly progressing field. Amino acid sequences are widely used for these analyses due to the abundance and ease of working with sequence data, but there are disadvantages to such methods. For example, small protein sequence changes can result in dramatic structural changes that alter the function of the protein, and conversely, proteins with low sequence

similarity can have similar folds and perform similar functions **[2]**. Comparisons of protein structure could overcome these limitations, as structures are generally more conserved than protein sequences and are more closely tied to protein function **[3]**. Historically, researchers have been limited by the availability of experimentally determined structures, but recent advances in protein folding prediction tools, such as AlphaFold and ESMFold, and protein search tools like Foldseek have brought us into a new era of protein analysis **[4][1][5][6]**.

Three main methods are typically applied to represent protein space: classification, networks, and maps **[7]**. These can be created based on sequence, structure, or other characteristics. Classification sorts proteins into hierarchical categories. For example, SCOP (structural classification of proteins) and CATH (class, architecture, topology, homologous superfamily) databases sort protein domains into categories based on folds or structure **[8][9][10]**. Networks represent proteins as nodes that are connected to related proteins by edges **[2][7]**. The most common type of network is a sequence similarity network (SSN), where protein nodes are connected by edges that represent some sequence similarity threshold defined by the user. Networks are useful because they can be used to cluster proteins into sub-groups. Finally, maps visualize a high-dimensional protein space representing complex information (like protein structural characteristics) as a collection of points in a low-dimensional space, often generated via classic dimensionality reduction tools like principal component analysis (PCA) and multidimensional scaling (MDS) **[11][12]**.

Many of the analyses done with these three methods are aimed at understanding the whole protein universe, or all protein structures that have been experimentally solved or predicted **[7][8][9][10][11][12][13][14][15][16]**. While these analyses are extremely useful for understanding large-scale protein evolution and for understanding how proteins as a whole relate to each other, they are computationally complex and can be difficult to interpret if you want to know more about individual proteins or protein families.

Our solution

We developed a pipeline to rapidly and intuitively identify and visualize groups of proteins with similar structures across user-defined protein families (Figure 1). The ProteinCartography pipeline uses a combination of networks for clustering analysis and maps for visualization, and focuses these protein space representations at the

protein family level to allow for rapid and intuitive analyses. The pipeline starts with a protein of interest provided by the user and searches available sequence and structure databases. After obtaining the AlphaFold-predicted structures of each match, the pipeline uses Foldseek to perform all-v-all structural comparison, which it uses to generate a similarity network for identifying groups of structurally related proteins. The pipeline then performs dimensionality reduction to create a visual "map" for exploratory analysis. Informative protein features can be overlaid on the map, such as cluster association, taxonomy, sequence conservation to the query, and annotation information. This allows you to generate hypotheses about what properties could be driving functional differences within protein families and identify outlier proteins where innovations might lie across taxa.

In this pub, we'll take you through some general uses of the ProteinCartography pipeline, as well as an example of how the ProteinCartography pipeline works, what the results look like, and how to analyze them. This is all contained in the "ProteinCartography in action" section directly after this paragraph. For more in-depth information about the limitations of ProteinCartography and individual steps and parameters of the pipeline, see the "Comprehensive overview of the pipeline" section. To learn more about plans we have for improving the pipeline see the "Next steps" section and to provide feedback check out the "What do you think?" section.

> **TRY IT**: The ProteinCartography pipeline is available in this GitHub repository (DOI: 10.5281/zenodo.8388050), along with instructions to get started.

ProteinCartography in action

You can use the ProteinCartography pipeline to generate hypotheses and make predictions about individual proteins. For example, it can identify proteins that are structurally similar to an input protein, or it can identify outlier proteins. Downstream analyses could tell you which regions of the protein are important for function, and further investigation could determine whether these protein regions differ across clusters. Additionally, you could use the ProteinCartography pipeline to annotate proteins of unknown function or to provide support for annotation predictions.

The pipeline also lets you explore subfamilies within larger protein families. For example, it could be used to make hypotheses about whether distantly related proteins in the same family are members of the same subfamily. You could use it to identify especially interesting subfamilies for further examination (like subfamilies composed of only proteins from a particular taxonomic group). This is perhaps the most common use we've encountered so far, and we'll dive into this use case more below. Importantly, we use the pipeline as a starting point to generate hypotheses and make predictions, but encourage users to test their hypotheses and predictions with additional analyses.

Running the pipeline

Before we discuss the results of the pipeline and how to interpret them, we provide a brief walkthrough of how a run of the ProteinCartography pipeline typically works. To jump to a detailed description of each step in the "Comprehensive overview of the pipeline" section, click the link at the beginning of each step below.

The pipeline generally starts with a protein of interest, or input protein, but it can start with multiple proteins. A PDB (structure) file and/or FASTA (sequence) file are required for each input protein. The package provides utilities to fetch these from UniProt or AlphaFold based on accession number, or to fold short sequences (less than 400 amino acids) using ESMFold **[4][17]**. For proteins longer than 400 amino acids, you can use outside tools like ColabFold to fold your proteins and import them into the pipeline **[18]**.

1. Sequence-based search: The pipeline performs a protein BLAST search against the NCBI non-redundant (nr) database to identify proteins based on sequence similarity to the input **[19]**.
2. Structure-based search: The pipeline performs a Foldseek search against the AlphaFold/UniProt50 v4, AlphaFold/Swiss-Prot v4, and AlphaFold/Proteome v4 databases to identify proteins based on structural similarity to the input **[1][5][6][16]**.
3. Aggregate and filter hits: The pipeline combines BLAST and Foldseek results and filters hits based on a user-defined length, fragment status, and whether the protein is marked as inactive.

4. Download hits: Using the combined hits list, the pipeline downloads PDB files from the AlphaFold database and metadata from UniProt for all hits. Metadata include fields such as taxonomic information, annotation information, protein characteristics, and others **[5][6][17]**.
5. Construct all-v-all similarity matrix: Using the downloaded structure files, the pipeline compares the structure of every protein to the structure of every other protein to identify the best matches using `foldseek search`. For those matches, the pipeline uses `foldseek aln2tmscore` to calculate a similarity score, or a TM-score, where generally, a value of 1 means two structures are identical and values closer to zero mean the structures are less similar **[1][20]**. These scores are aggregated in an all-v-all similarity matrix. For more information about TM-scores, see the "Construct all-v-all similarity matrix" section.
6. Cluster proteins: The all-v-all similarity matrix is then used to cluster proteins into groups of similar proteins. The pipeline uses two different clustering algorithms, Foldseek's TM-align greedy set clustering and the Leiden algorithm **[1][21]**. While both are provided in the final results, we default to Leiden clustering for visualizations. The Leiden algorithm is a clustering method that iteratively groups proteins, in this case attempting to optimize the modularity of the network **[21]**. More information on clustering algorithms and why we default to Leiden clustering can be found in the detailed "Cluster proteins" section.
7. Calculate cross-cluster similarity matrix: Once clusters have been created, the pipeline performs cluster-related analyses, including calculating the cross-cluster similarity matrix. The cross-cluster similarity matrix is a heatmap representing the mean TM-score of all structures in each cluster versus all other proteins in each other cluster. The diagonal of this matrix tells us how similar all proteins are within a cluster. We average the values of the diagonal to determine a "cluster compactness" score that we use as a heuristic for how well-clustered the proteins are overall.
8. Perform semantic analysis: Using clustering information and metadata obtained from UniProt, the pipeline performs semantic analysis to evaluate and visualize the most frequently occurring existing annotations for each cluster **[17]**. The pipeline determines the most common annotations per cluster and shows them as individual bar charts, as well as the most commonly used annotation words per cluster and shows them as word clouds.

9. Create maps: The pipeline uses the all-v-all similarity matrix of TM-scores to perform dimensionality reduction and create a map of the protein family. It uses the all-v-all similarity matrix to calculate a principal component analysis (PCA), which it then uses to calculate a t-distributed stochastic neighbor embedding (t-SNE) and a uniform manifold approximation projection (UMAP) **[22][23]**. For more information on these dimensionality reduction techniques, see the "Create map" section. Both t-SNE and UMAP result in 2D maps that are meant for visualization.
10. Generate interactive plots: To make t-SNE and UMAP plots maximally useful, the pipeline uses the Plotly Python package **[24]** to create an interactive and navigable map with color overlays corresponding to protein metadata, including length, broad taxon, TM-score to input from Foldseek local search, source (Foldseek vs. BLAST), average pLDDT (structure quality), and others.

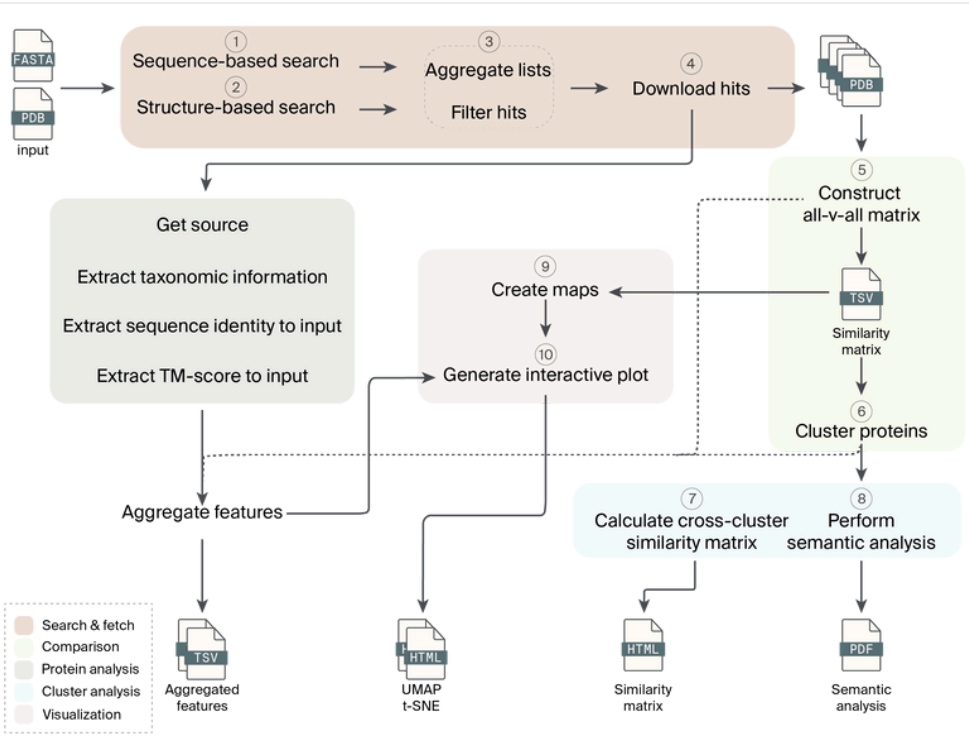

Figure 1

ProteinCartography at a glance.

ProteinCartography starts with a user input (FASTA and/or PDB) and then runs proteins through both a BLAST and Foldseek search. It then fetches structures of identified hits and metadata for each protein. It performs a structural comparison step and then groups proteins with similar structures together in clusters. Additionally, it does analysis based on each protein and based on each cluster. Finally, it combines all of the information gathered throughout the analysis and uses it to generate an interactive map of the data for exploration of the protein family.

Visualizing the mitogen-activated protein kinase 10 (MAPK10) family with ProteinCartography outputs

As an example, we ran mitogen-activated protein kinase 10 (MAPK10), also called c-Jun N-terminal kinase 3 (JNK3), one of the top 200 most-studied human proteins,

through the ProteinCartography pipeline **[25]**. We refer to this protein as MAPK10 throughout this pub and abbreviate it as MK10 in figures. For the input, we used P53779 (Figure 2, A). The pipeline carried out the steps listed above and produced the following outputs, which we explore further in the next section:

- A UniProt features TSV file containing a summary of the UniProt metadata as well as clustering information:

tsv `MAPK_aggregated_features.tsv` Download

- An HTML file containing cross-cluster similarity matrix (Figure 3)
- An HTML file containing interactive t-SNE plot with color overlays (Figure 4)
- An HTML file containing interacting UMAP plot with color overlays (Figure 5)
- An HTML file containing semantic analysis (Figure 6)

A summary of the outputs relevant to our interpretation is in Figure 2, but we will go through each output throughout the section.

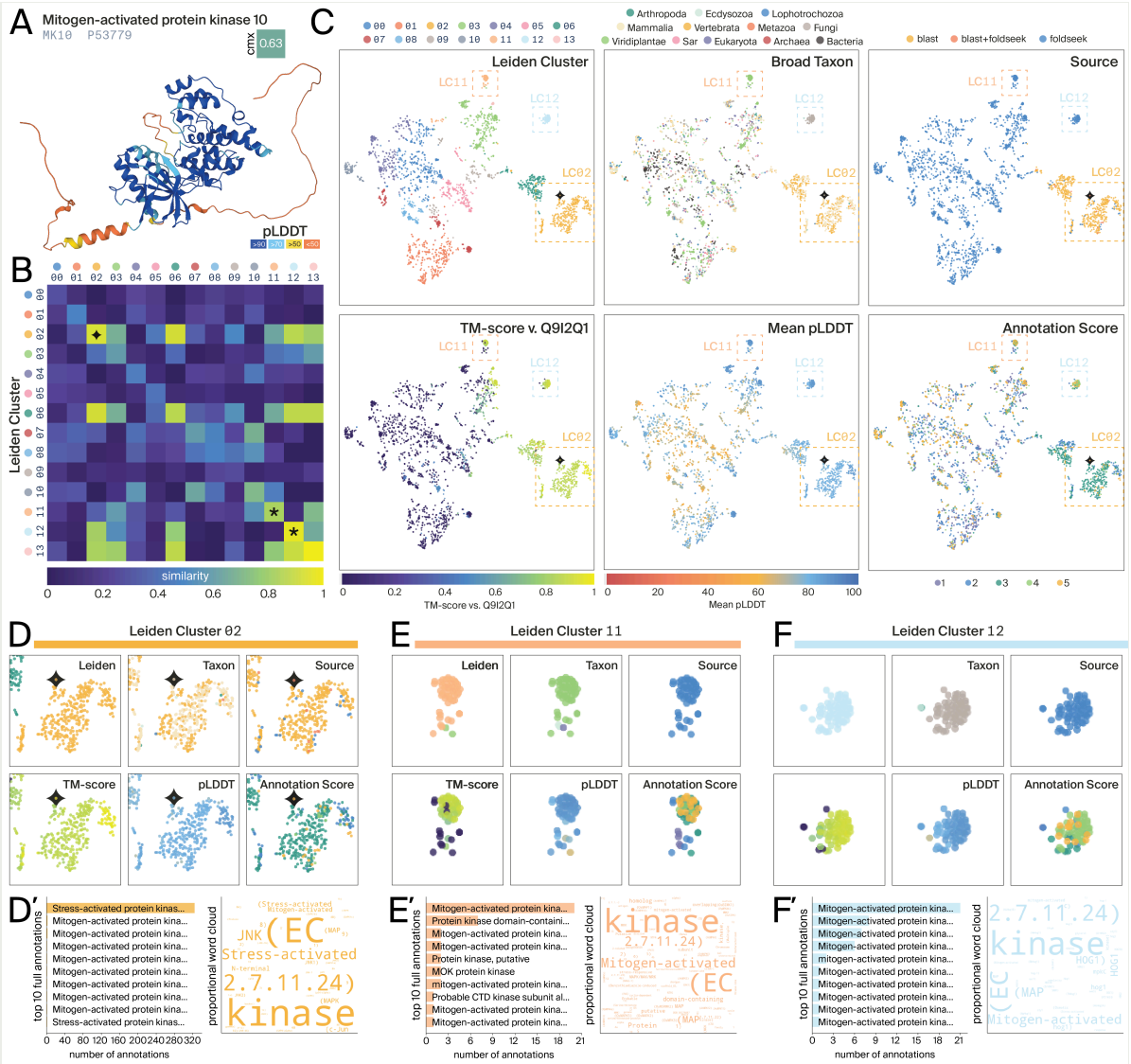

Figure 2

Fungal and plant MAPK10 proteins identified using ProteinCartography.

(A) The structure of the MAPK10, showing the protein generally has a well-defined structure with some disordered edges and a confident AlphaFold prediction.

(B) Similarity matrix for the clustering of MAPK10 hits shows a diagonal with a higher within-cluster TM-score, suggesting the clusters are compact. The protein treated as the query in the comparison is on the y-axis, and the target is on the x-axis. Colored dots along the axes of the chart correspond to the colors of the Leiden cluster shown in the maps in C–F. The cluster containing the input protein is marked with a four-pointed star marker. The two other clusters we focus on further are annotated with asterisks.

(C) t-SNE visualization created for MAPK10 and the proteins identified as similar to it. The overlay applied to the map is shown in the upper right corner of each graph, but briefly we show Leiden cluster, broad taxon, source of the protein, the TM-score to the input protein, the average pLDDT of each protein, and the annotation score of each protein. The star in each map represents the input protein, and the dotted boxes show the three clusters that we focus on in D–F.

(D) Zoom in of LC02, the cluster that contains the input protein.

(D′) Semantic analysis of LC02, which contains the input protein.

(E) Zoom in of LC11, interesting because of its compactness and because it is composed of primarily proteins from plants.

(E′) Semantic analysis of LC11.

(F) Zoom-in of LC12, interesting because of its compactness and because it is composed primarily of proteins from fungi.

(F′) Semantic analysis of LC12.

Exploring the MAPK10 family with ProteinCartography

Mitogen-activated protein kinase 10 (MAPK10), or c-Jun N-terminal kinase 3 (JNK3), is a serine/threonine kinase that is a member of the MAP kinase family. MAP kinases are involved in a number of cellular functions, including everything from proliferation to apoptosis **[26]**. These proteins form signaling cascades, or chains of interactions that result in a final signal being delivered. This particular kinase, MAPK10, is a neuronal kinase that is often involved in stress response, where its activation results in phosphorylation of several transcription factors that result in neuronal apoptosis **[27]**.

MAP kinases are found in almost all eukaryotic organisms, but individual MAP kinases are not always well conserved. In particular, MAP kinases in the JNK pathway seem to have emerged more recently in evolutionary time, as they are usually only described in vertebrates **[28]**, with a somewhat similar pathway described in yeast as the “HOG pathway” **[29]**. Using the ProteinCartography pipeline, we can ask whether MAPK10-

like proteins exist in earlier-diverging organisms, like fungi and plants. Additionally, we can ask if these MAPK10-like proteins are structurally similar or distinct based on how they cluster, and we can identify clusters and proteins for further computational or experimental analysis.

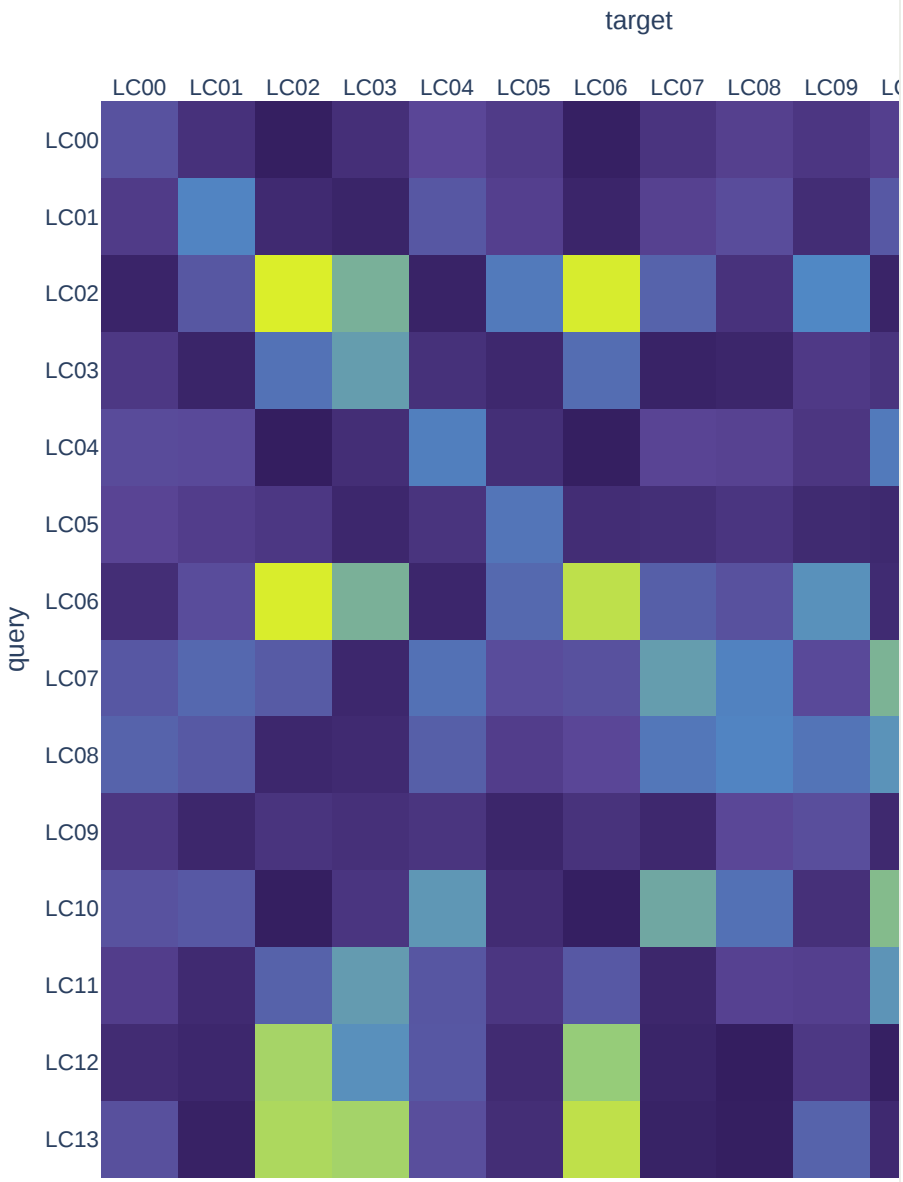

Figure 3

Cross-cluster similarity matrix for MAPK10 suggests some clusters are compact and distinct.

This interactive cross-cluster similarity matrix is a visual representation that shows the mean TM-score of all of the structures in each cluster versus all other proteins in each other cluster. The protein treated as the query in the comparison is on the y-axis, and the target is on the x-axis. The diagonal of the heatmap shows how similar proteins are within clusters, or how compact clusters are. The input

protein for this analysis can be found in LC02. LC: Leiden cluster.

You can view a static version in Figure 2, B.

First, to determine how well the clustering performed, we looked at the cross-cluster similarity matrix (Figure 2, B and Figure 3). The diagonal of the matrix shows how structurally similar proteins are within each cluster, or how compact the clusters are. We refer to the average value of the diagonal as “cluster compactness.” The goal of clustering is to group similar proteins together, so when clusters contain structures that are not very similar, it can suggest issues with the clustering that may have to do with the proteins themselves or the clustering parameters. Cluster compactness doesn’t take into account all the ways that the pipeline might fail or succeed, but it does give us a general idea of whether a run produced interpretable results. We dig more into this in the “Testing the limits of the pipeline section.” We also consider cluster distinctness (how similar each cluster is to other clusters) when evaluating clustering. In this case, the clusters are not very distinct, but we expect this is because we’re evaluating a family of closely related proteins (Figure 2, B and Figure 3).

For this MAPK10 map, the cluster compactness value is 0.63 (Figure 2, B and Figure 3), which is around average for the analyses conducted for this pub. Among the clusters in this analysis, Leiden clusters 11 (LC11) and 12 (LC12) drew our attention, as they are compact, suggesting that they might hold proteins that are well clustered. Looking beyond the diagonal, the similarity matrix also tells us which clusters are similar to each other. In this example, we see that LC02 and LC06 are quite similar and may be more related to each other than they are to other proteins in the map (Figure 2, B and Figure 3).

Figure 4

Interactive t-SNE plot with color overlays for MAPK10.

The input protein is represented by a four-pointed star, which you can toggle on and off using the “Input Proteins” button. You can change the color overlay using the “color” drop-down. More information on the color overlays themselves is in the “Plot overlays” section. LC: Leiden cluster.

Figure 5

Interactive UMAP plot with color overlays for MAPK10.

The input protein is represented by a four-pointed star, which you can toggle on and off using the “Input Proteins” button. You can change the color overlay using the “color” drop-down. More information on the color overlays themselves is in the “Plot overlays” section. LC: Leiden cluster.

We next looked at the clusters themselves. Our protein of interest appears in LC02 (Four-pointed star in Figure 2, C–D, Figure 4, and Figure 5). In addition to looking at which cluster our protein belongs to, overlaying additional information on the map provides more insight. For example, the pipeline categorizes proteins based on their broad taxonomic grouping. By examining the taxonomic groupings of proteins in the neighborhood of our input protein, we observe that our protein and the surrounding proteins originate from mammals and other vertebrates ("Taxon" panels in Figure 2, C–D; "Broad taxon overlay" panels in Figure 4 and Figure 5). It's important to note that the taxonomic depth is not uniform and is instead chosen to be generally interpretable and useful to people while staying within the limitation presented by the available number of colors. Advanced users can also customize the taxonomic groups and colors based on their organisms of interest. In this view, we observe that LC12, one of the tight clusters we saw in Figure 2, B, contains primarily fungal proteins, whereas LC11 is composed of primarily plant proteins ("Taxon" panels in Figure 2, D–F; "Broad taxon overlay" panels in Figure 4 and Figure 5).

Intrigued that we identified clusters of fungal and plant proteins in our MAPK10 analysis, we explored features calculated by the pipeline to determine if these proteins really are MAPK10 or JNK proteins. We looked at the quality of predicted structures (mean pLDDT), and found that the proteins in these clusters were high-quality, or closer to 100 ("pLDDT" panel in Figure 2, C–F; "pLDDT" overlay in Figure 4 and Figure 5). We also looked at the structural similarity to our input (TM-score to P53779) and saw that the structures in LC11 and LC12 were generally structurally related to our input protein ("TM-score" panel in Figure 2, C–F; "TM-score to input" overlay in Figure 4 and Figure 5). Note that the TM-score to input values shown in the maps are calculated during the all-v-all comparison step of the pipeline. During this step, TM-scores are only calculated for pairs of proteins that meet the default threshold. The rest of the proteins are marked as zero. For more information see the "Construct all-v-all similarity matrix" section.

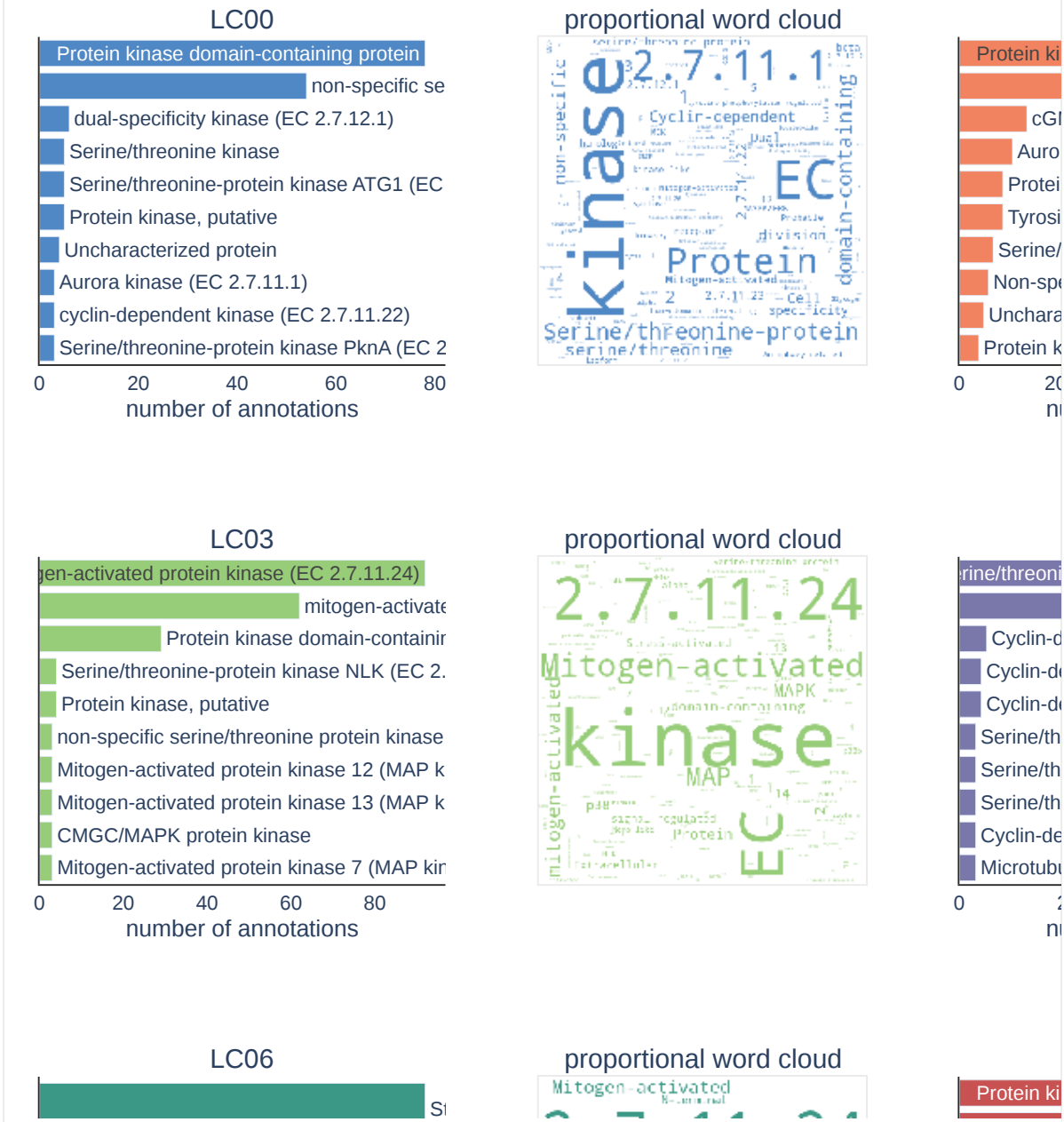

Figure 6

Semantic analysis of MAPK10 provides a human-readable method for understanding cluster composition.

Each cluster has a different-colored ranked bar chart and word cloud that correspond to their Leiden cluster color in the interactive maps. The bar chart summarizes the most common full annotation and the word cloud summarizes the most common annotation words.

To provide additional context, the pipeline generates simple semantic analysis visualizations that summarize existing annotation information for proteins retrieved

from UniProt (Figure 6). Comparing the three clusters of interest, we see the top annotation for the cluster containing our input protein is “Stress-activated protein kinase JNK (EC 2.7.11.24)”, while the top annotation in LC11 is “Mitogen-activated protein kinase (EC 2.7.11.24)” and in LC12 is “Mitogen-activated protein kinase HOG1 (MAP kinase HOG1) (EC 2.7.11.24)” (Figure 2, D′, E′, F′, Figure 6). “EC 2.7.11.24” refers to the enzyme class to which these MAPK proteins belong, suggesting that they do all fit in the same enzyme class. The HOG pathway from yeast is similar to the JNK pathway **[29]**, so it makes sense that LC12, which is primarily composed of yeast proteins, would contain many proteins annotated as a HOG1 protein. To determine how useful these annotations are, we can overlay the UniProt annotation score on the map (“Annotation score” panels in Figure 2, C–F; “Annotation score overlay” drop-down in Figure 4 and Figure 5). The annotation score is assigned by UniProt and ranges from 1 to 5, where a score of 5 means that the annotation is backed by experimental evidence and a score of 1 generally means that annotations were predicted or inferred **[17]**. For both of these clusters, there are several proteins supported by an annotation score of 4 or 5, suggesting that at least some of the annotations in each cluster are likely backed by experimental evidence (“Annotation score overlay” panels in Figure 2, C–F; “Annotation score overlay” drop-down Figure 4, and Figure 5).

Thus, using the ProteinCartography pipeline, we could now hypothesize that there are MAPK10 or MAPK10-like proteins in fungi and plants. However, we would want to test these hypotheses with additional experiments.

Pursuing hypotheses generated with ProteinCartography

From a large list of candidates, the pipeline helped us identify specific groups of proteins of interest in diverse taxonomic groups and make predictions about their function in relation to our input. In this case, it’s especially interesting to note that the sequence identity of the proteins identified in plants and fungal species is quite low (~30%) and these were identified via Foldseek, suggesting that the pipeline was able to identify relatives that would have been missed using a BLAST search alone (Figure 3, Figure 4). However, because the pipeline provides information based on predicted protein structures, further analysis would be necessary to draw definitive conclusions about protein function. For this example, downstream analyses like assessing the presence or absence of interacting proteins upstream and downstream of MAPK10 in

the signaling cascade, looking for the conservation of specific and known catalytic residues, determining evolutionary history, or performing biochemical assays to directly test protein function, could be used to help determine the function of proteins of interest.

Comprehensive overview of the pipeline

For users interested in learning more about the inner workings of the ProteinCartography pipeline, the following section dives into the details and parameters that we used to build it. It also provides additional in-depth information for each step and a meta-analysis that tests the limits of the pipeline. To jump straight to the next section, “Next steps,” click here.

To run the pipeline, clone the GitHub repository and follow the instructions there for installation. The current version of the pipeline takes around 30 to 90 minutes to run for small- to average-sized (< 400 amino acids) proteins, assuming default search parameters.

> **TRY IT**: The ProteinCartography pipeline is available in this GitHub repository, along with instructions to get started.

Input proteins

To run the pipeline, you will need a FASTA and/or PDB file for your input protein(s). The pipeline accepts a single or multiple input proteins. Each input protein will be used to perform independent BLAST and Foldseek searches. You can fetch FASTA and PDB files for most proteins in UniProt. The pipeline can fold proteins less than 400 amino acids in length using ESMFold **[4]**. You can generate PDB files for larger structures using tools such as ColabFold **[18]**.

After running this pipeline on a variety of proteins, we noticed that certain proteins resulted in more interpretable and useful maps than others. For example, shorter proteins with high structure quality tended to have the best performance. While we are

working to improve the pipeline to be effective at analyzing a broad diversity of proteins, we wanted to provide some guidance to users for what proteins might work well in this preliminary release. We therefore performed a meta-analysis to identify the limits of the pipeline.

Testing the pipeline limits by sampling from the most-studied human proteins

To identify the characteristics that make proteins appropriate for the current version of the pipeline, we analyzed a set of proteins sampled from the list of the 200 most-studied human proteins, as reported in **[25]**. We selected 25 proteins from a distribution of two features we noticed are important: protein length and structure quality, represented by average pLDDT **[30]** (Figure 7, Table 1). The pLDDT is a value that represents how a predicted structure will align with an experimental structure at each residue based on the distance difference, so by averaging this value across the length of the protein, we can determine how much of the protein lacks a defined structure. We used these 25 proteins as individual input proteins for 25 separate runs of the ProteinCartography pipeline.

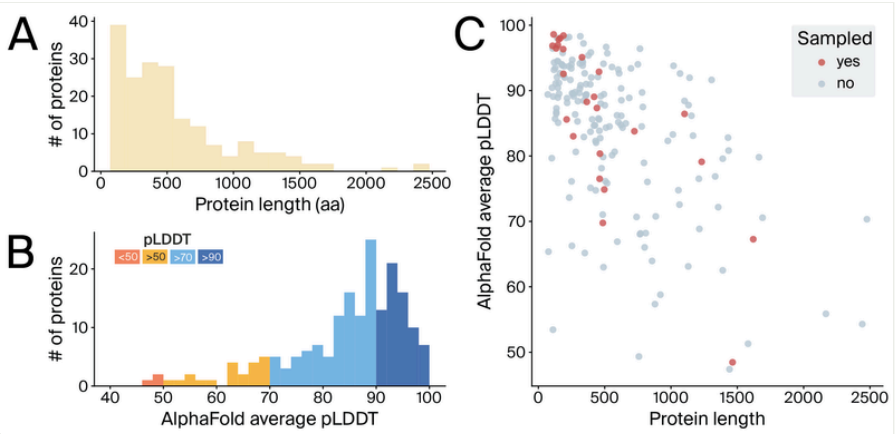

Figure 7

Characteristics of the 200 most-studied human proteins.

(A) Distribution of protein length across the 200 most-studied human proteins reported in Li & Buck (2021) that had structures available on UniProt.

(B) Distribution of the average pLDDT across the 200 most-studied human proteins from the same study. The average pLDDT was found by taking the average value of the per-residue pLDDT for each protein. The colors correspond to the confidence levels associated with pLDDT as represented by the key.

(C) Bivariate analysis of the length versus the average pLDDT for each protein. Based on this plot, we randomly sampled 25 of the 200 proteins. The proteins we sampled are represented by red dots.

UniProt ID	Protein name	Protein symbol (in figures)
Q9UM73	ALK tyrosine kinase receptor	ALK
Q96RI1	Bile acid receptor	NR1H4
P43235	Cathepsin K	CATK

UniProt ID	Protein name	Protein symbol (in figures)
P08603	Complement factor H	CFAH
P00374	Dihydrofolate reductase	DYR
P98170	E3 ubiquitin-protein ligase XIAP	XIAP
P49841	Glycogen synthase kinase-3 beta	GSK3B
P01112	GTPase HRas	RASH
P68871	Hemoglobin subunit beta	HBB
P04439	HLA class I histocompatibility antigen, A alpha chain	HLAA
P01834	Immunoglobulin kappa constant	IGKC
P14174	Macrophage migration inhibitory factor	MIF
P53779	Mitogen-activated protein kinase 10	MK10
Q15596	Nuclear receptor coactivator 2	NCOA2
Q99497	Parkinson disease protein 7	PARK7
P62937	Peptidyl-prolyl cis-trans isomerase A	PPIA
Q13451	Peptidyl-prolyl cis-trans isomerase FKBP5	FKBP5
P27986	Phosphatidylinositol 3-kinase regulatory subunit alpha	P85A
O75530	Polycomb protein EED	EED
P28074	Proteasome subunit beta type-5	PSB5
P19793	Retinoic acid receptor RXR-alpha	RXRA
P50120	Retinol-binding protein 2	RET2
P00441	Superoxide dismutase [Cu-Zn]	SODC
Q93009	Ubiquitin carboxyl-terminal hydrolase 7	UPB7
P40337	von Hippel-Lindau tumor suppressor	VHL

Table 1. The 25 proteins we sampled from the 200 most-studied human proteins.

SHOW ME THE DATA: You can find the metadata, BLAST and Foldseek hits, protein structures, and all the results from the pipeline for these 25 proteins on Zenodo (DOI: 10.5281/zenodo.8377393).

We evaluated the quality of the clusters generated using the cluster compactness metric discussed above (Figure 8, A). In addition to protein length and average pLDDT, we explored how metrics such as the number of domains, the fraction sequence identity to the input, and the TM-score to the input impacted compactness (Figure 8, B–F). For each of these metrics, we determined the value for every protein in each map and examined how their distributions varied in comparison to compactness (Figure 8). Among the 25-protein sample, shorter proteins with higher pLDDT and fewer domains tended to result in maps with more compact clusters than longer proteins with lower average pLDDT values and more domains (Figure 8, B–D). Proteins with lower pLDDT values could be lower-confidence predictions, but they could also be proteins with more intrinsic disorder. In either case, these proteins might not be appropriate for ProteinCartography analysis using structural comparisons. Additionally, AlphaFold structures are treated as rigid bodies and are aligned as such in the alignment step of our analysis. Slight differences due to protein dynamics could therefore be missed by alignment. This is something to consider especially when working with proteins that contain multiple domains connected by flexible regions (Figure 8, D).

The TM-scores shown here are calculated during the all-v-all structural comparison, which does not evaluate every possible protein-protein comparison comprehensively and therefore may contain missing values. For more information see the "Construct all-v-all similarity matrix" section. This means that zeros don't necessarily represent non-matches. They just represent less good matches relative to the rest of the proteins. To visualize the median without the presence of these zero-values, we supply a median TM-value for each analysis with (white circle) and without (white diamond) zeros (Figure 8, F).

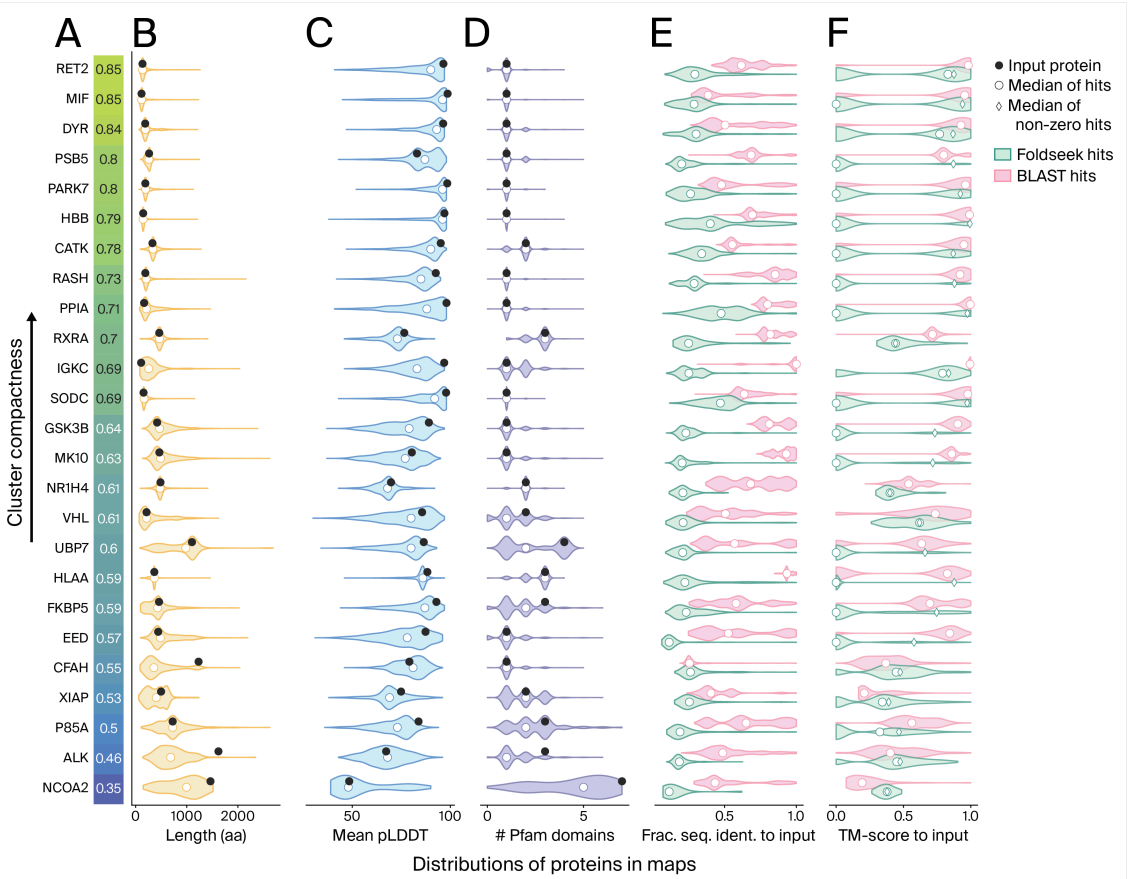

Figure 8

Protein characteristics affect the output of the ProteinCartography pipeline.

(A) Cluster compactness for the analyses conducted using each of the 25 sampled proteins as input proteins.

(B–D) Length (B), average pLDDT (C), and number of PFAM domains (D) of all the proteins identified and clustered in each analysis. The black dot represents the input protein and the white dot represents the median. Note that smaller proteins with higher pLDDT and fewer domains tend to have more compact clusters.

(E–F) Fraction sequence identity (Frac. seq. ident.) (E) and TM-score (F) compared to the input for each protein identified and clustered in each analysis. Green represents Foldseek hits identified based on structural similarity, and pink represents BLAST hits identified based on sequence similarity. White circles show the median for each, and the white diamonds show the median for each with the zeros removed.

There are also certain proteins that are not well represented in the AlphaFold database. For example, viral proteins have been excluded from the current draft of the AlphaFold database, meaning searches involving these proteins will be limited **[5][6]**. We wanted to know whether the taxonomic diversity or total number of hits in an analysis impacted pipeline performance. The number of hit proteins didn't correlate well with cluster compactness for our limited analysis (Figure 9, A, B, F). However, there are clear cases when having very few proteins resulted in poor analyses. For example, NCOA2 only had 93 proteins identified and had the lowest cluster compactness (Figure 9, D, F). We also observed that more diverse taxonomic distributions may result in more compact clusters, but need additional analyses to determine if this trend holds (Figure 9, C–F).

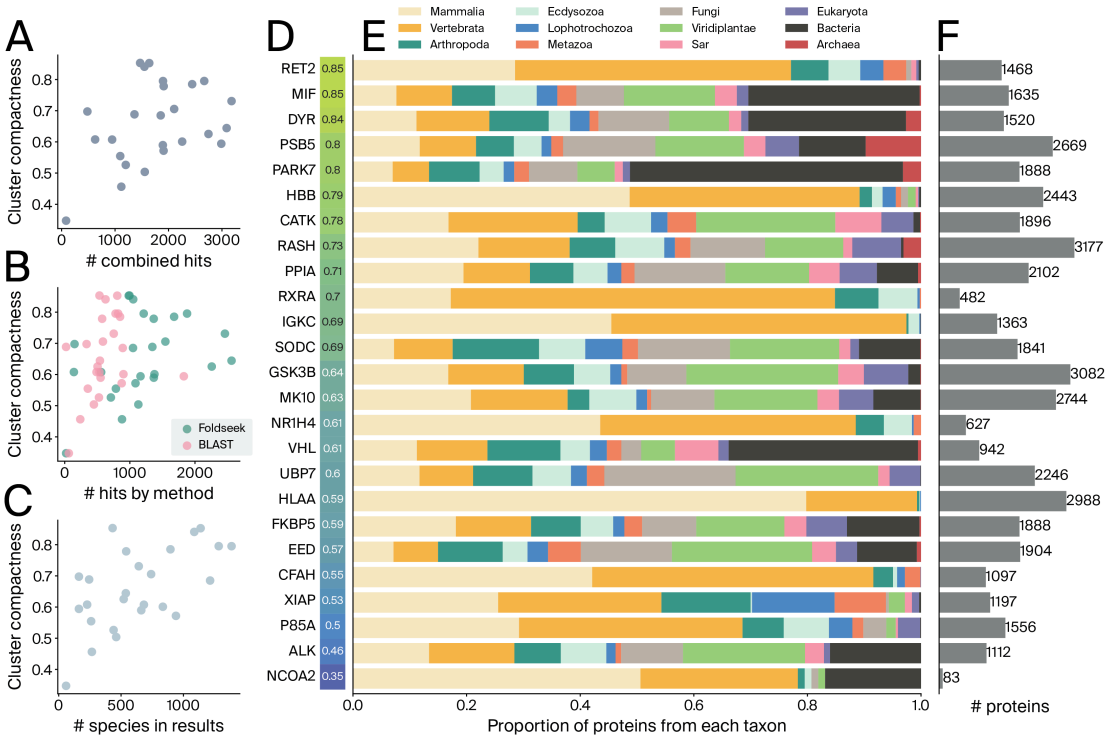

Figure 9

Protein family diversity may impact clustering.

(A) Bivariate plot showing the number of combined hits (Foldseek and BLAST) vs. cluster compactness for the 25 proteins sampled.

(B) Bivariate plot showing the number of hits, either Foldseek (green) or BLAST (pink) vs. cluster compactness for the 25 proteins we sampled.

(C) Bivariate analysis showing the number of species in each analysis result vs. the cluster compactness for each of the 25 analyses.

(D) Proteins are ordered by their cluster compactness, as represented by the colored bar — more compact clusters are at the top and less compact clusters at the bottom.

(E) For each of the 25 analyses, we sorted hits into a broad taxon, and graphed the proportion of the total number of hits in each broad taxon bucket as a bar chart.

(F) The total number of proteins represented in each analysis.

In summary, we advise you to consider the characteristics of each input protein when analyzing it with the ProteinCartography pipeline, as not all proteins are equally appropriate for structural comparison and could instead be evaluated using sequence, shapemer (or short fragment of a protein structure), or protein language model embedding comparisons. We will continue to use these 25 proteins throughout the following sections to evaluate our methods.

> **SHOW ME THE DATA**: You can find the metadata, BLAST and Foldseek hits, protein structures, and all the results from the pipeline for these 25 proteins on Zenodo (DOI: 10.5281/zenodo.8377393).

Protein searches

Once you've designated an input protein or proteins, the first step of the ProteinCartography pipeline involves searching protein structure and sequence databases.

Sequence-based search

The pipeline performs a sequence-based search for each input protein using NCBI Protein BLAST to search against the full NCBI non-redundant (nr) database **[19]**. It runs BLAST using a query to the web API. You can customize the number of hits returned with a default cutoff of 3,000. In general, there is no taxonomic constraint applied. Because we're querying such a large database and asking for relatively few hits, we also don't generally use an E-value cutoff for our BLAST search. In our runs so far, the median sequence identities for our BLAST hits have been consistently higher than for our Foldseek hits, and generally above 50% (Figure 8, E). However, including quality cutoffs in our BLAST search or including the ability for users to set a cutoff for this is something that we hope to include in future versions.

From the BLAST search, the pipeline retrieves a list of RefSeq or GenBank identifiers. To retrieve predicted structures from AlphaFold and retrieve protein metadata, it maps these identifiers to UniProt accessions **[17]**. At this step, some proteins are usually lost because not all proteins present in the non-redundant NCBI database are present in UniProt.

Structure-based search

For the structure search, the pipeline uses a Foldseek web API query to search against the AlphaFold/UniProt50 v4, AlphaFold/Swiss-Prot v4, and AlphaFold/Proteome v4 databases using the 3Di search mode with no taxonomic restraints **[1][5][6][16]**. The AlphaFold/UniProt50 v4 database uses MMseqs2 clustering at 50% sequence identity and returns a representative from each cluster that has the highest structure quality (average pLDDT) instead of all of the most closely related proteins **[16]**. The AlphaFold/Swiss-Prot v4 database contains UniProt proteins with high-quality annotations **[31]**. The AlphaFold/Proteome v4 database contains proteomes from a set of 48 model organisms and global health-related organisms **[6]**. The pipeline currently runs the Foldseek structure-based search step using a query to the web API, and each database search returns a maximum of 1,000 sequences. This is a constraint set by Foldseek.

Aggregate, filter, and download hits

Using the combined set of BLAST and Foldseek hits, the pipeline queries the UniProt database to retrieve metadata for each protein, including the protein name, gene name, organism, protein length, cross-references to annotation databases such as Pfam and InterPro, and other metadata (see example TSV) **[17][32][33]**. It then uses this metadata to filter and remove hits based on user-defined size cutoffs (if applied), whether the protein is marked as a fragment, and whether the protein is marked as inactive.

Next, the pipeline downloads the structure files (PDB files) for the proteins that are in the AlphaFold database, which includes only protein structures predicted using AlphaFold **[5][6]**. We again lose some proteins at this step that were identified via BLAST but don't have AlphaFolded structures, but one could use AlphaFold or ESMFold to fold these unfolded proteins **[4][5]**.

Protein structure comparisons

Construct all-v-all similarity matrix

Once all of the structures have been obtained and compiled in a single folder, the pipeline uses Foldseek to compare every protein structure to every other protein structure to create an all-v-all comparison for network analysis **[1][16][34]**. This works by first performing `foldseek search` on the user's machine, this time searching each hit against every other hit that was downloaded from AlphaFold **[1]**. From these alignments, the pipeline obtains E-values for each comparison. The pipeline uses the default E-value threshold of 0.001 set by Foldseek to determine which pairs of proteins to compare using TM-align. The pipeline then aligns any pairs of structures with E-values that satisfy the threshold using `foldseek aln2tmscore` to obtain a TM-score (template modeling score). A TM-score is a metric for the structural similarity of protein structures that ranges from 0–1 **[20]**. A TM-score of 1 means the compared structures are identical, while two protein structures with scores above 0.5 are usually similar and proteins with scores of 0.17 or lower are likely unrelated **[20]**. Any comparisons that did not satisfy the threshold E-value do not return a TM-score; we set these missing values to 0 for the purposes of clustering. In addition to the E-value threshold, `foldseek search` defaults to returning a maximum of 1,000 TM-scores for each protein analyzed, including the input. This means that not all comparisons will have calculated TM-scores. We also treat these missing values as 0 for the purposes of clustering.

This type of thresholding is common in network analyses to help slice the space into groups of differing depths **[35][36][37]**, but we have not yet determined the optimal thresholding parameters for our analyses. Consequently, in our visualizations, there is an inflated number of comparisons with a TM-score of 0 — users should treat these zeros as missing values, rather than a true absence of structural similarity. In future versions of the pipeline, we plan to explore these thresholds in a more principled way to determine what is appropriate for different types of analysis and will provide configuration parameters for users to tune the stringency of filtering. We will also provide complete TM-score calculations for input proteins for visualization purposes.

Next, the ProteinCartography pipeline arranges TM-scores into an all-v-all similarity matrix. Additionally, the pipeline adds the score of each hit protein compared to the input protein(s) to the output TSV file and uses it in the final visualization steps.

Cluster proteins

After generating an all-v-all similarity matrix, the pipeline groups proteins into clusters based on how similar they are to each other. Originally, we used Foldseek's TM-align greedy set clustering algorithm to generate structural clusters **[16]**. Foldseek's clustering algorithm utilizes Linclust and MMseqs2 **[16][38][39]**. Briefly, protein structures are represented as 3Di sequences **[1]**. Linclust extracts short sections of these sequences and uses them to sort the sequences into groups. The longest sequence in each group is identified as the representative. Foldseek's structural clustering uses the representatives in an initial structural alignment, the output of which feeds into MMseqs2 clustering. During this step, MMseqs2 clusters representatives based on TM-score. MMseqs clustering is a greedy set-cover algorithm, meaning that it chooses a single representative structure and adds it into a new or existing cluster and repeats this until all sequences belong to clusters **[38]**.

In our 25-protein analysis, we observed that many analyses only contained a small number of structural clusters as defined by Foldseek. 15 out of 25 maps contained fewer than five clusters (Figure 10). Moreover, these clusters didn't tend to be very compact. While Foldseek's clustering method may be appropriate for some uses, we wondered whether other clustering algorithms may be better suited for our intended use case.

In particular, we were interested in Leiden clustering, a method that has become popular for identifying groupings within single-cell expression networks **[21][40][41]**. Leiden clustering is another algorithm used to identify communities, or preliminary clusters, within a pre-existing network in a three-phase process. First, proteins are grouped to find the highest quality community separations. Next, the algorithm undergoes a refinement step where proteins can be switched to other communities. Finally, the network is aggregated **[21]**. This process is generally done several times. We used the implementation of Leiden clustering found in the popular single-cell RNA-seq analysis package Scanpy **[41]**. This implementation takes a matrix of counts and performs principal component analysis (PCA), generating a neighborhood graph using an implementation of uniform manifold approximation projection (UMAP). We used parameters n_pcs = 30 and n_neighbors = 10 for this implementation of Leiden clustering, and performed Leiden clustering until optimized (scanpy default, n_iterations = –1).

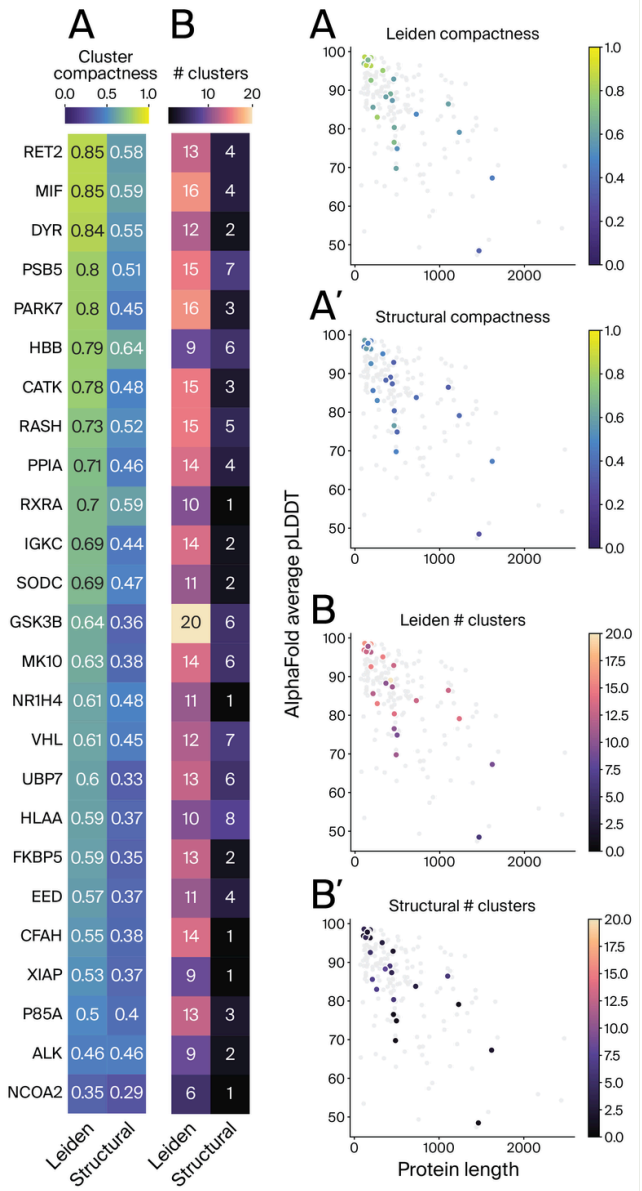

Figure 10

Leiden clustering produces more numerous and compact clusters than Foldseek TM-align greedy set clustering.

(A) Cluster compactness for Leiden clusters (left) and structural clusters calculated using the Foldseek clustering algorithm (right) for each of the 25 analyses completed using the randomly sampled proteins from the 200 most-studied human proteins.

(B) Number of clusters for Leiden clusters (left) and structural clusters calculated using the Foldseek clustering algorithm (right) for each of the 25 analyses.

(C–D′) Bivariate analysis of protein length vs average pLDDT for the full 200 most-studied proteins, with the sampled proteins represented by colored dots and the non-sampled proteins shown in light gray. Dots are colored based on cluster compactness of Leiden clusters (C), cluster compactness of structural clusters (C′), number of Leiden clusters (D), and number of structural clusters (D′).

To compare the clustering methods, we applied each to the 25 proteins we sampled in Figures 7–9 (Figure 10). To determine which algorithm better sorted the proteins into clusters, we measured cluster compactness and number of clusters in the resulting map (Figure 10). In all 25 cases, the structural similarity scores within clusters were higher when using Leiden clustering (Figure 10, A, C, C′) and there were more clusters (Figure 10, B, D, D′), suggesting that Leiden clustering might be more appropriate for identifying sub-groups within groups of structurally similar proteins. We hypothesize that Foldseek's structural clustering might be better for larger-scale analyses looking across families, but did not test this.

The current version of the pipeline provides both structural clustering using Foldseek's algorithm and Leiden clustering results in the final "aggregated_features.tsv" file, but it defaults to Leiden clustering in plots and analyses. We have not fully optimized the standard parameters and this could contribute to the differences in clustering quality for different protein families. In future iterations of the pipeline, we hope to experiment more with these parameters to develop more generalizable clustering approaches.

Cluster analysis

Calculate cross-cluster similarity matrix

To allow for better understanding of the quality and content of the clusters, the pipeline calculates a cross-cluster similarity matrix (Figure 2, Figure 11). For each cluster, it calculates the mean TM-score of all structures in that cluster versus all other proteins in each other cluster. Clusters with a greater mean cross-cluster TM-score are more structurally similar (with a maximum value of 1). Within this visualization, the y-axis represents the protein that is treated as the query protein, while the x-axis represents the target protein in each comparison. The diagonal of the matrix represents the similarity of all proteins within a cluster, which we can use to assess cluster compactness (cmx), or the average value of the diagonal. Clusters with a low within-cluster mean TM-score are likely to contain assortments of unrelated or dissimilar proteins. The pipeline visualizes the results of this comparison using a heatmap.

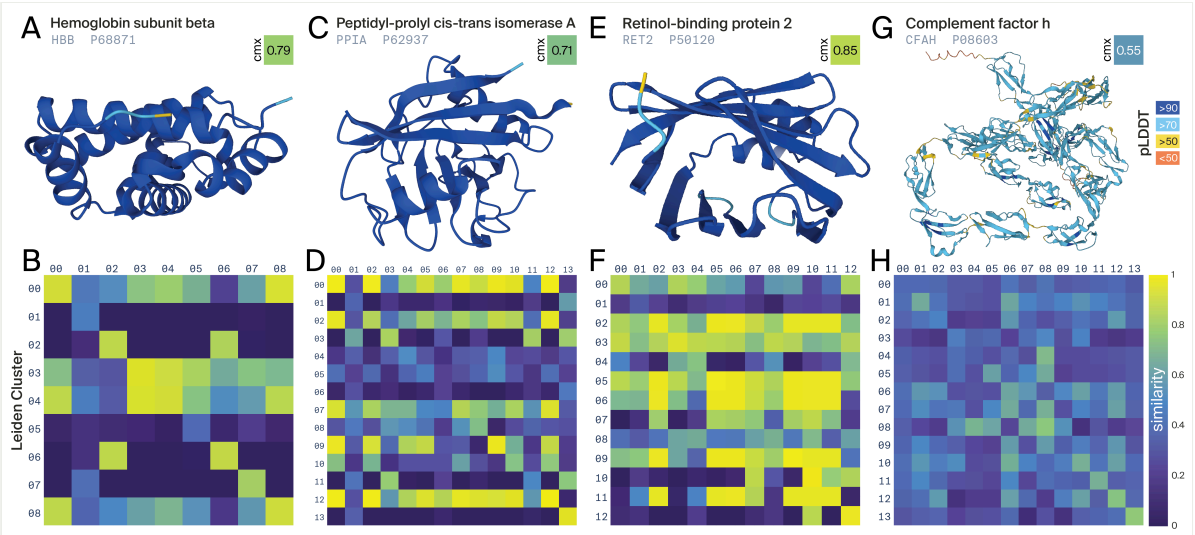

Figure 11

Cross-cluster similarity matrices allow for evaluation of clustering effectiveness.

(A, C, E, G) Structures of Hemoglobin subunit beta (A), Peptidyl-prolyl cis-trans isomerase A (C), Retinol-binding protein 2 (E), and Complement factor H (G) structures. In all cases the color of the structure responds to the per-residue pLDDT value represented in the key on the right. The cmx value is the “cluster compactness” score for each analysis. The number under each protein name is the corresponding accession number.

(B, D, F, H) Similarity matrices for Leiden clusters produced from the analyses for Hemoglobin subunit beta (B), Peptidyl-prolyl cis-trans isomerase A (D), Retinol-binding protein 2 (F), and Complement factor H (H).

While the cluster compactness score provides a general idea of the quality of the clustering, the heatmaps produced with each run of the pipeline can help users more thoroughly evaluate clustering effectiveness for their analyses. We show heatmaps from analysis of four example proteins, hemoglobin subunit beta (HBB), peptidyl-prolyl cis-trans isomerase A (PPIA), retinol-binding protein 2 (RET2), and complement factor h (CFAH) from our sample of most-studied human proteins (Figure 11). These examples reflect a range of potential outcomes for clustering. The first three proteins all have high cluster compactness (HBB: 0.79, PARK7: 0.71, RET2: 0.85), but their heatmaps reflect variable levels of interpretability. Clusters in the HBB analysis show a high level of compactness, with clusters 00, 02, 03, 04, 06, 07, and 08 showing mean TM-scores

> 0.5 along the diagonal (Figure 11, A–B). However, examining the other cells of the matrix reveals that several of these clusters may not be very distinct — for example, clusters 00, 03, 04, and 08 show high levels of mutual similarity, suggesting that these clusters might be combined into a single larger cluster (Figure 11, A–B).

The PPIA analysis shows a similar result, where several clusters could potentially be fractions of a larger cluster (Figure 11, C–D). Particularly interesting with the PPIA matrix, there are strong horizontal lines that suggest there are some clusters that are similar to all other clusters (Figure 11, C–D). The RET2 analysis shows a more extreme example: most clusters, with the exception of 01 and 12 (and, to a lesser extent, 04 and 10) show relatively strong similarity to all other clusters, suggesting that the clustering analysis was not able to identify distinguishable sub-groupings (Figure 11, E–F). In some cases, such as for large proteins like CFAH, clustering does not appear to produce compact or distinct clusters (Figure 11, G–H). For these types of proteins, the large number of domains and lower overall pLDDT might impede structural comparisons due to the limitations of rigid-body structural comparison, and other types of comparison networks — such as sequence, shapemer, or protein language model embedding — might be more amenable to clustering analysis.

Perform semantic analysis

The pipeline retrieves protein metadata from UniProt, which can include gene and protein names **[17]**. While not always reliable, especially for understudied organisms, these annotations can provide a more human-readable method of understanding what kinds of proteins exist in each cluster **[42]**. To summarize protein annotations from each cluster, we implemented a visualization that we refer to as “simple semantic analysis.” For each Leiden cluster, the pipeline aggregates the most frequently occurring annotations and individual annotation words, and represents these as a ranked bar chart and proportional word cloud, respectively (Figure 5).

For example, the semantic analysis plot for the MAPK10 analysis is shown in Figure 5. Our input protein is in LC02, where the most represented annotation is “Stress-activated protein kinase JNK (EC 2.7.11.24)”. In the word cloud, we can also see,“JNK,” “Mitogen-activated,” and “MAPK,” all suggesting that clustering analysis correctly aggregated these proteins together, since we know that MAPK10 is a member of the JNK family. LC11, a cluster composed of primarily plant proteins, has the top annotation “Mitogen-activated protein kinase (EC 2.7.11.24),” suggesting it comes from the same

enzyme class, EC 2.7.11.24. For LC12, a primarily fungal cluster, the top annotation is “Mitogen-activated protein kinase HOG1 (MAP kinase HOG1) (EC 2.7.11.24),” which is consistent with the literature showing that the JNK pathway is similar to the fungal HOG pathway **[43]**. This analysis can provide broader biological context for the contents of each cluster.

Visualization

Create maps

The pipeline uses standard dimensionality reduction approaches to create a visual representation of protein space. It starts by using the original all-v-all similarity matrix to calculate a principal component analysis (PCA) with 30 components **[44]**. The PCA results are then passed to an analysis to calculate the t-distributed stochastic neighbor embedding (t-SNE) and the uniform manifold approximation projection (UMAP) **[22][23]**. For the t-SNE, it returns two components, the perplexity is set to 50, and the number of iterations to run is set to 2,000. For the UMAP, it returns two components, the number of neighbors is set to 80, and the minimum distance between neighbors is set to 0.5. The parameters used here are defaults used in other analogous analyses, but in the future, we plan to optimize them for our particular use cases.

Both UMAP and t-SNE are non-linear, graph-based methods for dimensionality reduction **[22][23]**. They are both meant for visualization — we do not treat the 2D maps generated by these techniques as fully representative of the higher-dimensional relationships between proteins. They each follow the same general principle: create a high-dimensional graph, then reconstruct it in a lower-dimensional space while retaining the structure. t-SNE moves the graph from high dimension to lower dimension point by point, while UMAP compresses the high-dimensional graph **[22] [23]**. We provide both in this pipeline so that the user can choose which visualization is easier to navigate for each protein family. Often, t-SNE creates more space between clusters, while the UMAP plot appears more connected. However, users should not interpret distances in the 2D axis of UMAP or t-SNE plots as quantitative.

Generate interactive plots

Finally, the pipeline uses all the data collected above to create an interactive and navigable map that you can use to explore the protein family (Figure 2, Figure 3). The pipeline produces HTML file maps that allow dynamic visualization, built using the Plotly Python package **[24]** This allows you to interact with graphs and apply multiple overlays as shown in the above examples and detailed more thoroughly below. A toggle button allows you to see the input protein(s) in the map as black, four-pointed star markers, and metadata for each protein is displayed in a tooltip when the mouse cursor hovers over a point. In addition to the interactive plot, the pipeline produces a file that contains all the UniProt features along with the information calculated throughout the pipeline for each protein.

Plot overlays

To empower researchers and make these plots maximally useful, the pipeline has a color drop-down that colors points according to protein metadata. The default view colors each point by its Leiden cluster. Clustering (separating the protein structures into similar groups based on the all-v-all similarity matrix) and mapping (visualization via dimensionality reduction) take place independently in the pipeline. However, at this final step, these two representations of protein space are combined when the results of the clustering analysis are overlaid onto the map.

You can color points by the following:

- Leiden cluster
- Annotation score, a metric to measure the annotation content of a UniProt protein **[17]**
- Broad taxon, which can be either eukaryotic- or bacterial-focused with this current version of the pipeline; see the GitHub README for how to customize the taxonomic groups
- Length
- Source (Foldseek vs. BLAST)
- TM-score vs. the input protein (this TM-score is the value from the Foldseek local search and may not reflect the true TM-score due to E-value thresholding)
- Fraction sequence identity vs. the input protein

- Average pLDDT of the protein
- [Experimental] Concordance vs. the input protein (see below)

Using this visualization and the accompanying file containing this information in a tabular format, we can begin to make predictions and hypotheses about the relationship of proteins to each other and even how these proteins might function.

Additional methods

We used ChatGPT to write, clean up, and comment code. We also used it to suggest wording ideas that we edited extensively.

Next steps

This pipeline is a work in progress — we are actively building and adding features. Many of the features we hope to improve and add are recorded as GitHub issues in the ProteinCartography GitHub repository. As we move forward, we hope to build in four important areas: broad software improvements, validation, new analysis features, and linkages with other software packages. We lay out our plans below, but would love feedback on what you'd like to see us tackle next.

Broad software improvements

We built the ProteinCartography pipeline using Snakemake, which lets us develop flexible workflows that can run on most computers. However, we plan to provide a Nextflow version of the pipeline in the future. Additionally, we plan to decrease our reliance on APIs in general, but in particular, we hope to avoid using the Foldseek API. This will also allow us to support different databases for comparisons. It's really important to us that this pipeline is not just useful, but also usable, so we plan to work on increasing usability and adding features that allow researchers to more easily interpret the space.

Validation

To provide users with more definitive and useful information, we must first provide more validation and principled statistical analysis. We plan to develop a clearer understanding of what characteristics make proteins amenable to analysis using the pipeline by performing additional large-scale analyses of diverse proteins and performing statistical tests to understand how well different protein features correlate with pipeline results. To perform such validation, we also need to expand our metrics for evaluating pipeline performance from focusing on cluster compactness to also include cluster distinctness, evaluation of over- and under-clustering, annotation distinctness (how well annotations line up with clusters), and other measures. By assessing these metrics, we will also be able to develop automated methods for parameter selection to identify sensible defaults that work well across protein families of diverse size and composition.

Additionally, we plan to explore the current parameters of the pipeline and how we might be able to optimize them for different use cases.

Finally, we are currently working on in-lab biochemical validation to show that the interesting predictions and hypotheses we have been able to make about protein function based on the results of the pipeline are actually indicative of true functional differences. We're looking for proteins with established purification protocols and assays that come from diverse and interesting protein families. If you have any suggestions for proteins that fit these criteria and would be especially useful for this type of biochemical validation, please let us know in a comment!

New analyses and features

While our pipeline is able to aggregate results from sequence and structural searches and provide maps for exploration, the pipeline does not yet perform detailed analysis of the features within proteins that make them distinct from each other. We'd like to add analyses that allow us to identify the specific regions of proteins that result in differences between clusters.

We want to make overall exploration of the maps easier and more intuitive by employing analyses to identify interesting proteins or outliers that might be of particular interest. We also want to find easy ways of pointing out which areas of the

maps have high-quality clusters and which areas of the map users should consider avoiding. We'd like to find ways to identify the specific structural features between clusters that make them unique.

For example, we are interested in identifying proteins that are convergently evolved — composed of divergent sequences but which fold into similar structures — as well as proteins with high sequence identity but apparently divergent structures. The current version of the pipeline provides a rudimentary measure of the relationship between sequence identity and structural similarity, which we call "concordance." This is a simple measure that subtracts structural similarity from sequence identity. More positive values mark proteins with greater structural similarity than sequence identity, while negative values mark the opposite. While the current measure is not statistically principled, as TM-score and sequence identity do not follow the same linear scale, we are exploring methods to compare sequence identity to structural similarity to identify proteins that meaningfully diverge or converge from expectations.

> **FEATURE REQUESTS:** Are there features that would be useful to you? Let us know in a comment on this pub!

Software linkages

We'd like to incorporate phylogenetic information and sequence information to complement the structural information that the pipeline provides. Additionally, we'd like to integrate this pipeline with other resources from Arcadia, including NovelTree and PreHGT **[45][46]**.

> **TRY IT**: The ProteinCartography pipeline is available here, along with instructions to get started.

If you use it, let us know in a comment on this pub! We'd love to hear your use case and what you learned from your own protein mapping. Additionally, we welcome outside suggestions as GitHub issues and contributions through pull requests.

What do you think?

We'd particularly interested in getting your feedback on the following:

- Would this kind of pipeline be useful for your own work?
- How could we make it more useful for you?
- Do you have any recommendations for types of analyses or validation?
- Do you have any recommendations for protein families that you think would be particularly interesting to look at?

References

1. van Kempen M, Kim SS, Tumescheit C, Mirdita M, Lee J, Gilchrist CLM, Söding J, Steinegger M. (2023). Fast and accurate protein structure search with Foldseek. https://doi.org/10.1038/s41587-023-01773-0

2. Sykes J, Holland BR, Charleston MA. (2022). A review of visualisations of protein fold networks and their relationship with sequence and function. https://doi.org/10.1111/brv.12905

3. Illergård K, Ardell DH, Elofsson A. (2009). Structure is three to ten times more conserved than sequence—A study of structural response in protein cores. https://doi.org/10.1002/prot.22458

4. Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, Smetanin N, Verkuil R, Kabeli O, Shmueli Y, dos Santos Costa A, Fazel-Zarandi M, Sercu T, Candido S, Rives A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. https://doi.org/10.1126/science.ade2574

5. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, Bridgland A, Meyer C, Kohl SAA, Ballard AJ, Cowie A, Romera-Paredes B, Nikolov S, Jain R, Adler J, Back T, Petersen S, Reiman D, Clancy E, Zielinski M, Steinegger M, Pacholska M, Berghammer T, Bodenstein S, Silver D, Vinyals O, Senior AW, Kavukcuoglu K,

Kohli P, Hassabis D. (2021). Highly accurate protein structure prediction with AlphaFold. https://doi.org/10.1038/s41586-021-03819-2

6 Varadi M, Anyango S, Deshpande M, Nair S, Natassia C, Yordanova G, Yuan D, Stroe O, Wood G, Laydon A, Žídek A, Green T, Tunyasuvunakool K, Petersen S, Jumper J, Clancy E, Green R, Vora A, Lutfi M, Figurnov M, Cowie A, Hobbs N, Kohli P, Kleywegt G, Birney E, Hassabis D, Velankar S. (2021). AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. https://doi.org/10.1093/nar/gkab1061

7 Ben-Tal N, Kolodny R. (2014). Representation of the Protein Universe using Classifications, Maps, and Networks. https://doi.org/10.1002/ijch.201400001

8 Knudsen M, Wiuf C. (2010). The CATH database. https://doi.org/10.1186/1479-7364-4-3-207

9 Fox NK, Brenner SE, Chandonia J-M. (2013). SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. https://doi.org/10.1093/nar/gkt1240

10 Chandonia J-M, Guan L, Lin S, Yu C, Fox NK, Brenner SE. (2021). SCOPe: improvements to the structural classification of proteins – extended database to facilitate variant interpretation and machine learning. https://doi.org/10.1093/nar/gkab1054

11 Hou J, Jun S-R, Zhang C, Kim S-H. (2005). Global mapping of the protein structure space and application in structure-based inference of protein function. https://doi.org/10.1073/pnas.0409772102

12 Hou J, Sims GE, Zhang C, Kim S-H. (2003). A global representation of the protein fold space. https://doi.org/10.1073/pnas.2628030100

13 Choi I-G, Kim S-H. (2006). Evolution of protein structural classes and protein sequence families. https://doi.org/10.1073/pnas.0606239103

14 Levitt M. (2009). Nature of the protein universe. https://doi.org/10.1073/pnas.0905029106

15 Osadchy M, Kolodny R. (2011). Maps of protein structure space reveal a fundamental relationship between protein structure and function. https://doi.org/10.1073/pnas.1102727108

16 Barrio-Hernandez I, Yeo J, Jänes J, Wein T, Varadi M, Velankar S, Beltrao P, Steinegger M. (2023). Clustering predicted structures at the scale of the known protein universe. https://doi.org/10.1101/2023.03.09.531927

17 Consortium TU. (n.d.). UniProt: the universal protein knowledgebase in 2021. https://doi.org/10.1093/nar/gkaa1100

18 Mirdita M, Schütze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M. (2022). ColabFold: making protein folding accessible to all. https://doi.org/10.1038/s41592-022-01488-1

19 Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. (1990). Basic local alignment search tool. https://doi.org/10.1016/s0022-2836(05)80360-2

20 Zhang Y, Skolnick J. (2004). Scoring function for automated assessment of protein structure template quality. https://doi.org/10.1002/prot.20264

21 Traag VA, Waltman L, van Eck NJ. (2019). From Louvain to Leiden: guaranteeing well-connected communities. https://doi.org/10.1038/s41598-019-41695-z

22 Belkina AC, Ciccolella CO, Anno R, Halpert R, Spidlen J, Snyder-Cappione JE. (2019). Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. https://doi.org/10.1038/s41467-019-13055-y

23 McInnes L, Healy J, Saul N, Großberger L. (2018). UMAP: Uniform Manifold Approximation and Projection. https://doi.org/10.21105/joss.00861

24 Plotly Technologies Inc. (2015). Collaborative data science. https://plot.ly

25 Li Z, Buck M. (2021). Beyond history and “on a roll”: The list of the most well-studied human protein structures and overall trends in the protein data bank. https://doi.org/10.1002/pro.4038

26 Morrison DK. (2012). MAP Kinase Pathways. https://doi.org/10.1101/cshperspect.a011254

27 Neidhart S, Antonsson B, Gilliéron C, Vilbois F, Grenningloh G, Arkinstall S. (2001). c-Jun N-terminal kinase-3 (JNK3)/stress-activated protein kinase-β (SAPKβ) binds and phosphorylates the neuronal microtubule regulator SCG10. https://doi.org/10.1016/s0014-5793(01)03090-3

28 Li M, Liu J, Zhang C. (2011). Evolutionary History of the Vertebrate Mitogen Activated Protein Kinases Family. https://doi.org/10.1371/journal.pone.0026999

29 Gustin MC, Albertyn J, Alexander M, Davenport K. (1998). MAP Kinase Pathways in the Yeast Saccharomyces cerevisiae. https://doi.org/10.1128/mmbr.62.4.1264-1300.1998

30 Mariani V, Biasini M, Barbato A, Schwede T. (2013). lDDT: a local superposition-free score for comparing protein structures and models using distance

difference tests. https://doi.org/10.1093/bioinformatics/btt473

31 Poux S, Arighi CN, Magrane M, Bateman A, Wei C-H, Lu Z, Boutet E, Bye-A-Jee H, Famiglietti ML, Roechert B, UniProt Consortium T. (2017). On expert curation and scalability: UniProtKB/Swiss-Prot as a case study. https://doi.org/10.1093/bioinformatics/btx439

32 Blum M, Chang H-Y, Chuguransky S, Grego T, Kandasaamy S, Mitchell A, Nuka G, Paysan-Lafosse T, Qureshi M, Raj S, Richardson L, Salazar GA, Williams L, Bork P, Bridge A, Gough J, Haft DH, Letunic I, Marchler-Bauer A, Mi H, Natale DA, Necci M, Orengo CA, Pandurangan AP, Rivoire C, Sigrist CJA, Sillitoe I, Thanki N, Thomas PD, Tosatto SCE, Wu CH, Bateman A, Finn RD. (2020). The InterPro protein families and domains database: 20 years on. https://doi.org/10.1093/nar/gkaa977

33 Mistry J, Chuguransky S, Williams L, Qureshi M, Salazar GA, Sonnhammer ELL, Tosatto SCE, Paladin L, Raj S, Richardson LJ, Finn RD, Bateman A. (2020). Pfam: The protein families database in 2021. https://doi.org/10.1093/nar/gkaa913

34 Zhang Y. (2005). TM-align: a protein structure alignment algorithm based on the TM-score. https://doi.org/10.1093/nar/gki524

35 Atkinson HJ, Morris JH, Ferrin TE, Babbitt PC. (2009). Using Sequence Similarity Networks for Visualization of Relationships Across Diverse Protein Superfamilies. https://doi.org/10.1371/journal.pone.0004345

36 Copp JN, Akiva E, Babbitt PC, Tokuriki N. (2018). Revealing Unexplored Sequence-Function Space Using Sequence Similarity Networks. https://doi.org/10.1021/acs.biochem.8b00473

37 Otasek D, Morris JH, Bouças J, Pico AR, Demchak B. (2019). Cytoscape Automation: empowering workflow-based network analysis. https://doi.org/10.1186/s13059-019-1758-4

38 Hauser M, Steinegger M, Söding J. (2016). MMseqs software suite for fast and deep clustering and searching of large protein sequence sets. https://doi.org/10.1093/bioinformatics/btw006

39 Steinegger M, Söding J. (2018). Clustering huge protein sequence sets in linear time. https://doi.org/10.1038/s41467-018-04964-5

40 Heumos L, Schaar AC, Lance C, Litinetskaya A, Drost F, Zappia L, Lücken MD, Strobl DC, Henao J, Curion F, Aliee H, Ansari M, Badia-i-Mompel P, Büttner M, Dann E, Dimitrov D, Dony L, Frishberg A, He D, Hediyeh-zadeh S, Hetzel L, Ibarra IL, Jones MG, Lotfollahi M, Martens LD, Müller CL, Nitzan M, Ostner J, Palla G, Patro R, Piran Z, Ramírez-Suástegui C, Saez-Rodriguez J, Sarkar H, Schubert B,

Sikkema L, Srivastava A, Tanevski J, Virshup I, Weiler P, Schiller HB, Theis FJ. (2023). Best practices for single-cell analysis across modalities. https://doi.org/10.1038/s41576-023-00586-w

41 Wolf FA, Angerer P, Theis FJ. (2018). SCANPY: large-scale single-cell gene expression data analysis. https://doi.org/10.1186/s13059-017-1382-0

42 de Crécy-lagard V, Amorin de Hegedus R, Arighi C, Babor J, Bateman A, Blaby I, Blaby-Haas C, Bridge AJ, Burley SK, Cleveland S, Colwell LJ, Conesa A, Dallago C, Danchin A, de Waard A, Deutschbauer A, Dias R, Ding Y, Fang G, Friedberg I, Gerlt J, Goldford J, Gorelik M, Gyori BM, Henry C, Hutinet G, Jaroch M, Karp PD, Kondratova L, Lu Z, Marchler-Bauer A, Martin M-J, McWhite C, Moghe GD, Monaghan P, Morgat A, Mungall CJ, Natale DA, Nelson WC, O'Donoghue S, Orengo C, O'Toole KH, Radivojac P, Reed C, Roberts RJ, Rodionov D, Rodionova IA, Rudolf JD, Saleh L, Sheynkman G, Thibaud-Nissen F, Thomas PD, Uetz P, Vallenet D, Carter EW, Weigele PR, Wood V, Wood-Charlson EM, Xu J. (2022). A roadmap for the functional annotation of protein families: a community perspective. https://doi.org/10.1093/database/baac062

43 Wold S, Esbensen K, Geladi P. (1987). Principal component analysis. https://doi.org/10.1016/0169-7439(87)80084-9

44 Dutton RJ, Reiter T. (2024). PreHGT: A scalable workflow that screens for horizontal gene transfer within and between kingdoms. https://doi.org/10.57844/ARCADIA-JFBP-7P11

45 Celebi FM, Chou S, McGeever E, Patton AH, York R. (2024). NovelTree: Highly parallelized phylogenomic inference. https://doi.org/10.57844/ARCADIA-Z08X-V798