Chapter 12

Navigating Among Known Structures in Protein Space

Aya Narunsky, Nir Ben-Tal, and Rachel Kolodny

Abstract

Present-day protein space is the result of 3.7 billion years of evolution, constrained by the underlying physicochemical qualities of the proteins. It is difficult to differentiate between evolutionary traces and effects of physicochemical constraints. Nonetheless, as a rule of thumb, instances of structural reuse, or focusing on structural similarity, are likely attributable to physicochemical constraints, whereas sequence reuse, or focusing on sequence similarity, may be more indicative of evolutionary relationships. Both types of relationships have been studied and can provide meaningful insights to protein biophysics and evolution, which in turn can lead to better algorithms for protein search, annotation, and maybe even design.

In broad strokes, studies of protein space vary in the entities they represent, the similarity measure comparing these entities, and the representation used. The entities can be, for example, protein chains, domains, supra-domains, or smaller protein sub-parts denoted themes. The measures of similarity between the entities can be based on sequence, structure, function, or any combination of these. The representation can be global, encompassing the whole space, or local, focusing on a particular region surrounding protein(s) of interest. Global representations include lists of grouped proteins, protein networks, and maps. Networks are the abstraction that is derived most directly from the similarity data: each node is the protein entity (e.g., a domain), and edges connect similar domains. Selecting the entities, the similarity measure, and the abstraction are three intertwined decisions: the similarity measures allow us to identify the entities, and the selection of entities influences what is a meaningful similarity measure. Similarly, we seek entities that are related to each other in a way, for which a simple representation describes their relationships succinctly and accurately. This chapter will cover studies that rely on different entities, similarity measures, and a range of representations to better understand protein structure space. Scholars may use publicly available navigators offering a global representation, and in particular the hierarchical classifications SCOP, CATH, and ECOD, or a local representation, which encompass structural alignment algorithms. Alternatively, scholars can configure their own navigator using existing tools. To demonstrate this DIY (do it yourself) approach for navigating in protein space, we investigate substrate-binding proteins. By presenting sequence similarities among this large and diverse protein family as a network, we can infer that one member (pdb ID 4ntl; of yet unknown function) may bind methionine and suggest a putative binding mechanism.

Key words Protein space navigation, Structure space, Evolutionary relationships in protein space

Tobias Sikosek (ed.), *Computational Methods in Protein Evolution*, Methods in Molecular Biology, vol. 1851,
https://doi.org/10.1007/978-1-4939-8736-8_12,

1 Introduction

1.1 Protein Structure Space

Protein structure space is an abstract model which we use when we study large, representative, sets of protein structures and their interrelationships. Inspecting these large datasets allows us to better understand protein evolution and biophysics. While protein space is not real, the entities that populate it are: for example, these can be protein chains or domains; furthermore, their comparisons are meaningful. Thus, the first and essential step when studying protein structure space is to decide on the set of entities and the measure of similarity among them (coupled with a method to compute it). We can then calculate all-against-all comparisons of these entities to construct the initial dataset. Because the abstract model is derived from these comparisons, it is essential that this initial set is as accurate and comprehensive as possible. Navigating in protein structure space is in many ways navigating within this initial dataset, and we can do this either locally or globally.

1.2 Navigation Modes

Navigating "locally" or "globally" in protein structure space is a metaphor, which describes how we study the dataset. By "local," we mean that we identify small sets of comparisons, which we deem relevant. Given a query protein chain, or query protein domain, we think of the comparisons of that protein and its near structural neighbors (i.e., other proteins in the dataset that are similar to it) as covering its local region in structure space. Navigating locally is moving between (overlapping) local regions, akin to moving between landmarks when using a navigation app. By "global," we mean that we derived a model which integrates information from many (possibly all) comparisons and explore this model. Alternatively, we can think of this model as a data structure that organizes all entities based on the relationships between them. Navigating globally means that we either explore the properties of this data structure, akin to staring at a map, or move between proteins based on their location in the data structure.

1.3 The Potential of Studying Protein Structure Space

Studying protein structure space can help us better understand protein evolution and biophysics. It may also have a practical value: insights could be used in protein structure prediction, protein function prediction, and protein design. By way of motivation, we list a few examples; there are many more (e.g., those listed in [1, 2].) Evolution scholars have navigated protein space looking for clues in the remnants of evolutionary processes [3, 4]. For example, Choi et al. [5] derive the "multiple birth model" for proteins from maps, Dokholyan et al. [6] offered support for all proteins evolving from a few precursors, Alva et al. [7] studied the relationship between convergent and divergent evolution, Farias-Rico et al. traced the evolutionary relationships between ancient superfolds

[8], and Nepomnyachiy et al. [9] highlighted the complex nature of reuse patterns, which often overlap with each other. Studying protein structure space also revealed biophysical properties of proteins: examples include the work of Skolnick et al. [10], Nepomnyachiy et al. [11], and Mackenzie et al. [12]. Understanding the space of all structures can help in protein structure prediction and in better organizing the databases for structure search [13]. A global perspective also offered a hint to the relationship between protein structure and function, showing that there is a localized region of high function diversity [14]. Notice that one size does not fit all: different insights were gained from representations of protein space that varied in the sets of entities curated and in the way the entities were compared to each other.

2 Materials and Methods

2.1 The Entities

The entities are derived from the proteins of known structure in the Protein Data Bank (PDB) [15] and can be parts of proteins of different scales, depending on the question at hand. With minimal processing, these can be protein complexes or protein chains. One could also consider protein domains [16, 17] (or even supra-domains [18]), or meaningful sub-domain entities: protein fragments (e.g., [19, 20]), protein themes [9], protein interfaces [21], protein-peptide complexes [22], repetitive secondary structure elements (e.g., Smotifs [23]), or tertiary structural motifs (TERMS) [12]. Alternatively, the structures could possibly be predictions [24], or homology models [25]. Typically, one would use datasets that were curated by others (e.g., the domain sets in SCOP [26], CATH [27], or ECOD [28]). It is important to consider if the entities are mutually exclusive, or not. For example, domains are mutually exclusive because when partitioning chains to domains, each residue is associated with only a single domain; in contrast, themes cover multiple (nested) segments in a protein chain.

2.2 Relating the Entities

Comparing proteins can be based on their sequences, structures, or functions. The most straightforward measure is *sequence* similarity, which suggests shared evolutionary ancestor(s) [29]. Sequence alignment tools vary in sensitivity: less sensitive methods rely directly on the protein sequences (e.g., BLAST). More sensitive methods rely on an enriched version of the sequences: either sequence profiles (e.g., PSI-BLAST) or HMMs (e.g., HHSearch [30] or HHMER [31]); these are probabilistic models that include not only the protein sequence but also sequences of its close homologues [30, 31]. Using sensitive sequence aligners like HHSearch or HHMER reveals more distant evolutionary relationships. To avoid relating pairs of proteins that have diverged beyond what we would consider similar, scholars add an additional restriction

that the structures of the aligned residues be similar [11, 32]; it is not impossible that structural changes emerged upon evolution though (and anyway, proteins often undergo conformational changes [33, 34]). Note that using profile or HMM-based sequence aligners requires calculating these profiles or HMMs; one can use pre-calculated ones (which influences the set of entities available). Alternatively, it is possible to compare the *structures* of the proteins. Structure similarity is often viewed as a method for relating proteins that were similar further back in evolutionary history, with sequences that diverged beyond the point where one can identify their common ancestry; for example, the SCOP "fold," CATH "Architecture," and ECOD "X" levels are based on structure similarity. This is akin to using a more powerful telescope to look back in time [35]. A concern when relying only on structure similarity to study protein evolution is that these proteins share structures because these structures are especially favorable from a biophysical perspective. In other words, that what we see is merely a consequence of the biophysical properties and constraints [36], perhaps due to convergent evolution. To compare structures, we use one of many structural alignment methods. In fact, structural alignment is a vast field with many intricacies, far beyond the scope of this chapter. For more details, *see* [37–40] and below in the section highlighting structural alignment servers.

The similarity measure (be it based on sequence or on structure) can be local or global.[1] In global similarity, the proteins are considered in their entirety. In contrast, in local similarity, we consider subsections, so that proteins can be identified as similar even if there is only a partial match. The disadvantage of using a global similarity measure is that to be meaningful, we must first segment our proteins to pieces, which are similar in their entirety (e.g., domains); this creates a chicken-and-egg situation, because we want to segment the proteins in a way that we can find globally meaningful similarities. The disadvantage of using a local similarity measure is that it leads to non-transitive relationships: protein A that is locally similar to protein B, protein B that is locally similar to protein C, and at the same time proteins A and C have nothing in common ([1] has an illustration of this). Non-transitive relationships are counterintuitive when we think of the notion of similarity and especially when we integrate all these relationships into a unified (global) model of protein space.

2.3 Addressing Redundancy

The PDB is redundant, and some proteins are far more abundant than others (e.g., due to research interests of the scholars studying these proteins) [41]. This suggests that when seeking a global

[1] Notice that the terms used here characterize the similarity measure, not the style of navigation in protein space, to use the same terms as in the Needleman–Wunsch and Smith–Waterman sequence alignment algorithms.

perspective, one should either rely on nonredundant datasets or alternatively remove, or cull, the redundancy on their own. Notice that we consider an entity redundant if the dataset includes another copy of that entity: i.e., one that is (globally) similar to it. Hence, both the definition of the entities and the measures of similarity influence this redundancy removal process. There are software packages, and servers, that implement algorithms for removing redundancy: two popular ones are CD-HIT [42] and PISCES [43].

2.4 Data Structures for Global Representation

For a global perspective, one must derive a data structure, or an abstract model, from the dataset of all proteins and their comparisons. Scholars used three types of models: (1) networks, (2) classifications, and (3) maps (for a review of these, *see* [2]). A network is the data structure closest to the raw data. To construct it, one only needs to list the meaningful similarities, and the network is a straightforward representation of the entities (as nodes) and the similarities (as edges connecting these nodes.) A classification groups the entities into nonoverlapping sets of proteins. It is assumed that proteins in the same set in the classification (i.e., with the same classification) are similar to each other, while those not in the same set are not (or less so). The classifications are hierarchical, and proteins are grouped with decreasing degrees of similarity. Hence, to construct a classification, one needs to weight the importance of the similarities identified among the protein entities: emphasizing the ones that are within a set and downplaying the ones between sets. Finally, in a map, each protein is represented by a point, and the points are positioned in two or three dimensions, so that the distance between them approximates the dissimilarity between the proteins they represent. The mapping is calculated by first converting the measures of similarity between the protein entities to an all-by-all dissimilarity matrix, followed by a multidimensional scaling (MDS) to project this matrix to a lower (two or three) dimension. Because the position of a protein is not indicative of its relationship to other proteins in a straightforward manner, maps were not used for local navigation. Rather, the insights were derived from a global perspective [5, 14, 35, 44, 45].

2.5 Publicly Available Navigators for Protein Structure Space

Defining a meaningful nonredundant set of entities, calculating the relationships between them, and collecting all this information to a centralized data structure require both ingenuity and computational resources. Even more so, as the database of all protein structures (the PDB) is constantly growing, the calculations need to be routinely updated. Consequently, many groups have set up web servers with data for navigating protein structure space; these navigators have datasets which were curated, compared, and organized—some at a single time point (but possibly with a more elaborate organization)—while others are maintained up-to-date.

The navigators enable users to move in protein structure space as if they are using a navigation app. Some of the navigators offer their users a global perspective of protein structure space as well.

2.6 Navigators with a Global Perspective

The most established resources for navigating protein structure space are the hierarchical classifications; the popular ones are SCOP from the Murzin lab, CATH from the Orengo lab, and ECOD from the Grishin Lab; another popular classification—Pfam [46]—is not discussed here because it is based on sequence rather than structure. For a recent and extensive review of the classifications, *see* [47]. The classifications organize the data in a hierarchy: a user can gain a perspective of the whole space by drilling down, starting at the top. For example, starting at the highest level of SCOP, we see that structure space has regions of all-alpha domains, all-beta domains, alpha+beta domains, and alpha/beta domains, where the two latter classes include both alpha and beta elements, separated or intertwined, respectively [48]. Alternatively, one can search for a specific protein and consider the classification of its domains and the list of all its related proteins—ones whose domains are classified similarly (at different levels of the hierarchy.) In short, the data structure that is used in the hierarchies is a collection of sets (or lists), organized as a tree; each entity is classified in several (nested) sets (depending on the height of the hierarchy). The similarity measure used is based on the sequences (at the lower levels of the hierarchy) and structures (at the higher levels of the hierarchy). The entities classified are domains: nonoverlapping subsections of the protein chains, which cover all chain residues (or, in other words, each PDB chain is segmented into one or more domains such that each residue is part of exactly one domain). There is much discussion, and controversy, on what is the correct definition of domains [49–51]; that there are several domains databases (rather than one) is a clear indication of this.

In practical terms, domains are the entities classified in SCOP, CATH, ECOD, or in servers curating domains like CDD [52]. More formally, there are several (not necessarily overlapping) definitions of a domain [16, 17, 53]: (1) a structurally distinct region (perhaps a compact unit) [54], (2) a segment that is identified as an evolutionary unit based on observations of reuse in protein space, (3) an independently folding unit, and (4) a section with assigned biochemical function. The domains in the hierarchical classifications are defined based on reuse. Unfortunately, these domains, which are classified in the different databases, are not the same ones (for comparisons, *see* [50, 51, 55, 56]); a recent study estimates that only 60% of CATH domains have a similar SCOP counterpart [53]. Nonetheless, the domains in the hierarchical classifications have similar lengths of approximately 100 residues; this is the average for the distributions of domain lengths in the

SCOP, CATH, and ECOD (*see* Fig. 8b in [28]). Indeed, splitting a protein chain into domains is challenging [49], leading to many algorithmic methods devoted to this task (e.g., [54, 57–59]), and a significant amount of human intervention in some of the classifications (rather than only relying on automatic domain assignment procedures). Regardless of how automatic the procedure for identifying the domain boundaries, a fundamental problem remains if the domains are defined based on reuse: the reuse patterns in protein space are not simply reuse of segments of an appropriate length (~100 residues). Rather, it is a complicated pattern of nested segments that are reused to different extents [9, 27]. Consequently, there is more than one way to reduce this complex pattern into domain definitions. Due to this very same complexity, once the domains are defined, there are many instances of common parts (segments) between domains that are not wholly similar and are thus classified differently (at different levels of the hierarchy) [11, 29, 60–62].

The classification hierarchies maintain an up-to-date dataset representing the complete and current PDB, with an intuitive user interface. In CATH and ECOD, one can drill down the tree to explore different members of the sets; CATH also has a sunburst visualization, which indicates the relative sizes of the classified sets. Since the last version (1.75 in 2009) of the classic SCOP, the classification diverged into two variants: SCOPe and SCOP2. SCOPe [63] is a continuously and (mostly) automatically updated extension of classic SCOP. In contrast, SCOP2 [64] changed the data structure: rather than the classic tree of sets, it uses a network; the network representation (sometimes called graphs) is implemented with a web tool based on the visualization software Graphviz [65]. In all classifications, the user can search for a specific protein chain or domain and explore the local context of that protein within the data structure (typically, within the hierarchy), allowing the user to see proteins of similar sequence (with the same classification at the lower levels) and of similar structure (with the same classification at higher levels.)

2.7 Publicly Available Navigators for Local Environments of Structure Space

Another way of navigating protein structure space is zooming into a local region, while ignoring the global view, and exploring, by moving between such local environments. Starting from the protein of interest, we think of its local environment as a list of its structural neighbors (sorted from near to distant ones); we can then move in space by selecting one of these neighbors to see its slightly shifted local environment (centered on this neighbor.) We think of this process as navigating in protein structure space, like a driver following a navigation app without seeing the full landscape. For this, all one must have is the list of neighbors for each protein in the dataset. The entities considered are typically both PDB chains and domains (either taken from the classifications or calculated with

an automatic domain parser). Because the overall data structure is not considered, the structural alignment remains the most important computational component. Thus, such navigators were often set up by groups developing structural alignment methods. What transforms a structural alignment server into a useful navigator is speed: to navigate comfortably, the server must be fast. This is because when navigating, we search for structural neighbors repeatedly, each time starting at a different protein. Indeed, significant sophistication is needed to build servers that are up-to-date, fast, and comprehensive.

The differences between the structural alignment servers are largely due to the differences between the structural alignment methods. We list examples of structural alignment servers that allow users to locally navigate in protein structure space. The PDB website has precomputed structural alignments for a representative nonredundant dataset, calculated using the FATCAT aligner [66]. The European PDB website has PDBeFold [67], a structural alignment server based on the SSM aligner [68]. NCBI's server is called VAST+ and is based on the aligner VAST [69]. PhyreStorm [70] is a new server, which relies on TM-align [71] and offers a very comfortable navigation experience. Another new server is TopSearch (using the structural aligner TopMatch), which has the unique feature that it considers larger entities of protein oligomer [72].

2.8 DIY: Build-Your-Own Navigator

There are several reasons why scholars may want to customize their own navigator to explore protein structure space, or parts of it. First, the entities they wish to include may be specific to their problem: a set of proteins that is not covered in the public servers (perhaps a more redundant one), unpublished structures, or even predicted ones. Also, one may want to study subsections of proteins, which are different from chains or domains, for example, shorter themes [9] or loops [73]. Second, scholars may want to compare the entities themselves, as it gives them flexibility in the choice of a specific sequence or structure alignment program, full control over the parameters used, and the ability to enforce additional conditions when comparing proteins (e.g., a minimal alignment length). In some cases, even though there is a publicly available structural alignment server, it is not fast enough for navigating structure space; for these, one may prefer to pre-calculate all-against-all comparisons (e.g., using the parallel power of a computer cluster). We list just a few examples of comparison methods that were used in a similar context: HHSearch [30], Matt [74], CE [75], Mammoth [76], 3D-BLAST [77], FragBag [78], TM-align [71], SSM [68], GRASP [79], and STRUCTAL [80]. Third, the structural alignment servers do not offer a global perspective of structure space, only a local one, and one may be interested in this global perspective. Finally, scholars have different preferences when

exploring structures in a molecular viewer, both in terms of the viewer they are using and its configuration.

If the navigator is based on a network data structure, it is easy to build your own navigator with the network visualization tool Cytoscape [81] and its molecular viewer configuration apps CyToStruct [82] or structureViz [83]. To represent a part of protein space as a network, one needs to define the list of nodes (entities to be compared and the edges that connect them (pairs of entities that are similar). This is very easy to do with Cytoscape: a (fantastic) open-source network analysis and visualization tool. Given the list of nodes and edges, Cytoscape visualizes the information as a well-laid two-dimensional network; one can configure this visualization easily and extensively. For example, the color of the nodes may depend on the structural class of the entities they represent, and the thickness of the edges may depend on the similarity of the entities they connect. This provides the global perspective. To gain a local perspective, one would like to use a molecular viewer to study the nodes or edges and the structures or structural alignments they represent.

Molecular viewers need to be configured: these are sophisticated software tools, with many alternative settings. By configuring the molecular viewer, one can display and highlight the relevant parts in the protein structure. Popular molecular viewers are PyMOL [84], UCSF Chimera [85], JMol [86], VMD [87], and recently NGL—a particularly fast web-based viewer [88]; for a review of these and more, *see* [89]. There are two methods of configuring molecular viewers: (1) manually, using the graphical user interface (GUI) and (2) by running a script in the language specific to that viewer. Configuring the viewer manually is easier for a novice but far more tedious; configuring it via scripts requires command of the scripting language but facilitates repeated visualizations dramatically. To link the entities in Cytoscape with a molecular viewer, one can install one of two Cytoscape apps: structureViz or CyToStruct. structureViz is tightly coupled with UCSF Chimera. In structureViz, node attributes can specify PDB names, so that the corresponding pdb file opens in UCSF Chimera; the molecular viewer can also be configured via its GUI. In contrast, CyToStruct is suited for users who configure the molecular viewers via scripts; it is very powerful in that it allows using any molecular viewer, and within that viewer configuring anything that can be specified via a script, or equivalently, computed with that software.

CyToStruct can run any molecular viewer (and any external program in general) from all nodes and edges (a menu opens when right-clicking on it), with scripts that are tailored to each node or edge. To configure CyToStruct, the user has to specify the external program, a template of script to be run, and a file with node- or edge-specific data for that template. CyToStruct then creates the runnable script by infusing the node- or edge-specific data into the

template and runs the molecular viewer with a copy of this script. The source code of CyToStruct is publicly available (https://bitbucket.org/sergeyn/cytostruct/wiki/Home), along a series of demos that users can rely on as a starting point. The demos include visualization using the four popular molecular viewers (each with their own syntax), configuring the visualization of complete structures, protein interfaces, structurally aligning multiple structures, and selecting specific residues. CyToStruct can also be used within the web-based version of Cytoscape (Cytoscape.js), to provide an online visualization combining a network and a molecular viewer.

We present two examples for DIY navigators. The first is the navigator that Nepomnyachiy et al. customized for a global view of protein structure space [11]. The entities, or nodes in the network, are 9710 SCOP domains (70% nonredundant set). These domains were compared using the structural aligner SSM [68]; for sufficiently meaningful alignments, Nepomnyachiy et al. calculated measures of the similarity of the domains. Then, they define several networks, each characterized by its edges, which connect all domain pairs that were aligned with parameters better than some fixed thresholds: a minimal alignment length (55, 75 residues), maximal RMSD (2, 2.5, and 3 Å), and minimal percent sequence similarity (30, 40, and 50%). By coloring the nodes based on their SCOP class, all-alpha, all-beta, alpha/beta, and alpha+beta, they could see that protein structure space has a continuous region (the alpha/beta domains) and discrete regions [11]. The Cytoscape networks provide a global view, but navigating in specific regions of structure space is also interesting. Nepomnyachiy et al. link and configure the molecular viewer using CyToStruct [82] to see the domains and the alignments and package and distribute the data and configuration files (http://cs.haifa.ac.il/~trachel/domain_motif_networks/), allowing anyone to study protein structure space in this way.

2.9 Case Study

We present here a new example, where Cytoscape and CyToStruct are used to navigate protein space for function inference. The navigator helps because a careful examination of populated regions in the protein universe can help decipher unknown qualities of proteins found in these regions. Here, we demonstrate this using substrate-binding proteins (SBPs) [90]. SBPs are involved in transport of substrates into the cell, where their role is to recognize the substrate and relay it to its transmembrane transporter. Although they vary in size and share relatively low sequence similarity, they share a similar, highly conserved, fold. In general, their shape is a lung-like structure, formed of two structurally similar globular domains, connected by a hinge. The hinge facilitates alteration between substrate-free and substrate-bound conformations; substrate binding to a cavity between the two domains brings them closer to one another, into a bound, or "closed," conformation.

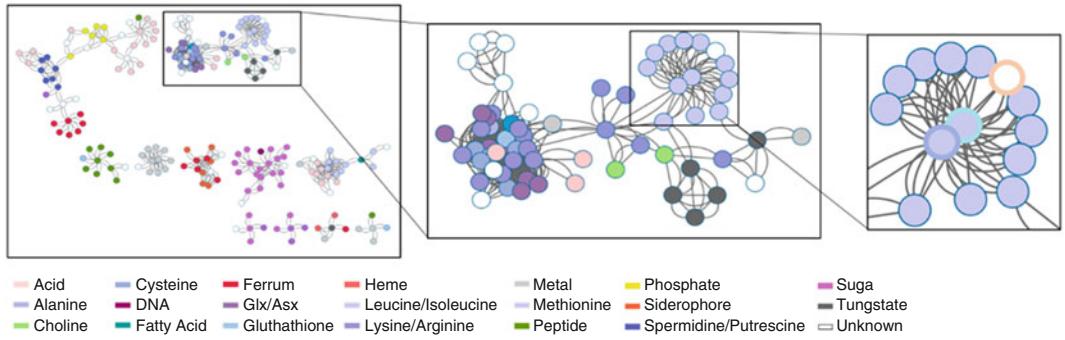

Fig. 1 Navigating protein structure space to study proteins with unknown function. *Left panel*: network of substrate-binding proteins. Each node represents a single PDB chain; two nodes are connected by an edge if they share some sequential and structural similarity. The nodes are colored according to the substrate; see color-code at the bottom. White nodes represent proteins of unknown function. Middle panel: zooming-in on the top-right cluster. This cluster is composed mostly of amino acid binding proteins. *Right panel*: zooming-in on one connected component. Violet nodes represent methionine binding proteins. 4ntl, represented here by a white node encircled in orange, has no bound substrate, and its function is unknown. It is connected to the two central nodes, 4qhq and 3tqw (encircled in blue and purple). The figure was created using Cytoscape [94]

A dataset of binding proteins was collected from the 70% NR PDB, by using the website text search. This dataset was extended by adding proteins that share at least 30% of their sequence, over a segment of at least 35 residues, with an RMSD lower than 3.5 Å, with the proteins in the initial dataset. Cytoscape generated the network (Fig. 1, left panel), where each node represents a protein in the dataset and two nodes are connected if the proteins are deemed related (more than 30% sequence similarity, over more than 35 residues, with less than 3.5 Å RMSD). With this particular choice, several clusters are formed, so that in general SBPs which bind similar substrates (as evident in their PDB structures) belong to the same cluster (Fig. 1, left panel). Thus, their binding preferences and modes of interaction with the substrate can be predicted by the cluster they are found in. For example, one cluster is formed by SBPs that bind amino acids (Fig. 1, middle panel). A connected component within this cluster contains SBPs that generally bind methionine (Fig. 1, right panel). The substrate of one of these SBPs (white, encircled in orange, pdb 4ntl) is unknown. However, in this case we can suggest a likely hypothesis is that it also binds methionine. The sequence identity between the query and its neighbors is less than 40%; thus this functional inference, which is in keeping with the conjecture listed in CDD [52], is not trivial [91].

Using CyToStruct [82] and our molecular viewer of choice, we can examine this hypothesis in detail. Reassuringly, comparison of this query protein with its first neighbors in protein space (Fig. 1, right panel, the two nodes at the center of the cluster, encircled in cyan and green) supports this inference, as they share high structural similarity to the query (Fig. 2a). As both neighbors (pdb 4qhq

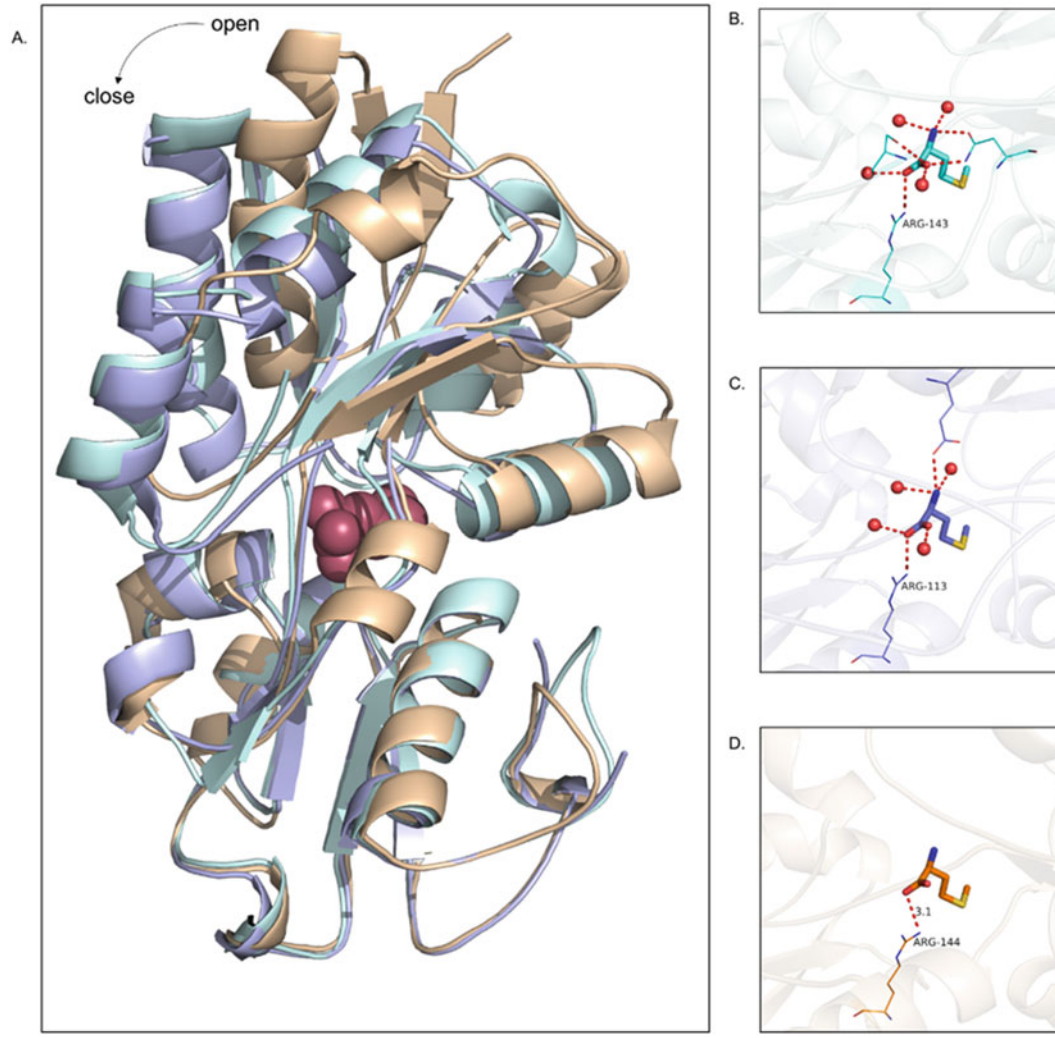

Fig. 2 Methionine binding in the SBPs 4ntl, 4qhq, and 3tqw. (**A**) Structural superposition of the 4ntl query (orange) with 4qhq and 3tqw (blue and purple), respectively. The superposition is over the C-terminal lobe to highlight the conformational change between the bound (close; 4qhq and 3tqw) and unbound (open; query) states of the SBPs. The bound methionine is shown in red spheres. (**B**) The methionine binding site in 4qhq. Methionine is presented using sticks model, and the polar residues of the binding site are depicted as wireframes. The hydrogen bonds that mediate methionine's interactions with these residues and with water molecules (red sphere) are marked as red dashed lines. The highly conserved Arg143 is also marked. (**C**) The methionine binding site in 3tqw. The highly conserved Arg113, equivalent of Arg143 in panel **B**, is marked. (**D**) Putative encounter complex between methionine and the query. Arg144 (depicted as wireframe) has the same location and rotameric state as its equivalents: Arg144 of 4qhq and Arg113 of 3tqw. The dashed line shows the putative hydrogen bond, which could form between the arginine and the methionine carbonyl group. The figure was created using the Pymol molecular viewer [84]

and pdb 3qwl) have a bound methionine in their PDB structure (Fig. 2b, c), a superposition of the structures can even be used to suggest a putative binding site (Fig. 2d). Evolutionary analysis,

using ConSurf [92, 93], shows that the binding cavity is highly conserved, providing further support for the inferred function and binding mode. In particular, the three binding sites feature a highly conserved arginine residue (conservation grade of 9 on a 1–9 scale). Furthermore, in all three proteins, the arginine populates the exact same rotameric state, which allows it to form a hydrogen bond with the methionine substrate (Fig. 2b–d). In addition, water molecules that participate in the binding are also found in all the structures. However, not all the interactions that are found in the two bound states have equivalents in the query, and the structural superposition indicates that it is in an open conformation (Fig. 2a). It suggests that binding may follow the population shift theory, where methionine is initially recognized by the conserved arginine residue in the open conformation. This interaction may induce a shift of the protein to its closed conformation, where additional residues interact with methionine. Further investigation is needed to examine this suggestion.

3 Conclusions and Outlook

How did proteins emerge in evolution, and how do they evolve? Theoretically, a protein could emerge and evolve by linking one amino acid after another. Scholars believe that this approach is doomed, because the vast majority of polypeptide chains would not even fold. Thus, we presume that proteins emerged by mixing and matching short amino acid fragments (peptides) from the primordial soup, evolving by recombination, decoration, and mutation. Lupas et al. wrote an insightful review of this [29]. While most protein scientists would agree with this suggested scenario, the mechanics and details of the process which gave rise to proteins, and that govern their evolution, is still yet to be understood.

This leads to two observations: (1) We can look for clues to address these fundamental questions in current proteins by studying the reuse patterns in all proteins of known structure. (2) We can mine the evolutionary signal to identify common ancestry and improve methods of protein similarity search, function annotation, and design. For both of these, navigating in protein space can be very useful.

References

1. Kolodny R, Pereyaslavets L, Samson AO, Levitt M (2012) On the universe of protein folds. Annu Rev Biophys 42:559. https://doi.org/10.1146/annurev-biophys-083012-130432
2. Ben-Tal N, Kolodny R (2014) Representation of the protein universe using classifications, maps, and networks. Israel J Chem 54:1286
3. Zeldovich KB, Shakhnovich EI (2008) Understanding protein evolution: from protein

physics to Darwinian selection. Annu Rev Phys Chem 59:105–127

4. Trifonov EN, Berezovsky IN (2003) Evolutionary aspects of protein structure and folding. Curr Opin Struct Biol 13(1):110–114
5. Choi IG, Kim SH (2006) Evolution of protein structural classes and protein sequence families. Proc Natl Acad Sci U S A 103 (38):14056–14061. https://doi.org/10.1073/pnas.0606239103
6. Dokholyan NV, Shakhnovich B, Shakhnovich EI (2002) Expanding protein universe and its origin from the biological big bang. Proc Natl Acad Sci 99(22):14132–14136. https://doi.org/10.1073/pnas.202497999
7. Alva V, Remmert M, Biegert A, Lupas AN, Söding J (2010) A galaxy of folds. Protein Sci 19(1):124–130. https://doi.org/10.1002/pro.297
8. Farías-Rico JA, Schmidt S, Höcker B (2014) Evolutionary relationship of two ancient protein superfolds. Nat Chem Biol 10 (9):710–715. https://doi.org/10.1038/nchembio.1579 http://www.nature.com/nchembio/journal/v10/n9/abs/nchembio.1579.html#supplementary-information
9. Nepomnyachiy S, Ben-Tal N, Kolodny R (2017) Complex evolutionary footprints revealed in an analysis of reused protein segments of diverse lengths. Proc Natl Acad Sci U S A 114:11703
10. Skolnick J, Arakaki AK, Lee SY, Brylinski M (2009) The continuity of protein structure space is an intrinsic property of proteins. Proc Natl Acad Sci 106:15690. https://doi.org/10.1073/pnas.0907683106
11. Nepomnyachiy S, Ben-Tal N, Kolodny R (2014) Global view of the protein universe. Proc Natl Acad Sci 111:11691. https://doi.org/10.1073/pnas.1403395111
12. Mackenzie CO, Zhou J, Grigoryan G (2016) Tertiary alphabet for the observable protein structural universe. Proc Natl Acad Sci U S A 113(47):E7438–E7447
13. Kolodny R, Petrey D, Honig B (2006) Protein structure comparison: implications for the nature of 'fold space', and structure and function prediction. Curr Opin Struct Biol 16 (3):393–398
14. Osadchy M, Kolodny R (2011) Maps of protein structure space reveal a fundamental relationship between protein structure and function. Proc Natl Acad Sci 108 (30):12301–12306. https://doi.org/10.1073/pnas.1102727108
15. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The Protein Data Bank. Nucleic Acids Res 28(1):235–242
16. Koehl P (2006) Protein structure classification. In: Reviews in Computational Chemistry. John Wiley & Sons, Inc., New York, pp 1–55. https://doi.org/10.1002/0471780367.ch1
17. Ponting CP, Russell RR (2002) The natural history of protein domains. Annu Rev Biophys Biomol Struct 31(1):45–71. https://doi.org/10.1146/annurev.biophys.31.082901.134314
18. Vogel C, Berzuini C, Bashton M, Gough J, Teichmann SA (2004) Supra-domains: evolutionary units larger than single protein domains. J Mol Biol 336(3):809–823. https://doi.org/10.1016/j.jmb.2003.12.026
19. Kolodny R, Koehl P, Guibas L, Levitt M (2002) Small libraries of protein fragments model native protein structures accurately. J Mol Biol 323(2):297–307
20. Vanhee P, Verschueren E, Baeten L, Stricher F, Serrano L, Rousseau F, Schymkowitz J (2011) BriX: a database of protein building blocks for structural analysis, modeling and design. Nucleic Acids Res 39(Suppl 1):D435–D442
21. Davis FP, Sali A (2005) PIBASE: a comprehensive database of structurally defined protein interfaces. Bioinformatics 21(9):1901–1907
22. Vanhee P, Reumers J, Stricher F, Baeten L, Serrano L, Schymkowitz J, Rousseau F (2009) PepX: a structural database of non-redundant protein–peptide complexes. Nucleic Acids Res 38(Suppl 1):D545–D551
23. Fernandez-Fuentes N, Dybas JM, Fiser A (2010) Structural characteristics of novel protein folds. PLoS Comput Biol 6(4):e1000750
24. Ovchinnikov S, Park H, Varghese N, Huang P-S, Pavlopoulos GA, Kim DE, Kamisetty H, Kyrpides NC, Baker D (2017) Protein structure determination using metagenome sequence data. Science 355(6322):294–298
25. Pieper U, Eswar N, Davis FP, Braberg H, Madhusudhan MS, Rossi A, Marti-Renom M, Karchin R, Webb BM, Eramian D (2006) MODBASE: a database of annotated comparative protein structure models and associated resources. Nucleic Acids Res 34(Suppl 1): D291–D295
26. Lo Conte L, Ailey B, Hubbard TJP, Brenner SE, Murzin AG, Chothia C (2000) SCOP: a structural classification of proteins database. Nucleic Acids Res 28(1):257–259
27. Orengo C, Michie A, Jones S, Jones D, Swindells M, Thornton J (1997) CATH-a hierarchic classification of protein domain structures. Structure 5(8):1093–1108

28. Cheng H, Schaeffer RD, Liao Y, Kinch LN, Pei J, Shi S, Kim B-H, Grishin NV (2014) ECOD: an evolutionary classification of protein domains. PLoS Comput Biol 10(12): e1003926. https://doi.org/10.1371/journal.pcbi.1003926
29. Lupas AN, Ponting CP, Russell RB (2001) On the evolution of protein folds: are similar motifs in different protein folds the result of convergence, insertion, or relics of an ancient peptide world? J Struct Biol 134 (2–3):191–203
30. Soding J (2005) Protein homology detection by HMM-HMM comparison. Bioinformatics 21(7):951–960
31. Eddy SR (2009) A new generation of homology search tools based on probabilistic inference. Genome Inform 1:205–211
32. Alva V, Söding J, Lupas AN (2016) A vocabulary of ancient peptides at the origin of folded proteins. elife 4:e09410
33. Kosloff M, Kolodny R (2008) Sequence-similar, structure-dissimilar protein pairs in the PDB. Proteins 71(2):891–902
34. Narunsky A, Nepomnyachiy S, Ashkenazy H, Kolodny R, Ben-Tal N (2015) ConTemplate suggests possible alternative conformations for a query protein of known structure. Structure 23(11):2162–2170
35. Holm L, Sander C (1996) Mapping the protein universe. Science 273(5275):595–603
36. Skolnick J, Gao M, Zhou H (2014) On the role of physics and evolution in dictating protein structure and function. Israel J Chem 54 (8–9):1176–1188
37. Hasegawa H, Holm L (2009) Advances and pitfalls of protein structural alignment. Curr Opin Struct Biol 19(3):341–348
38. Kolodny R, Koehl P, Levitt M (2005) Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. J Mol Biol 346(4):1173–1188
39. Kolodny R, Linial N (2004) Approximate protein structural alignment in polynomial time. Proc Natl Acad Sci U S A 101 (33):12201–12206
40. Carugo O (2007) Recent progress in measuring structural similarity between proteins. Curr Protein Pept Sci 8(3):241
41. Yanover C, Vanetik N, Levitt M, Kolodny R, Keasar C (2014) Redundancy-weighting for better inference of protein structural features. Bioinformatics 30(16):2295–2301
42. Li W, Godzik A (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22(13):1658–1659
43. Wang G, Dunbrack RL (2003) PISCES: a protein sequence culling server. Bioinformatics 19 (12):1589–1591. https://doi.org/10.1093/bioinformatics/btg224
44. Choi I-G, Kim S-H (2007) Global extent of horizontal gene transfer. Proc Natl Acad Sci 104(11):4489–4494. https://doi.org/10.1073/pnas.0611557104
45. Orengo CA, Flores TP, Taylor WR, Thornton JM (1993) Identification and classification of protein fold families. Protein Eng 6 (5):485–500. https://doi.org/10.1093/protein/6.5.485
46. Finn RD, Bateman A, Clements J, Coggill P, Eberhardt RY, Eddy SR (2014) Pfam: the protein families database. Nucleic Acids Res 42: D222. https://doi.org/10.1093/nar/gkt1223
47. Pearl FMG, Sillitoe I, Orengo CA (2015) Protein structure classification. In: eLS. John Wiley & Sons, Ltd., New York. https://doi.org/10.1002/9780470015902.a0003033.pub3
48. Levitt M, Chothia C (1976) Structural patterns in globular proteins. Nature 261 (5561):552–558
49. Holland TA, Veretnik S, Shindyalov IN, Bourne PE (2006) Partitioning protein structures into domains: why is it so difficult? J Mol Biol 361(3):562–590
50. Hadley C, Jones DT (1999) A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. Structure 7 (9):1099–1112
51. Day R, Beck DAC, Armen RS, Daggett V (2003) A consensus view of fold space: combining SCOP, CATH, and the Dali Domain Dictionary. Protein Sci 12(10):2150–2160. https://doi.org/10.1110/ps.0306803
52. Marchler-Bauer A, Lu S, Anderson JB, Chitsaz F, Derbyshire MK, DeWeese-Scott C, Fong JH, Geer LY, Geer RC, Gonzales NR (2010) CDD: a conserved domain database for the functional annotation of proteins. Nucleic Acids Res 39(Suppl 1):D225–D229
53. Kelley LA, Sternberg MJ (2015) Partial protein domains: evolutionary insights and bioinformatics challenges. Genome Biol 16(1):1–3. https://doi.org/10.1186/s13059-015-0663-8
54. Veretnik S, Gu J, Wodak S (2009) Identifying structural domains in proteins. In: Gu G, Bourne P (eds) Structural bioinformatics, 2nd edn. Wiley-Blackwell, Hoboken, NJ, pp 485–513
55. Schaeffer RD, Jonsson AL, Simms AM, Daggett V (2011) Generation of a consensus protein domain dictionary. Bioinformatics 27

(1):46–54. https://doi.org/10.1093/bioinformatics/btq625
56. Csaba G, Birzele F, Zimmer R (2009) Systematic comparison of SCOP and CATH: a new gold standard for protein structure analysis. BMC Struct Biol 9(1):23
57. Redfern OC, Harrison A, Dallman T, Pearl FM, Orengo CA (2007) CATHEDRAL: a fast and effective algorithm to predict folds and domain boundaries from multidomain protein structures. PLoS Comput Biol 3(11): e232. https://doi.org/10.1371/journal.pcbi.0030232
58. Zhou H, Xue B, Zhou Y (2007) DDOMAIN: dividing structures into domains using a normalized domain–domain interaction profile. Protein Sci 16(5):947–955. https://doi.org/10.1110/ps.062597307
59. Alexandrov N, Shindyalov I (2003) PDP: protein domain parser. Bioinformatics 19 (3):429–430. https://doi.org/10.1093/bioinformatics/btg006
60. Krishna SS, Grishin NV (2005) Structural drift: a possible path to protein fold change. Bioinformatics 21(8):1308–1310
61. Pascual-García A, Abia D, Ortiz ÁR, Bastolla U (2009) Cross-over between discrete and continuous protein structure space: insights into automatic classification and networks of protein structures. PLoS Comput Biol 5(3): e1000331. https://doi.org/10.1371/journal.pcbi.1000331
62. Edwards H, Deane CM (2015) Structural bridges through fold space. PLoS Comput Biol 11(9):e1004466
63. Fox NK, Brenner SE, Chandonia J-M (2014) SCOPe: structural classification of proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Res 42(D1):D304–D309. https://doi.org/10.1093/nar/gkt1240
64. Andreeva A, Howorth D, Chothia C, Kulesha E, Murzin AG (2013) SCOP2 prototype: a new approach to protein structure mining. Nucleic Acids Res 42:D310. https://doi.org/10.1093/nar/gkt1242
65. Ellson J, Gansner E, Koutsofios L, North SC, Woodhull G (2001) Graphviz—open source graph drawing tools. In: International symposium on graph drawing. Springer, Heidelberg, pp 483–484
66. Prlić A, Bliven S, Rose PW, Bluhm WF, Bizon C, Godzik A, Bourne PE (2010) Pre-calculated protein structure alignments at the RCSB PDB website. Bioinformatics 26 (23):2983–2985. https://doi.org/10.1093/bioinformatics/btq572
67. Krissinel E, Henrick K (2003) Protein structure comparison in 3D based on secondary structure matching (SSM) followed by C-alpha alignment, scored by a new structural similarity function. Proceedings of the 5th International Conference on Molecular Structural Biology, Vienna, vol. 88
68. Krissinel E, Henrick K (2004) Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr D 60(Pt 12 Pt 1):2256–2268
69. Madej T, Lanczycki CJ, Zhang D, Thiessen PA, Geer RC, Marchler-Bauer A (2014) MMDB and VAST+: tracking structural similarities between macromolecular complexes. Nucleic Acids Res D42:D297. https://doi.org/10.1093/nar/gkt1208
70. Mezulis S, Sternberg MJE, Kelley LA (2016) PhyreStorm: a web server for fast structural searches against the PDB. J Mol Biol 428 (4):702–708. https://doi.org/10.1016/j.jmb.2015.10.017
71. Zhang Y, Skolnick J (2005) TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Res 33 (7):2302–2309. https://doi.org/10.1093/nar/gki524
72. Wiederstein M, Gruber M, Frank K, Melo F, Sippl Manfred J (2014) Structure-based characterization of multiprotein complexes. Structure 22(7):1063–1070. https://doi.org/10.1016/j.str.2014.05.005
73. Berezovsky IN, Guarnera E, Zheng Z (2017) Basic units of protein structure, folding, and function. Prog Biophys Mol Biol 128:85–99. https://doi.org/10.1016/j.pbiomolbio.2016.09.009
74. Menke M, Berger B, Cowen L (2008) Matt: local flexibility aids protein multiple structure alignment. PLoS Comput Biol 4(1):e10
75. Shindyalov I, Bourne P (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng 11(9):739–747
76. Ortiz A, Strauss C, Olmea O (2002) MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. Protein Sci 11(11):2606–2621
77. Tung CH, Huang JW, Yang JM (2007) Kappa-alpha plot derived structural alphabet and BLOSUM-like substitution matrix for rapid search of protein structure database. Genome Biol 8(3):R31
78. Budowski-Tal I, Nov Y, Kolodny R (2010) FragBag, an accurate representation of protein structure, retrieves structural neighbors from

the entire PDB quickly and accurately. Proc Natl Acad Sci U S A 107(8):3481–3486. https://doi.org/10.1073/pnas.0914097107

79. Petrey D, Xiang Z, Tang CL, Xie L, Gimpelev M, Mitros T, Soto CS, Goldsmith-Fischman S, Kernytsky A, Schlessinger A, Koh IY, Alexov E, Honig B (2003) Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. Proteins 53(Suppl 6):430–435. https://doi.org/10.1002/prot.10550
80. Subbiah S, Laurents DV, Levitt M (1993) Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core. Curr Biol 3(3):141–148
81. Saito R, Smoot ME, Ono K, Ruscheinski J, Wang P-L, Lotia S, Pico AR, Bader GD, Ideker T (2012) A travel guide to Cytoscape plugins. Nat Methods 9(11):1069–1076
82. Nepomnyachiy S, Ben-Tal N, Kolodny R (2015) CyToStruct: augmenting the network visualization of cytoscape with the power of molecular viewers. Structure 23(5):941–948
83. Morris JH, Huang CC, Babbitt PC, Ferrin TE (2007) structureViz: linking Cytoscape and UCSF chimera. Bioinformatics 23 (17):2345–2347. https://doi.org/10.1093/bioinformatics/btm329
84. Schrodinger, LLC (2010) The PyMOL molecular graphics system, Version 1.3r1. Schrodinger, LLC, New York
85. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE (2004) UCSF chimera—a visualization system for exploratory research and analysis. J Comput Chem 25(13):1605–1612
86. Jmol: an open-source java viewer for chemical structure in 3D. http://www.jmol.org/
87. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14(1):33–38
88. Rose AS, Hildebrand PW (2015) NGL viewer: a web application for molecular visualization. Nucleic Acids Res 43(Web Server issue): W576–W579. https://doi.org/10.1093/nar/gkv402
89. O'Donoghue SI, Goodsell DS, Frangakis AS, Jossinet F, Laskowski RA, Nilges M, Saibil HR, Schafferhans A, Wade RC, Westhof E (2010) Visualization of macromolecular structures. Nat Methods 7:S42–S55
90. Berntsson RP-A, Smits SH, Schmitt L, Slotboom D-J, Poolman B (2010) A structural classification of substrate-binding proteins. FEBS Lett 584(12):2606–2617
91. Radivojac P, Clark WT, Oron TR, Schnoes AM, Wittkop T, Sokolov A, Graim K, Funk C, Verspoor K, Ben-Hur A (2013) A large-scale evaluation of computational protein function prediction. Nat Methods 10 (3):221–227
92. Glaser F, Pupko T, Paz I, Bell RE, Bechor-Shental D, Martz E, Ben-Tal N (2003) ConSurf: identification of functional regions in proteins by surface-mapping of phylogenetic information. Bioinformatics 19(1):163–164
93. Ashkenazy H, Abadi S, Martz E, Chay O, Mayrose I, Pupko T, Ben-Tal N (2016) ConSurf 2016: an improved methodology to estimate and visualize evolutionary conservation in macromolecules. Nucleic Acids Res 44(W1): W344–W350
94. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13(11):2498–2504. https://doi.org/10.1101/gr.1239303