

MASARYK UNIVERSITY
FACULTY OF SCIENCE
NATIONAL CENTRE FOR BIOMOLECULAR RESEARCH

**Annotation and visualization
of protein secondary structure**

Doctoral thesis

Adam Midlik

Supervisor: doc. RNDr. Radka Svobodová, Ph.D.

Brno 2022

Bibliographic entry

Author: Mgr. et Mgr. Adam Midlik
Faculty of Science, Masaryk University
National Centre for Biomolecular Research

Title of the Thesis: Annotation and visualization of protein secondary structure

Degree Programme: Biomolecular chemistry and bioinformatics

Supervisor: doc. RNDr. Radka Svobodová, Ph.D.

Academic Year: 2021/2022

Number of Pages: x + 142

Keywords: protein structure; secondary structure element; SSE; annotation; cytochrome P450; CYP; secondary structure consensus; visualization; SecStrAnnotator; OverProt; 2DProts

Bibliografický záznam

Autor: Mgr. et Mgr. Adam Midlik
Přírodovědecká fakulta, Masarykova univerzita
Národní centrum pro výzkum biomolekul

Název práce: Anotace a vizualizace sekundární struktury proteinů

Studijní program: Biomolekulární chemie a bioinformatika

Školitel: doc. RNDr. Radka Svobodová, Ph.D.

Akademický rok: 2021/2022

Počet stran: x + 142

Klíčová slova: struktura proteinu; element sekundární struktury; SSE;
anotace; cytochrom P450; CYP;
konsensus sekundární struktury; vizualizace;
SecStrAnnotator; OverProt; 2DProts

Abstract

Secondary structure elements (SSEs) are inherent parts of protein structures, and their arrangement is characteristic for each protein family. Therefore, annotation of the SSEs can facilitate orientation in the vast number of structures which is now available for many protein families. The SSE annotation also provides a way to identify and annotate the key regions, like active sites and channels, and subsequently answer the key research questions, such as understanding the protein function. However, until recently, there were no automated methods for annotation of the SSEs. Similarly, there were no methods for finding the set of characteristic SSEs (the secondary structure consensus), though it would provide a useful overview of the general architecture of the whole family.

This doctoral thesis addresses several questions related to the SSE annotation. First, it presents SecStrAnnotator, a new tool for automated annotation of SSEs based on a provided annotation template. Then, it demonstrates an application of this tool in a detailed analysis of the SSEs in the cytochrome P450 family. Next, it introduces a tool for creating the secondary structure consensus, OverProt, and a database of its results for all available protein families. Finally, it shows how the SSE annotations can be exploited in the field of protein structure visualization, namely in the 2D diagram generator 2DProts.

Abstrakt

Elementy sekundární struktury (SSE) jsou nedílnou součástí proteinových struktur a jejich uspořádání je charakteristické pro každou proteinovou rodinu. Proto může anotace těchto SSE usnadnit orientaci v množství struktur, jaké je dnes dostupné pro mnoho proteinových rodin. Anotace SSE nám též umožňuje identifikovat a anotovat klíčové regiony, jako jsou aktivní místa a tunely, a posléze zodpovědět klíčové vědecké otázky, například porozumět funkcím proteinů. Nicméně až donedávna neexistovaly automatické metody pro anotaci SSE. Taktéž neexistovaly metody pro nalezení množiny charakteristických SSE (tj. konsensu sekundární struktury), přestože tento konsensus by nám dal užitečný náhled na obecnou architekturu celé rodiny.

Tato disertační práce se zaměřuje na několik otázek spojených s anotací SSE. Nejdříve prezentuje SecStrAnnotator, nový nástroj pro automatickou anotaci SSE na základě dodané anotační šablony. Pak demonstruje použití tohoto nástroje pro detailní analýzu SSE v rodině cytochromů P450. Poté představuje nástroj OverProt pro vytvoření konsensu sekundární struktury a databázi jeho výsledků pro všechny dostupné proteinové rodiny. Nakonec ukazuje, jak lze anotované SSE využít v oblasti vizualizace proteinových struktur, konkrétně v rámci nástroje 2DProts pro generování 2D diagramů.

Acknowledgement

At this point, I would like to thank my supervisor doc. RNDr. Radka Svobodová, Ph.D., my consultant prof. RNDr. Jaroslav Koča, DrSc., and doc. RNDr. Karel Berka, Ph.D., for their exciting ideas, encouragement, and advice during my studies. I am also thankful to my colleagues from the Laboratory of Computational Chemistry and the Structural Bioinformatics group, who were always willing to help when I needed them. My deepest gratitude belongs to my family and friends, for their support and inexhaustible jokes about my educational process.

Declaration

Hereby I declare that this thesis is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Brno, 2022

Adam Midlik

Original publications and author contribution

This thesis is based on four main publications of the author (Adam Midlik, AM).

1. Midlik,A., Hutařová Vařeková,I., Hutař,J., Moturu,T.R., Navrátilová,V., Koča, J., Berka,K., Svobodová Vařeková,R. (2019) Automated family-wide annotation of secondary structure elements. In Kister,A.E. (ed.), *Protein Supersecondary Structures*. Humana Press, New York, NY. Vol. 1958, pp. 47–71. https://doi.org/10.1007/978-1-4939-9161-7_3.

AM created and tested the software, co-wrote the manuscript, and prepared the figures.

2. Midlik,A., Navrátilová,V., Moturu,T.R., Koča,J., Svobodová,R., Berka,K. (2021) Uncovering of cytochrome P450 anatomy by SecStrAnnotator. *Scientific Reports*, **11**, 12345. <https://doi.org/10.1038/s41598-021-91494-8>.

AM created the software, prepared the datasets, analysed the data, participated in interpreting the results, co-wrote the manuscript, and prepared most of the figures.

3. Midlik,A., Hutařová Vařeková,I., Hutař,J., Chareshneu,A., Berka,K., Svobodová,R. (2022) OverProt: secondary structure consensus for protein families. *Bioinformatics*, (in press). <https://doi.org/10.1093/bioinformatics/btac384>.

AM created the software, deployed the web server, and co-wrote the manuscript.

4. Hutařová Vařeková,I., Hutař,J., Midlik,A., Horský,V., Hladká,E., Svobodová, R., Berka,K. (2021) 2DProts: database of family-wide protein secondary structure diagrams. *Bioinformatics*, **37**, 4599–4601.
<https://doi.org/10.1093/bioinformatics/btab505>.

AM participated in the design and testing of the software, implemented a small fraction of the software and worked on its integration with OverProt and SecStrAnnotator, participated in writing the manuscript, and prepared some of the figures for supplementary material.

Contents

1	Introduction	1
2	Theory and Methods	3
2.1	Protein secondary structure	3
	Irregularities	6
	Secondary structure assignment	7
	Annotation	8
	Visualization	9
2.2	Classification of protein structures	13
	SCOP	14
	CATH	15
	ECOD	15
3	Synopsis of the Results	17
3.1	Automated family-wide annotation of secondary structure elements	18
3.2	Uncovering of cytochrome P450 anatomy by SecStrAnnotator	20
3.3	OverProt: secondary structure consensus for protein families	20
3.4	2DProts: database of family-wide protein secondary structure diagrams	23
4	Conclusion	25
5	Main Publications	27
	Automated family-wide annotation of secondary structure elements	29
	Uncovering of cytochrome P450 anatomy by SecStrAnnotator	57
	OverProt: secondary structure consensus for protein families	83
	2DProts: database of family-wide protein secondary structure diagrams	103

6 Other Publications	117
LiteMol suite: interactive web-based visualization of large-scale macro- molecular structure data	118
Sanguinarine is reduced by NADH through a covalent adduct	120
Visualization and analysis of protein structures with LiteMol suite	122
7 Curriculum Vitae	125
List of Abbreviations	131
References	133

Chapter 1

Introduction

Since the first experimentally determined protein structure [1], the number of known protein structures has been constantly growing. The global archive of the experimentally determined macromolecular structures, Protein Data Bank (PDB) [2], currently contains more than 190 000 entries, most of them belonging to proteins (June 2022).

With this growth, it soon became clear that the individual proteins are not entirely unique, but can be grouped into a limited number of protein families with shared structural arrangement [3–5]. This led to the development of multiple databases focused on classification of the protein structures into families based on their structural similarity and evolutionary and functional relationships (e.g. CATH [6], SCOP [7]).

Nowadays, the number of known families is not growing significantly [6] (although the recent advances in protein structure prediction may soon reveal some new protein families [8, 9]). Nonetheless, the existing families expand as more and more protein structures get collected in each family. This allows us to study protein structures in the context of a whole family, which can help in understanding their biological functions and mechanisms of action [10].

However, the increasing size of the protein families also brings its own challenges. Orientation in the mass of structures in a large family can be difficult. From the bare structural data, it is not clear which part of one structure corresponds to which part of another. This problem can be solved by a consistent annotation, i.e. labelling the topologically equivalent parts of each structure with the same label. This annotation can be done on several levels, starting from the individual residues [11], through the secondary structure elements (helices and strands) [11, 12], to larger regions [6, 7, 13].

We humans naturally perceive a protein structure as a bunch of secondary structure elements (SSEs) and orient within the structure in relation to these elements. The communities around several protein families have even developed specialized labels for annotating the SSEs within the family [11, 12, 14, 15]. These labels are then very useful when comparing existing structures, describing new ones, or generalizing observations over the whole family [16, 17]. Annotated SSEs are also used as reference points to describe the position of key regions, such as catalytic sites [18], substrate recognition sites [19, 20], channels [21], or protein-protein interfaces [22].

A nice illustrative example is the cytochrome P450 family, with a well-established nomenclature of helices and strands [14, 23]. This SSE nomenclature set the foundation for a classification system of multiple different channels based on their position relative to the annotated SSEs [21], which in turn helped to elucidate the channel preferences of individual cytochromes P450 and their substrates [24, 25].

The SSEs in these families are usually annotated manually, based on another, already annotated structure – a template. However, manual annotation can be very tedious and subjectively biased. Therefore, I focused the first part of my work on automated methods of template-based SSE annotation, as described in [10] and [26].

The SSE annotation can be useful even in families without traditional nomenclature. It provides the correspondence between the SSEs in the individual members of the family, which can be useful in applications such as structure comparison, generalizing the family anatomy, or function prediction. The problem is that the annotation template is not available in these families. For this reason, I also focused on the possibilities of automatic generation of the template [27]. My approach is to aggregate the information from all family members and create the secondary structure consensus of the family, which can serve as the template. Additionally, the consensus provides a concise overview of the general architecture of the family and highlights the conserved features as well as the variations.

One of the possible applications of the SSE consensus and annotations is in the field of protein visualization. Many tools for 2D visualization give very different diagrams for very similar structures. This can be improved using the SSE annotations, as described in [28]. Our approach (2DProts) has already been integrated into the CATH database.

This thesis is structured as follows: Chapter 2 provides the theoretical background of the area, Chapters 3 and 4 briefly describe the achieved results, followed by the full texts of the related publications in Chapter 5. During my PhD studies, I also participated in other projects outside the area of SSE annotation. Their publication outcome is summed up in Chapter 6.

Chapter 2

Theory and Methods

2.1 Protein secondary structure

A protein structure is often described at different levels of abstraction. The **primary structure** level describes the sequence in which the amino acid residues are bound in a protein chain. The **secondary structure** level focuses on the geometry of short segments of neighbouring residues. The **tertiary structure** level describes the spatial arrangement of the whole chain, and the **quaternary structure** describes how multiple chains are put together to compose a larger protein complex [29, 30]. The term supersecondary structure is often used to describe the intermediate levels of detail between the secondary and tertiary structure (motifs, folds, domains) [31].

Secondary structure describes the patterns of geometrical arrangement and atomic interactions within short segments of neighbouring residues. A segment that follows a specific pattern is called a **secondary structure element (SSE)**. We can coarsely divide SSEs into three types: helix, strand, and loop.

Helices and **strands** exhibit repetitive secondary structure patterns [32, 33]. From the **geometric point of view**, their α -carbon atoms are placed on a helical curve defined by several parameters. These parameters are radius r , pitch p (translation per turn), and number of residues per turn n ; however, sometimes alternative parameters are used: rise = p/n (translation per residue), twist = $360^\circ/n$ (rotation per residue). To allow such an arrangement, the dihedral angles ϕ and ψ must be confined into an area of values (characteristic for each type), which can be visualized in a Ramachandran plot [29, 34]. The typical parameter values for each secondary structure type are shown in Table 2.1. However, the real-life helices and strands always deviate from the ideal geometry to a smaller or larger extent.

Table 2.1: Comparison of the common secondary structure elements with repetitive pattern.

	Residues per turn (n)	Pitch (p) [Å]	Radius (r) [Å]	ϕ	ψ	Hydrogen bonds	Occurrence (residue %)
π -helix	4.4	5.0	2.8	-57°	-70°	$\text{NH}_{i+5} \rightarrow \text{OC}_i$	0.3%
α -helix	3.6	5.5	2.3	-57°	-47°	$\text{NH}_{i+4} \rightarrow \text{OC}_i$	31%
3_{10} -helix	3.0	6.0	2.1	-49°	-26°	$\text{NH}_{i+3} \rightarrow \text{OC}_i$	4%
β -strand parallel	2.0	6.4	1.0	-119°	113°	$\text{NH}_i \rightarrow \text{OC}_{K+i-1}$ $\text{CO}_i \leftarrow \text{HN}_{K+i+1}$	20%
β -strand antiparallel	2.0	6.8	1.0	-139°	135°	$\text{NH}_i \rightarrow \text{OC}_{K-i}$ $\text{CO}_i \leftarrow \text{HN}_{K-i}$	

Notes: The exact values of the geometrical parameters and occurrences vary from author to author. The values in the table are taken from [30] (n , p , ϕ , ψ), [37] (r), [38] (helix occurrences), and [35] (strand occurrence). For hydrogen bonds, i denotes the index of a residue involved in a hydrogen bond. In helices, all residues are involved ($i = m, m + 1, m + 2 \dots n$); in strands, only every other residue is involved ($i = m, m + 2, m + 4 \dots n$). K is a constant for a particular β -ladder. The hydrogen-bonding patterns are illustrated in Figure 2.1.

From the **physical point of view**, helices and strands are stabilized mainly by hydrogen bonds formed between the backbone atoms, namely between an amide hydrogen atom (donor) and a carbonyl oxygen atom (acceptor). In the case of **helices**, these bonds are formed between residues with a constant small sequential distance. This distance defines the particular type of the helix, namely α , 3_{10} , and π (scarcer types include the polyproline II helix [35] and other left-handed helices [36]). On the other hand, **strands** are stabilized by hydrogen bonds to other strands, which may be very distant in the sequence and can even be located on a different chain. The hydrogen-bonding patterns are illustrated in Table 2.1 and Figure 2.1. The set of hydrogen bonds connecting two strands is called a **β -ladder**. There are two possible arrangements of these bonds: in a parallel ladder, the two strands are oriented in the same direction; in an antiparallel ladder, the two strands are oriented in the opposite direction. Each strand can participate in more than one ladder (typically the even residues form one ladder, while the odd residues are oriented to the opposite side and form another ladder). A set of strands that are connected via ladders (a connected component) is called a **β -sheet**.

As we can see in Table 2.1, the 3_{10} -helix and π -helix are much less common than the α -helix. In fact, they rarely create a separate helix longer than a few residues.

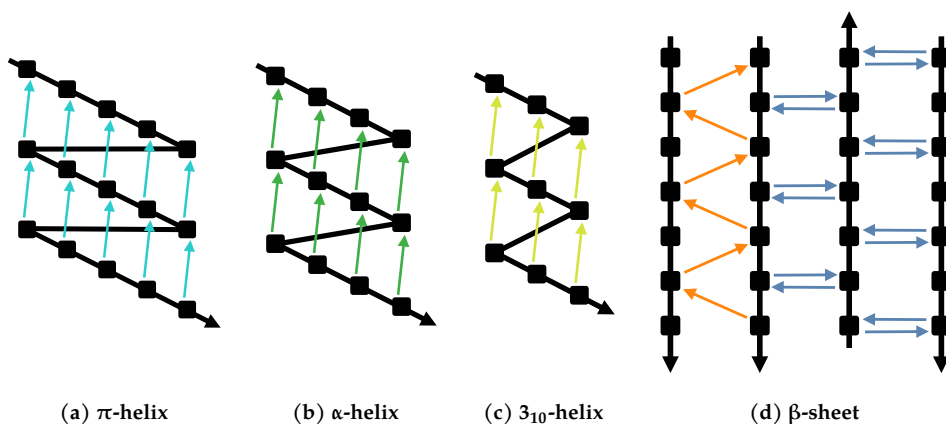


Figure 2.1: Regular hydrogen-bonding patterns in secondary structure elements: (a) π -helix; (b) α -helix; (c) 3_{10} -helix; (d) β -sheet consisting of four strands connected by a parallel ladder (orange) and two antiparallel ladders (blue). Residues are represented by black squares on the protein backbone (black line). Hydrogen bonds are represented by arrows pointing from the donor (provides an NH group) to the acceptor (provides a CO group).

Instead, they usually occur in combination with the α type, either at the ends of an α -helix or between two α -helices, together forming a longer helical segment [39, 40]. Visually, such a segment will be perceived as a single helix, and it is often useful to dismiss the detailed type distinction and understand it just as a single helix.

Segments that are neither helices nor strands are often loosely referred to as **loops** or **coils**. However, they can contain regular patterns and can be further classified: **Turns** are segments with a well-defined but non-repetitive secondary structure pattern. Their characteristic feature is the proximity of the first and last residue, and they typically occur between the helices and strands. Turns can be classified based on the number of residues (2–6) and their ϕ and ψ values. The **Ω -loops** are segments that have the first and the last residue in close proximity but lack a regular pattern in-between, often with a degree of flexibility. The term **random coil** is applied to segments without any regular pattern, with a high degree of flexibility [35].

Limitations: the above description of secondary structure assumes that the protein is fixed in a single conformation. However, it is known that proteins are dynamic objects, and in fact, conformational changes are crucial for the function of many proteins. This is especially true in the case of intrinsically disordered proteins and regions (IDP/IDR) [41]. It is important to keep in mind that the structural data we work with are just a model that represents one conformation of the protein (or an ensemble of conformations, in the case of NMR experiments).

Irregularities

Real-life SSEs sometimes depart from the simple description above. A **β -bulge** is a disruption of the repetitive hydrogen-bonding pattern in a β -ladder, formed by two or more residues on one strand (long side) opposite a single residue on the other strand (short side). It also affects the geometry of the strands. β -bulges are relatively frequent (on average two instances per protein) and occur primarily between antiparallel strands [42, 43].

A **helix kink** is a part of a helix that deviates noticeably from the ideal geometry and can suddenly change the direction of the helix (it often coincides with the presence of a proline residue, which disrupts the hydrogen-bonding pattern as it cannot be a donor of a hydrogen bond) [35]. As mentioned in the previous section, 3_{10} and π -helices often occur as parts of longer helical fragments; hence, we can look at them as irregularities within the standard α -helix pattern. A π -helix found within a longer α -helix (i.e. α - π - α) is named π -bulge or α -aneurism [35].

Helices and strands are also often curved [35]. Strands typically show a slight right-handed twist, resulting in a left-handed twist of the strand positions within the sheet [29]. All these irregularities are illustrated in Figures 2.2 and 2.3.

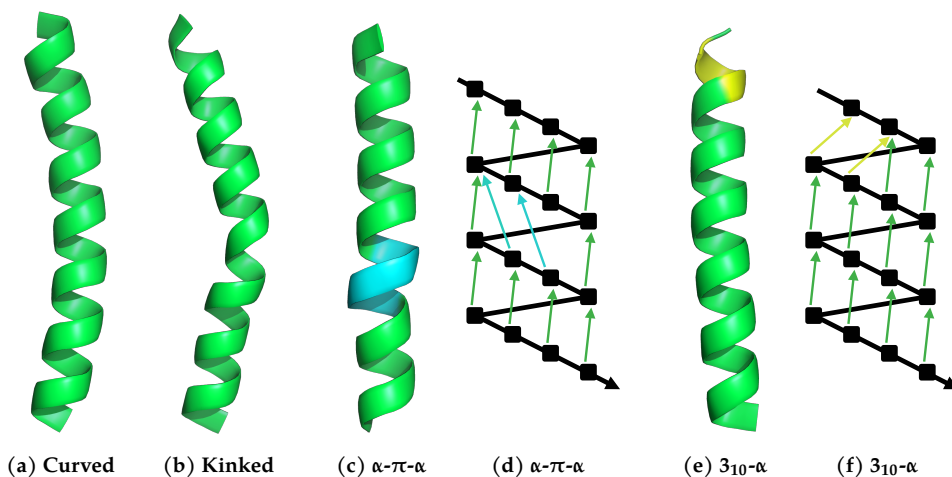


Figure 2.2: Irregularities in helices: (a) curved helix (PDB ID 1bgc, residues 143–172); (b) helix with a kink (PDB ID 1csc, residues 167–195); (c, d) π -helix as an irregularity within an α -helix (PDB ID 1c3w, residues 192–216); (e, f) 3_{10} -helix as an irregularity at the N-terminus of an α -helix (PDB ID 4rm4, residues 175–205).

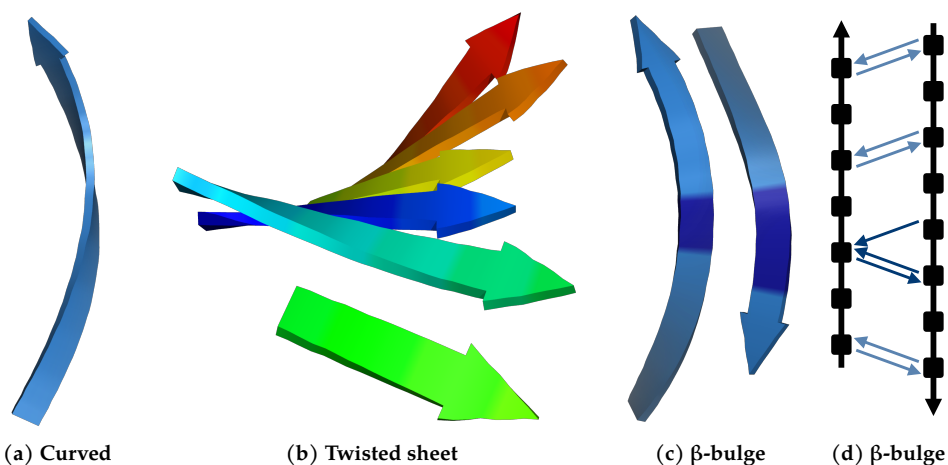


Figure 2.3: Irregularities in sheets: (a) curved (and twisted) strand (PDB ID 3kff, residues 51–60); (b) twisted sheet (PDB ID 1hdo); (c, d) sheet with a β -bulge (classic-type, PDB ID 9rub, residues 23–31 and 120–128, the β -bulge is highlighted in darker blue).

Secondary structure assignment

Secondary structure assignment (SSA) [10, 44] is the process of determining which segment of a protein structure forms what type of secondary structure element. The term SSA can also refer to the result of this process – a set of secondary structure elements and their β -connectivity (i.e. how the β -strands are connected via β -ladders).

Apart from the manual assignment, there are numerous automated methods of SSA. These methods differ mainly in their approach to the SSE definition – some focus on various geometrical features (distances, angles, dihedral angles, helical parameters, etc.), while others are based on the hydrogen-bonding patterns. Some methods provide only a coarse distinction of SSE type (helix, strand, loop), while others aim at a more detailed classification (α , 3_{10} , π -helix, etc.). They can also differ in their tolerance towards structural irregularities or treatment of the residues on the SSE boundary. Consequently, the results of these methods also differ. The reported residue-wise agreement between different methods ranges from 63% to 95% [44]. However, no SSA method can be considered the best one, and different methods may be appropriate for different applications.

Geometric methods include DEFINE [45], P-CURVE [46], P-SEA [47], PALSSE [48], STICK [49], XTLSSTR [50], KAKSI [44], SST [51], DISICL [52], ScrewFit [53], and methods used by Mitchell et al. [54] and Cao et al. [37]. **Hydrogen-bonding methods** include DSSP [39] and SECSTR [38]. Some tools,

such as STRIDE [55], combine both approaches; others use entirely different approaches (e.g. Voronoi contact maps in VoTAP [56]).

DSSP (Define Secondary Structure of Proteins) [39] is a well-established SSA method, based on hydrogen bond patterns. After many years it is still very popular [37, 44], and I adopted its main concepts in SecStrAnnotator [10], therefore I describe it in a little more detail. DSSP approximates the hydrogen bond energy with a simple formula and recognizes a hydrogen bond if its energy is below a certain threshold. Then it searches for the repetitive hydrogen-bonding patterns associated with each SSE type (α -helix (H), 3_{10} -helix (G), π -helix (I), and β -strand (E, B)). At least two hydrogen bonds are required in each pattern. The β -strand type is further split into two subtypes: β -bridge (B, two hydrogen bonds) and β -strand (E, three or more hydrogen bonds). The algorithm tolerates certain irregularities in the pattern (β -bulges). The last three recognized types are turn (T, a single helix-like hydrogen bond), bend (S, a region of high curvature without specific hydrogen bonds), and coil (-) [39].

Annotation

The term “annotation” is used to refer to any additional information associated with a protein structure or its part [57–59]. Examples include functional sites, channels, domains, ligand-binding sites, protein-protein interfaces, post-translational modification sites, structure validation issues, residue mapping to other databases, and more. A lot of effort has been invested to collect and integrate different types of annotations and make them easily accessible and searchable (PDBe-KB [57], SIFTS [58]).

By **secondary structure annotation** we understand assigning labels to the individual SSEs in the structure. Such annotation is commonly performed in well-studied protein families with many available structures with a common fold (but often large sequence variations), to allow comparison between the structures and provide a firm spot for describing the position of biologically relevant regions, such as active sites, binding sites, selectivity filters, channels, or protein-protein interfaces. Examples of families with a well-established SSE nomenclature include cytochromes P450 [14, 23], α/β -hydrolases [15, 17, 18], G-protein coupled receptors (GPCRs) [11], or immunoglobulins [12] (see Figure 2.4).

However, we can deduce from the literature that the SSE annotation is usually performed manually by the authors of the structure, based on the nomenclature which is unofficially accepted by the community around the family. There exists no

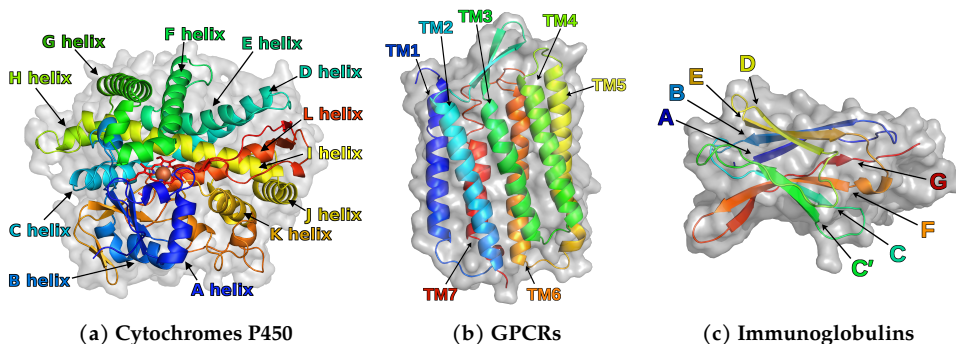


Figure 2.4: Examples of traditional annotations in protein families: (a) cytochromes P450 (PDB ID 2nnj, chain A, strands and minor helices are not labelled for clarity); (b) G-protein coupled receptors (PDB ID 5zim, chain A); (c) immunoglobulins (PDB ID 2mcp, domain H01).

widely recognized format or database for deposition of these annotations and they usually stay buried in the primary publications.

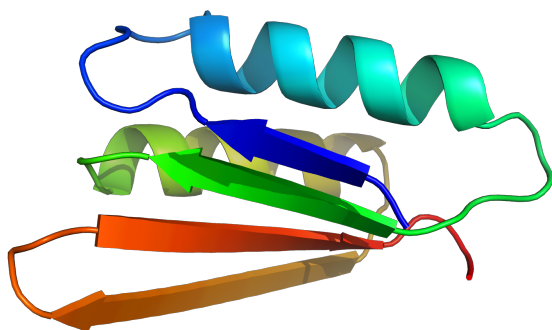
Furthermore, there can be a lot of inconsistency between the annotations produced by many different authors over many years (and sometimes even by the same author). Collecting the annotations from the literature or re-annotating the structures consistently can be extremely tedious when done by hand, and the results can be subjectively biased.

I am not aware of any automated tool designed specifically for SSE annotation prior to the publication of our tool SecStrAnnotator. Still, I should mention two tools that solve related problems, though they are not optimized for SSE annotation. The structure alignment tool SSM [60] is based on matching the SSEs of two input structures, which is in principle equivalent to template-based annotation (if one of the structures is understood as a template), but it is optimized mainly for structure alignment and comparison, not for producing the best annotations. Kocincová et al. [61] uses the concept of SSE matching for visual comparison of two structures in 1D; however, their algorithm is very simplistic, and its implementation is not provided.

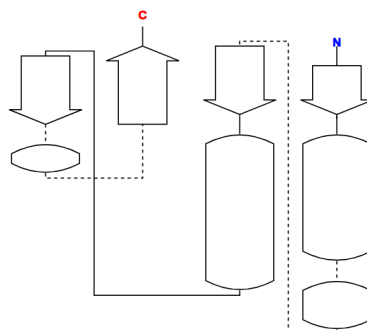
Visualization

The existing methods for automated visualization of the protein secondary structure use various approaches and differ in the level of detail (see Figure 2.5). The following text is a brief overview of the available methods, divided into three categories: 3D, 2D, and 1D methods.

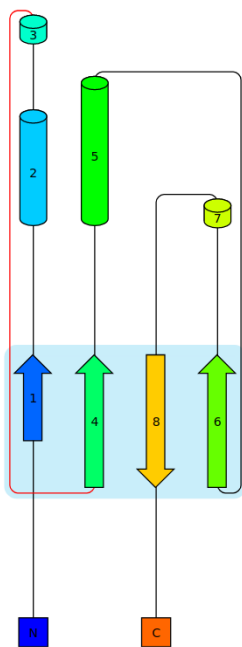
2. THEORY AND METHODS



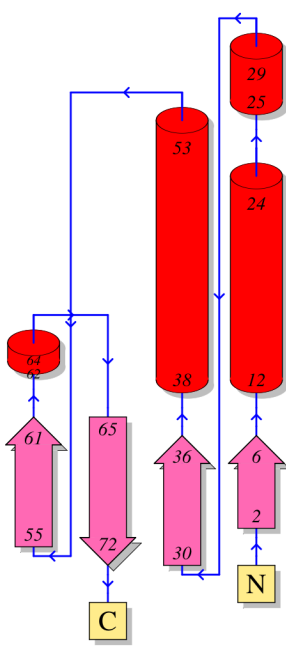
(a) Cartoon model in PyMOL



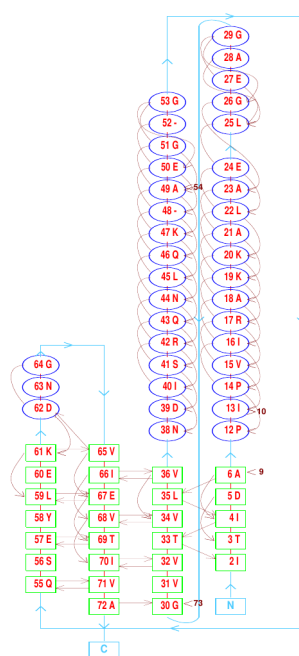
(b) PDB Topology Viewer



(c) Pro-origami

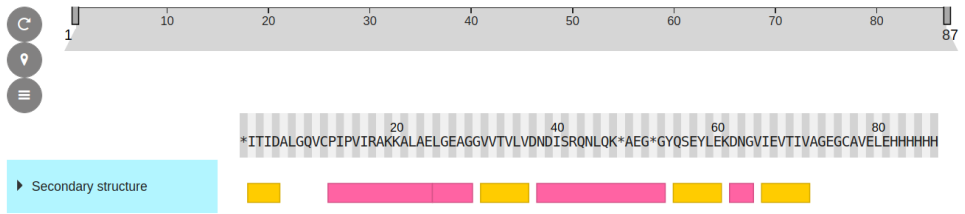


(d) HERA

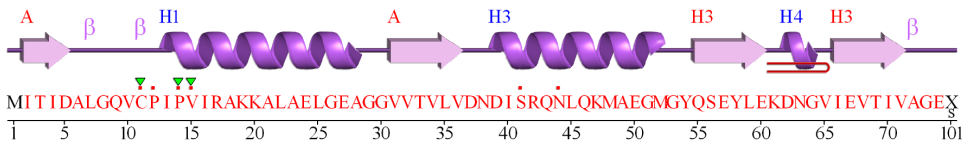


(e) HERA detailed diagram

2.1. Protein secondary structure



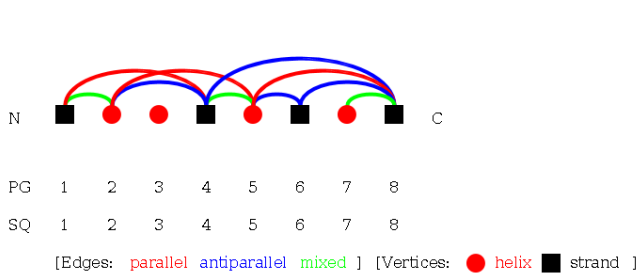
(f) ProtVista



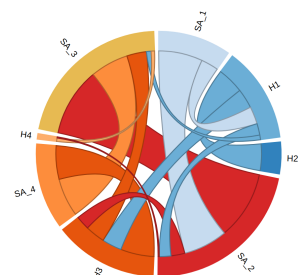
(g) PDBsum

RESNUM	2	12	22	32	42	54	64	74											
SEQ	ITIDALGQVCP P VIRAKKALAE LGEAGGV TVLVDND ISRQNLQKAE GGYQSEYLEK DNGV EVT IVAGE																		
CONSENSUS	EEEE	XT	00X	HHHHHHHHHHHH	TTTT	XXXX	00	EEEEEE	00										
DSSP	EEEE	0T	00T	HHHHHHHHHHHH	TTGG	00	EEEEEE	SS	HHHHHHHHHHHH	TT	0	EEEEEE	0	GGG	0	EEEEEE	0	00	
STRIDE	C	EEE	TTT	CC	HHHHHHHHHHHH	TTTT	C	EEEEEE	CC	HHHHHHHH	GGG	TT	EEEEEE	TTTT	EEEEEE	CCC			
STICKS	b	BBB	b	0000	a	AAAA	AAAA	AAAA	AAAA	a	0000	b	BBBB	b	0000	b	BBBB	b	0000

(h) 2Struc



(i) PTGL



(j) Protein Contact Atlas

Figure 2.5: Overview of (a) 3D, (b–e) 2D, and (f–j) 1D methods for protein secondary structure visualization, demonstrated on PDB entry 3hz7.

The most common **3D method** is the cartoon model (ribbon model), introduced by Richardson [62] and nowadays implemented in most 3D visualization programs. The cartoon model preserves the spatial arrangement of the protein, but it replaces the unnecessary details (coordinates of each atom) with 3D shapes representing the SSEs. Helices are typically shown as cylinders or as twisted helix-shaped strips; strands are shown as flat arrows. The β -connectivity is not shown explicitly but can be inferred from the position of the strands. For easier orientation, the direction of the chain can be expressed by colouring. The convention is to use the colours of the rainbow starting in blue (N-terminus) and ending in red (C-terminus). The main drawback of the 3D methods is the occlusion problem – the closer objects can hinder the farther objects. The 3D visualization programs manage the occlusion problem by interactivity (rotation, translation, zooming, clipping); however, in static visualization (e.g. in the figures in literature) the problem remains.

The **2D methods** represent the secondary structure in a 2D diagram, in which they try to include some of the spatial information or at least the β -connectivity. The 2D methods overcome the occlusion problem present in the 3D methods but often suffer from edge crossings and visual clutter [63]. Examples include PDB Topology Viewer [64], Pro-origami¹ [65], or TOPS² [66]. Some 2D visualization methods even provide a detailed overview of the hydrogen-bonding patterns (HERA² [67] and its successor PROMOTIF² [68]).

Perhaps the most trivial approach to the secondary structure visualization is the **1D approach**. In its simplest form, letters or shapes representing helices and strands are placed above or below the amino acid sequence. We can find examples of such visualization in the ProtVista viewer [69] (integrated into the PDB web), in PDBsum [70], 2Struc [71], and in many figures in scientific publications. More sophisticated methods aim at a detailed description of the β -connectivity or other interactions, such as PTGL [72] or the chord plot in Protein Contacts Atlas [73]. The drawback of the 1D approach is the complete lack of spatial information.

¹Currently available at <http://munk.cis.unimelb.edu.au/pro-origami/>.

²These programs are not available anymore. Precomputed HERA diagrams are available in PDBsum.

2.2 Classification of protein structures

Molecular evolution has given rise to a plethora of different proteins. Proteins with a common ancestry (homologs) typically have a similar sequence, structure, and function. A **superfamily** [5, 74, 75] is the largest group of proteins for which homology can be inferred. Since the sequence changes much faster with evolution than the structure, distant homologs can have very low sequence similarity while having a similar conserved structure. For this reason, structural information can be a valuable indicator of homology. However, this is complicated by the fact that unrelated proteins can also evolve into a similar structural organization (analogs). Structure similarity can be assessed by different approaches, from a simple comparison of secondary structure content to alignment-based measures like RMSD [76], Q-score [60], or TM-score [77].

On the highest level, protein structures can be divided into four broad **classes** introduced by Levitt and Chothia [78]:

- **all- α** (structures essentially formed by helices),
- **all- β** (structures essentially formed by strands),
- **α/β** (interspersed helices and strands),
- **$\alpha+\beta$** (segregated α and β regions).

Richardson [62] soon elaborated this division into a classification of major structural patterns – folds. The term “**fold**” usually refers to a particular arrangement of secondary structures, or to a set of structural domains that share such an arrangement.

Many protein chains are composed of multiple **domains** (multi-domain proteins) [79, 80] and most classifications apply to these individual domains rather than to whole protein chains. However, the term “domain” is used somewhat ambiguously in the literature, as pointed out also by Schaeffer [79] and others [30, 31]. Some definitions lean on the structural aspect (independent globular region / large subassembly that would be stable if cleaved from the rest of the structure [78] / the smallest cooperatively folding unit [79]). Other definitions of a domain are based on conserved function or sequence (unit of structure with a specific function that is found in diverse proteins [31]). In general, these definitions largely overlap and can even be combined (the pieces that could be expected to be stable as independent units or are analogous to other complete structures [62] / I know it when I see it [30]).

Several databases have been established to provide a rigorous classification of protein structural domains.

SCOP

The SCOP database (Structural Classification of Proteins) [5] was based on a hierarchical classification model. The highest level, **Class**, copies the traditional distinction: α , β , α/β , $\alpha+\beta$, and an additional class “Multi-domain” (a few more classes were added later). The **Fold** level groups structures with the same arrangement and topological connections of the major SSEs, without the necessity for a common evolutionary origin. The **Superfamily** level suggests a probable common origin, based on structure similarity. The **Family** level implies a clear common origin, based on sequence similarity or very good structure similarity. The two lowest levels are **Protein** (all isoforms and orthologs of a protein) and **Species** (structures of a specific isoform from a specific organism, including its artificial mutations).

Around 2012, SCOP split into two projects maintained by independent groups – SCOPe [81, 82] and SCOP2 [7, 83].

SCOPe (SCOP–extended) [81, 82] preserved the original SCOP hierarchy while improving the automatic classification algorithm and incorporated data from the related ASTRAL database [84]. An automated domain prediction and classification algorithm is run on each chain; if the classification confidence is high, it is classified automatically; otherwise, it is manually inspected. The hierarchy above the Species level is curated manually.

SCOP2 prototype [83] aimed to eliminate some pathologies in the SCOP design and capture the new discoveries in protein evolution. They decided to separate the structural and evolutionary relationships and replace the hierarchical tree-like model with a more flexible directed acyclic graph model (DAG), where a node can have multiple parents and levels can be skipped. However, the newest release of **SCOP2** [7] returns to the hierarchical model and achieves the separation of structural and evolutionary relationships by two explicit exceptions. First, a family can belong to a different fold than its superfamily (divergent evolution). Second, a family can contain a conserved combination of two or more structural domains from distinct superfamilies (due to this, domain boundaries can be defined differently on the family and superfamily level). To capture non-globular protein structures, Folds are complemented with IUPRs (Intrinsically Unstructured Protein Regions). Folds and IUPRs are also grouped into Protein types (soluble, membrane, fibrous, and intrinsically disordered), independently from their Class.

CATH

The CATH database [6, 75] was established shortly after SCOP. In contrast to SCOP, it aimed to provide classification procedures requiring minimal human intervention, and thus maximum objectivity. It is also based on a hierarchical classification. The highest level, **Class**, divides protein domains according to their SSE composition: “Mainly alpha”, “Mainly beta”, “Alpha Beta” (merges α/β and $\alpha+\beta$, as the Class level is agnostic to the sequential order of the SSEs), and two additional classes “Few Secondary Structures” and “Special” (non-globular proteins). The **Architecture** level reflects the gross arrangement of the SSEs regardless of their number and order. The **Topology** level also includes the SSE order (connections via sequence) and roughly corresponds to the separation into folds. The **Homologous superfamily** level groups domains with sufficient evidence for a common evolutionary origin. The lowest levels (**S35**, **S95**, **S100**) are based on sequence similarity rather than structure (they are clusters with sequence identity 35%, 95%, and 100%, respectively).

The classification is performed in a bottom-up manner with a high degree of automation, except for the Architecture level, which was arranged manually based on the common knowledge and literature. However, the Architectures improve the comprehensibility of the whole hierarchy. The newer features of CATH include the classification of protein sequences without known structure (via hidden Markov models [85]) and separation into FunFams (functional families [86]). The approximate correspondence of the SCOP and CATH hierarchy levels is shown in Table 2.2.

ECOD

The ECOD database [87] aims to accent the evolutionary relationships rather than structural fold and to group also distantly related homologs. To do this, it reversed the order of homology- and topology-based levels.

The highest level is **Architecture**, based on similar SSE composition and shapes. The **X level** is defined as possible homology (insufficient evidence for homology) and roughly corresponds to SCOP Folds. The **H level** (homology) groups domains with common ancestry but which can have slightly different topologies. The **T level** (topology) groups homologous domains with similar topological connections. The **F level** (family) is defined by significant sequence similarity.

The classification is based on SCOP but merges some superfamilies into a common X-group or H-group and classifies previously unclassified domains using a variety of structure- and sequence-based scores, literature evidence, and manual

inspection. Special architectures cover the domains that are very hard to classify (coiled-coils, peptides, fragments, largely disordered structures, and low-resolution structures). ECOD covers the whole PDB database and new structures are added weekly.

Table 2.2: Comparison of the classification hierarchies in SCOP2, CATH, and ECOD, including the total number of nodes at each level (based on SCOP 2022-06-29, CATH v4.3, ECOD 20220613).

SCOP2		CATH		ECOD	
Class	5	Class (C)	6		
		Architecture (A)	43	Architecture	20
Fold	1562	Topology (T)	1472	X	2460
				H	3715
Superfamily	2816			T	3950
Family	5936	Superfamily (H)	6631	F	15311
Protein	NA	S35	32388		
		S95	62915		
Species	36900	S100	122727*		
Domain	861631	Domain	500238	Domain	898380

* The number of the S100 nodes in CATH is significantly higher than the Species nodes in SCOP, despite they should be approximately equivalent. This is because the Species are based on a unique UniProt ID while the S100 clusters are based on 100% sequence identity (i.e. they distinguish mutations).

Chapter 3

Synopsis of the Results

This thesis is focused on the annotation of SSEs in protein structures and its application. The results are presented in four main publications (one book chapter and three articles in peer-reviewed scientific journals).

The work started with the development of SecStrAnnotator, a tool for template-based annotation of SSEs in protein structures. After successful implementation and testing, we published this tool in a book chapter [10]:

Midlik,A., Hutařová Vařeková,I., Hutař,J., Moturu,T.R., Navrátilová,V., Koča, J., Berka,K., Svobodová Vařeková,R. (2019) Automated family-wide annotation of secondary structure elements. In Kister,A.E. (ed.), *Protein Super-secondary Structures*. Humana Press, New York, NY. Vol. 1958, pp. 47–71. https://doi.org/10.1007/978-1-4939-9161-7_3.

Then, using this new tool, we analysed in detail the SSEs in our use-case protein family, cytochromes P450. We presented the results in the following article [26]:

Midlik,A., Navrátilová,V., Moturu,T.R., Koča,J., Svobodová,R., Berka,K. (2021) Uncovering of cytochrome P450 anatomy by SecStrAnnotator. *Scientific Reports*, **11**, 12345. <https://doi.org/10.1038/s41598-021-91494-8>.

However, SecStrAnnotator still had to rely on an annotation template manually prepared by the user, and this limited its wide-range application. Therefore we focused on the possibility of automatic template generation. The resulting software, OverProt, creates a secondary structure consensus, which can serve as a template, but also provides a valuable overview of the family as a whole. OverProt allowed us to

create secondary structure consensuses for all protein families in the CATH database, annotate all their members, and create the OverProt database. We published the software and the database in this article [27]:

Midlik,A., Hutařová Vařeková,I., Hutař,J., Chareshneu,A., Berka,K., Svobodová,R. (2022) OverProt: secondary structure consensus for protein families. *Bioinformatics*, (in press). <https://doi.org/10.1093/bioinformatics/btac384>.

The last paper focuses on an application of the SSE annotations in the field of visualization. Though there were several tools for visualizing protein structures by 2D diagrams, none of them took protein similarity into account. Therefore, very similar proteins could end up with very dissimilar diagrams, which complicated structure comparison. To address this, we implemented a new 2D visualization tool, 2DProts, which uses the SSE annotations provided by OverProt and SecStrAnnotator to ensure that the similarity in 3D is reflected in the 2D diagrams. We published 2DProts in this article [28]:

Hutařová Vařeková,I., Hutař,J., Midlik,A., Horský,V., Hladká,E., Svobodová,R., Berka,K. (2021) 2DProts: database of family-wide protein secondary structure diagrams. *Bioinformatics*, **37**, 4599–4601. <https://doi.org/10.1093/bioinformatics/btab505>.

The following sections provide a short summary of these four publications. Their full texts are available in Chapter 5.

3.1 Automated family-wide annotation of secondary structure elements

Protein structures can be classified into protein families based on their similarity and common evolutionary origin. The numbers of known structures in these families are continuously growing, and we are now able to study them systematically. To simplify orientation within the piles of structures belonging to our family of interest, the annotation of the SSEs can be very useful. This is witnessed by the communities around some protein families, like cytochromes P450, who routinely annotate the SSEs in the structures according to an established nomenclature and use these annotations to describe and compare the observed structures. They also use the

annotated SSEs as reference points to describe the position of biologically relevant regions like catalytic sites, channels, or protein-protein interfaces.

However, the annotation was traditionally performed by hand, which is very time-consuming and error-prone. To address this issue, we started developing SecStrAnnotator, a program for automatic template-based annotation of SSEs. Its input consists of two protein structures: an annotated “template” protein and a “query” protein that is to be annotated. These two proteins (or protein domains) must of course be structurally similar (e.g. members of the same protein family). The initial design of the program involved three steps: first, rotate and move the query protein to fit on the template (structural alignment); second, detect the SSEs in the query protein (secondary structure assignment); and third, match the template SSEs with the query SSEs while maximizing annotation score (matching).

I implemented the first version of SecStrAnnotator in programming language C#, using the simplest available method for each step. Then I gradually improved each step, evaluating the quality of the results on our sample family of cytochromes P450 and later on more families. As an example, the secondary structure assignment step was originally performed by a well-established program DSSP [39]; however, it proved inappropriate for the annotation purposes, as it often broke helices into smaller pieces. Therefore I had to design a new method that better reflects our intuitive perception of helices. Similarly, the original greedy algorithm for SSE matching produced many wrong annotations, so I replaced it with a dynamic programming algorithm (DP) [88]. The DP algorithm provided much better results but still did not take the β -connectivity into account, so I had to reformulate the annotation score function and introduce a new algorithm named “mixed ordered matching” (MOM).

Later I created additional Python scripts to automate the workflow for annotation of a whole protein family (get the list of family members, download structural data, etc.). The bottleneck of the whole procedure remained the preparation of the annotation template, which still had to be done manually.

In this article, we provided the instructions for the whole annotation workflow, including the selection of a template protein and preparation of the template annotation. We also described the principles of the algorithms used in SecStrAnnotator. The software is freely available at <https://webchem.ncbr.muni.cz/Wiki/SecStrAnnotator>.

My contribution: I developed and tested the whole SecStrAnnotator software. I wrote most of the manuscript and prepared the figures and tables.

3.2 Uncovering of cytochrome P450 anatomy by SecStrAnnotator

Having developed a tool for automated SSE annotation, we aimed to provide a use case demonstrating its application. Namely, we focused on the family of cytochromes P450 (CYPs), a wide family of biologically interesting enzymes present in all domains of life, and elaborated a detailed analysis of their SSEs.

The annotation of SSEs in CYPs is well-established and used routinely, though there is no officially accepted SSE nomenclature. Rather, the information about the SSE labels is scattered in the multitude of publications (especially for those SSEs that are not present in every structure), and there are even some inconsistencies. Therefore, we performed a literature review to summarize how the authors label the SSEs, and we created a consensus that could be used as an annotation template.

Then we annotated the whole CYP family and analysed each SSE in terms of occurrence, typical length, and amino acid sequence. We constructed sequence logos for each SSE and used them to create a generic residue numbering scheme, inspired by the schemes used for other protein families. We also described the differences between bacterial and eukaryotic CYPs and identified a small group of anomalous bacterial CYPs. In the supplementary material, we presented an analysis of secondary structure irregularities, perhaps the most interesting being a conserved π -helix located within the helix E.

We also accomplished a few improvements in the SecStrAnnotator software, namely switching from the PDB file format to the modern mmCIF format, detection of irregularities, automation of the whole SSE annotation and analysis pipeline (SecStrAnnotator Suite), visualization of the existing annotations via PyMOL plugin and SecStrAPI. Furthermore, we created the online version of SecStrAnnotator (<https://sestra.ncbr.muni.cz>).

My contribution: I developed the new SecStrAnnotator features, summarized the SSE nomenclature for CYPs from the literature, performed the analyses of SSEs in CYPs and participated in their interpretation, co-wrote the manuscript, and prepared most of the figures.

3.3 OverProt: secondary structure consensus for protein families

As mentioned before, the bottleneck of the SSE annotation by SecStrAnnotator was the manual preparation of the annotation template for the family of interest. To circumvent this, we focused on the possibility of automatic template generation.

Our idea was to extract the characteristic features of all the family members and create a **secondary structure consensus** – a set of consensus SSEs, each of which is characterized by its type (helix/strand), occurrence (percentage of structures it is present in), average length, average 3D position and its variability, etc. This secondary structure consensus would then be used as an annotation template.

Even though the number, order, and spatial arrangement of the SSEs are relatively consistent in each family, there is usually some variation – the exact length and 3D position of each SSE varies from structure to structure; some SSEs are missing in some structures; others appear only in a small fraction of the structures. Therefore creating a consensus is not straightforward.

There is an analogy between the worlds of primary structure (sequence) and secondary structure. A family of homologous sequences can be processed by multiple sequence alignment [89] to get a consensus sequence, visualized by a sequence logo [90]. This well-established method shows the essential features of the family and highlights the similarities and differences within the family. However, in the world of secondary structure, a method of creating and visualizing the consensus was not available before our tool OverProt (see Figure 3.1).

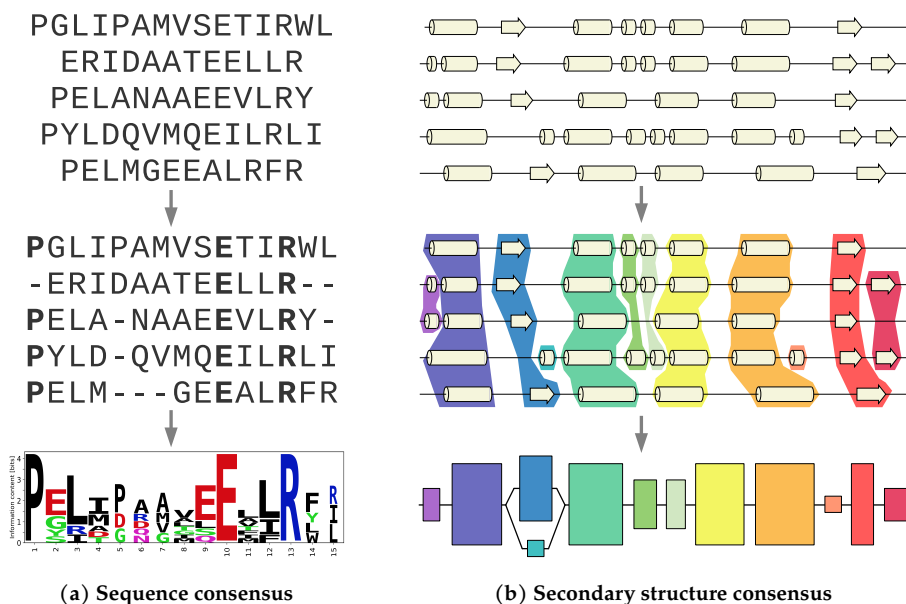


Figure 3.1: Analogy between the family consensus on the (a) primary and (b) secondary structure level (simplified to one dimension).

Our approach to this problem was to take the SSEs from all family members and apply a clustering algorithm to them. Topologically equivalent SSEs from different members would cluster together, and then each of the resulting clusters would correspond to a consensus SSE (and thus to an SSE label for annotation). Some additional constraints would have to be applied to the clustering, e.g. a cluster must not contain more than one SSE from the same family member. We kept this original design, yet it took several attempts to choose a clustering algorithm and apply it in such a way that would provide good enough results even in the problematic structurally diverse families and at the same time could handle even the largest protein families with reasonable time and memory requirements. We also developed a visualization of the results of the clustering, showing the order of the consensus SSEs, their occurrence (cluster size), typical SSE length, and β -connectivity. We realized that such a visualization gives us a valuable overview of the general architecture of the whole family, which could never be gained from one family member alone. Hence we named this new tool “OverProt” (**overview of protein family**).

We were then able to run OverProt on every protein family listed in the CATH database and make the results available in the form of a web server (we chose CATH over other databases because of its popularity, focus on structural similarity, and easy access to its data via SIFTS in PDB API). In this way, the users do not need to install the tool but can view precomputed results comfortably. We also added the possibility to upload a user-defined family (a list of protein domains) and wait for the results to be computed on our server. We created OverProt Viewer to visualize the results interactively rather than as a static image. The server also provides SSE annotations, produced by SecStrAnnotator based on the OverProt annotation template, thus we have annotations available for each protein domain in CATH. The disadvantage of these annotations is the use of generic SSE labels (H0, E1, E2...) unlike more human-friendly labels from manual annotations (e.g. helices A, B, C..., sheets β 1, β 2..., strands within sheets β 1-1, β 1-2...). OverProt is freely available at <https://overprot.ncbr.muni.cz>.

My contribution: I designed, implemented, and tested the OverProt software, including OverProt Viewer and OverProt Server. I created the database of precomputed results, and I maintain the web server. I co-wrote the manuscript and prepared the images.

3.4 2DProts: database of family-wide protein secondary structure diagrams

Before we created 2DProts, there were several tools that visualized the SSEs in protein structures by 2D diagrams. However, most of these did not reflect the 3D arrangement of the SSEs, so two SSEs that were next to each other in the 3D structure could be placed very far from each other in the 2D diagram. Furthermore, none of these tools took protein similarity into account, so two very similar proteins could have very dissimilar 2D diagrams. Thus, such diagrams provided only limited structural information and were practically useless for structure comparison.

This motivated us to develop 2DProts, a tool for generating 2D diagrams which would focus on preserving the structural information from 3D and respecting the similarity between the proteins within a protein family. The first goal was realized by defining an error function of a projection from 3D to 2D and minimizing its value. The latter goal was achieved by selecting a “start domain” in each family and penalizing the deviation of each diagram from the start domain diagram. However, to calculate the deviation, we must know the correspondence between the SSEs from different protein domains. This is where we utilized the SSE annotations provided by OverProt together with SecStrAnnotator – OverProt creates the secondary structure consensus which is then used by SecStrAnnotator as a template to annotate all family members. The resulting annotations provide the needed SSE correspondence.

Using the 2DProts tool, we constructed a database of protein secondary structure diagrams. The database contains an individual diagram for each protein domain listed in CATH but also a multiple diagram for each protein family. The multiple diagram shows the general SSE arrangement of the whole family and the individual variations (therefore we show the 2DProts diagrams also on the OverProt web). The 2DProts diagrams have also been incorporated into the CATH web itself. 2DProts is freely available at <https://2dprots.ncbr.muni.cz>.

My contribution: I participated in the design of 2DProts and discussed the results of the intermediate versions and possible improvements. I also helped to solve some problems that arose during the development, and I implemented small parts of the software. I performed some changes in OverProt and SecStrAnnotator to allow their integration with 2DProts. I participated in writing the manuscript and prepared some of the figures for the supplementary material.

Chapter 4

Conclusion

Thanks to the structure determination techniques, the number of available protein structures is constantly growing. These structures can be classified into protein families based on their similarity and common evolutionary origin. Each family also has a set of characteristic secondary structure elements (SSEs) with a relatively consistent arrangement. When properly annotated, these SSEs can simplify orientation within the structures, they can serve as reference points for describing and comparing these structures and locating their key regions, and ultimately help us understand the function of these protein structures.

This thesis addressed several questions related to the SSEs and mainly their annotation. First, we created SecStrAnnotator, a software tool for automatic template-based annotation of SSEs in protein structures. SecStrAnnotator uses a user-provided annotation of a “template” protein structure to annotate a structurally similar “query” protein structure. It can also be easily applied to annotate whole protein families.

Then we developed a workflow for detailed analysis of SSEs in a protein family and demonstrated it on cytochromes P450, a family of important biotransformation enzymes. This analysis reports occurrence, typical length, and amino acid sequence of each SSE and can be used to reveal differences between the subgroups of the family, to discover conserved structural irregularities, or to establish generic residue numbering schemes.

The other major topic of this thesis was the construction of secondary structure consensus for protein families, which provides an overview of the family anatomy. Here we contributed by developing OverProt, a tool for creating the secondary structure consensus for a given protein family. This consensus can also serve as an

4. CONCLUSION

annotation template, so the combination of OverProt and SecStrAnnotator allows us to automatically annotate all protein domains in all families without the need for manually created annotation templates.

The last part was focused on an application of the SSE annotations in the field of protein structure visualization. Our tool 2DProts creates 2D diagrams showing the SSEs in a protein structure. Contrary to other similar tools, 2DProts aims to reflect the 3D arrangement of the SSEs and considers structure similarity, so that proteins from the same family have comparable 2D diagrams.

All these tools are freely available as desktop programs and as web servers. In the case of OverProt and 2DProts, we provide databases of precomputed results covering all protein families listed in the CATH database.

The presented results were published in a book chapter and three articles in peer-reviewed scientific journals.

Chapter 5

Main Publications

Automated family-wide annotation of secondary structure elements

Adam Midlik^{1,2}, Ivana Hutařová Vařeková^{1,2,3}, Jan Hutař^{1,2}, Taraka Ramji Moturu¹, Veronika Navrátilová⁴, Jaroslav Koča^{1,2}, Karel Berka⁴, Radka Svobodová Vařeková^{1,2}

¹ CEITEC – Central European Institute of Technology, Masaryk University, Kamenice 5, 625 00 Brno, Czech Republic

² National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Kamenice 5, 625 00 Brno, Czech Republic

³ Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic

⁴ Department of Physical Chemistry, Regional Centre of Advanced Technologies and Materials, Faculty of Science, Palacký University, 17. listopadu 1192/12, 771 46 Olomouc, Czech Republic

In Kister, A.E. (ed.), *Protein Supersecondary Structures*.

Humana Press, New York, NY. Vol. 1958, pp. 47–71, **2019**.

https://doi.org/10.1007/978-1-4939-9161-7_3



Chapter 3

Automated Family-Wide Annotation of Secondary Structure Elements

Adam Midlik, Ivana Hutařová Vařeková, Jan Hutař, Taraka Ramji Moturu, Veronika Navrátilová, Jaroslav Koča, Karel Berka, and Radka Svobodová Vařeková

Abstract

Secondary structure elements (SSEs) are inherent parts of protein structures, and their arrangement is characteristic for each protein family. Therefore, annotation of SSEs can facilitate orientation in the vast number of homologous structures which is now available for many protein families. It also provides a way to identify and annotate the key regions, like active sites and channels, and subsequently answer the key research questions, such as understanding of molecular function and its variability.

This chapter introduces the concept of SSE annotation and describes the workflow for obtaining SSE annotation for the members of a selected protein family using program SecStrAnnotator.

Key words Annotation, Secondary structure, Secondary structure elements, Protein family, Protein domain, SecStrAnnotator, Structural alignment, Secondary structure assignment

1 Introduction

1.1 Background and Motivation

Protein structural data represent a highly valuable source of information, and important research results have been discovered based on them. All the data (currently ~140,000 entries) are accessible to the research community via Protein Data Bank [1], and the number of structures is continuously growing.

In the past, the newly determined structures differed from other available proteins, because only a few isolated islands in the chemical space of proteins were mapped. With the increasing number of known structures, protein families started to emerge, consisting of structurally and functionally similar proteins. Nowadays, more and more structures (which originate from various organisms, contain different ligands, or have various mutations) are being collected in each family. This trend is nicely demonstrated on five different protein families, mentioned in Table 1.

Table 1
Number of biomacromolecular structures in Protein Data Bank for selected protein families

Protein family	CATH code	Number of PDB entries in years			
		1990	2000	2010	2018
Globins	1.10.490.10	40	319	797	1090
Cytochrome P450	1.10.630.10	2	59	376	728
NADP-dependent oxidoreductase	3.20.20.100	0	21	197	353
Apoptosis regulator Bcl-2	1.10.437.10	0	7	79	133
Bulb-type lectin	2.90.10.10	0	7	16	30

To characterize these families, several databases focused on classification of protein structures based on their similarity have been developed (CATH [2], SCOPe [3]). With the vast amount of structural data about each protein family, the systematic study of larger datasets, as opposed to the study of individual structures, is gaining importance. Based on these data, it is possible to reach interesting and important research results—from understanding biomacromolecular functions and mechanisms of their action to the classification of types of diseases or the rational development of novel drugs.

But such studies can hardly rely on bare structural data; an additional layer of information is necessary. This new layer is *annotation*—assigning a name or any potentially biologically relevant information to a structure or its part. The annotated part can range from a single atom or residue through a secondary structure element or a functionally important region to a whole protein.

This chapter will focus on the annotation of secondary structure elements (SSEs). Each protein family has a set of characteristic SSEs with a well-defined arrangement, which is consistent even if the proteins originate from different species or perform different functions. Hence, the SSEs can serve as landmarks which enable easier orientation in the protein structures.

Annotation of SSEs has a long tradition in some protein families. For example, the nomenclature of helices and sheets in cytochrome P450 (CYP) family is well established [4, 5] and proves to be particularly useful when comparing existing structures, describing new ones, or generalizing observations over the whole family (*see* Fig. 1a). Furthermore, SSEs can be used as a reference to describe the position of other key regions, such as catalytic sites, channels, or protein-protein interfaces. A nice illustrative example is again the CYP family, with a well-established classification of multiple different channels based on their position relative to the traditionally named SSEs [6] (*see* Fig. 1b).

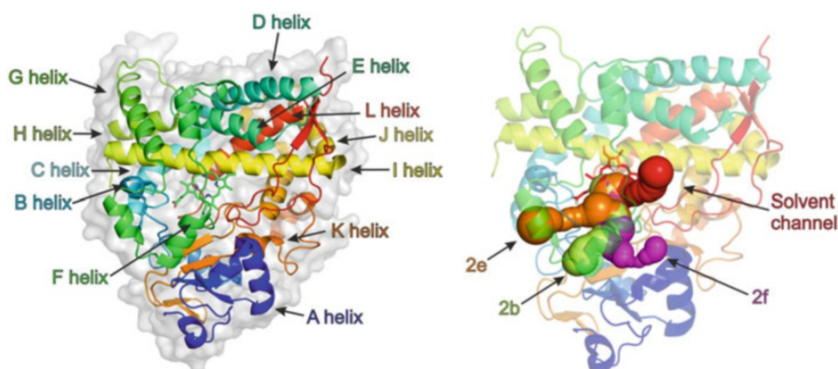


Fig. 1 Illustrative annotation of (a) secondary structure elements and (b) channels in a member of CYP family

Even in families that have no such traditional nomenclature, annotation of SSEs can still be valuable, because it provides the correspondence between the SSEs of individual members of the family—mutually corresponding SSEs are simply annotated by the same name, albeit arbitrarily created. This makes it possible to study the SSEs in the context of the family and describe the general anatomy of the family—the occurrence of the individual SSEs, their typical length, position, amino acid composition, and the variability of these properties and their relation to the function.

Visualization of protein structures can also benefit from SSE annotation. Currently available tools for the generation of 2D topology diagrams (e.g., HERA [7], PROMOTIF [8], Pro-origami [9]) treat each structure separately and do not take protein similarity into account. As a result, two structurally similar proteins (or even two structures of the same protein) can yield entirely dissimilar diagrams. SSE annotation can be used to modify the generation of topology diagrams in such way that the resulting diagrams would place conserved SSEs to similar positions within the whole family [10]. This highlights the parts of each structure which diverge from the general anatomy.

In this chapter, we describe methods for automated annotation of SSEs in protein structures. Our approach is template-based, meaning that a template annotation of one protein from the family is provided to the algorithm. The overall procedure therefore consists of three main stages: preparing the data for a selected protein family, preparing the template annotation, and running the annotation algorithm on each member of the family.

1.2 Terminology

To avoid later confusion, we provide a summary of the basic terms that will be used throughout the text.

Secondary structure element (SSE) is a contiguous region of a protein chain exhibiting some secondary structure pattern. SSEs can be coarsely divided into three types: *helix*, *β -strand* (or simply

strand), and *loop*. In this work, we focus only on the first two types. Further classification into different subtypes (α -helix, β_{10} -helix, etc.) is not necessary for the annotation purposes. SSE types are typically abbreviated as H (helix) and E (strand).

Each SSE within a structure can be identified by its position—*chain* identifier, *start* (index of its first residue), and *end* (index of its last residue)—and its *type*.

The situation gets more complicated in the case of β -strands because they are mutually connected via hydrogen bonds—we call this *β -connectivity*. The term *β -ladder* refers to a set of backbone-backbone hydrogen bonds between two particular β -strands. A single hydrogen bond is not considered a β -ladder. Each β -ladder can be classified as either *parallel* or *antiparallel*, based on the relative orientation of the two β -strands.

We define *β -graph* as an undirected edge-labelled graph whose vertices correspond to the β -strands in a structure and edges correspond to the β -ladders. The label of each edge denotes the type of the β -ladder (parallel or antiparallel).

The term *β -sheet* refers to a set of β -strands which are connected by β -ladders. Using the notion of β -graph, a β -sheet is defined as a connected component in β -graph. A β -sheet can contain β -strands from more than one chain.

Secondary structure assignment (SSA) consists of the set of SSEs found in a protein structure (each one described by its chain, start, end, and type) and optionally the β -graph. SSA can also refer to the process by which the SSEs and the β -graph are found.

Secondary structure annotation is assignment of names to some (or all) SSEs in a protein structure.

Protein family is a set of structurally similar *protein domains*. Each of these domains can be either a whole protein chain or only its part (in multidomain proteins).

2 Materials

2.1 Databases

1. **PDBe**: Protein Data Bank in Europe (PDBe) is one of the members of the Worldwide Protein Data Bank (wwPDB) [11] which maintains and provides access to the global repository of macromolecular structure models, the Protein Data Bank (PDB). Apart from the access to the structural data, PDBe provides a range of related services and tools. SIFTS (structure integration with function, taxonomy, and sequence [12]) provides cross-references to other biological databases, such as UniProt, CATH, or Pfam. PDBe REST API is a programmatic way to obtain information from the PDBe services.

<http://www.ebi.ac.uk/pdbe/>

2. **CATH**: The CATH database [2] provides structure-based hierarchical classification of protein domains found in the protein structures from PDB. CATH uses a four-level structural hierarchy, whose bottom level, homologous superfamily, corresponds to a demonstrable evolutionary relationship between domains.

<http://www.cathdb.info/>

3. **Pfam**: The Pfam database [13] classifies protein domains into families based on their sequence similarity. Each protein family is represented by a multiple sequence alignment and a hidden Markov model (HMM).

<https://pfam.xfam.org/>

2.2 Tools

1. **PyMOL**: PyMOL [14] is a commonly used molecular visualization tool. It is typically operated from graphical user interface (GUI), but it also supports interpretation of scripts from command line, without GUI. Besides other functionality, it provides commands for structural alignment and superimposition.

<https://pymol.org/>

2. **DSSP**: Define secondary structure of proteins (DSSP) [15] is a well-established algorithm for secondary structure assignment based on hydrogen bond patterns.

<https://swift.cmbi.umcn.nl/gv/dssp/index.html>

3 Methods

In this section, we will describe all steps which are necessary to obtain secondary structure annotations for a selected protein family. The annotation procedure contains three stages (*see* Fig. 2). First, we must obtain the list of domains that belong to the protein family and download their structures. The second step is the choice of the template domain and obtaining its annotation. Third, we run annotation algorithm on each domain in the family. The annotation algorithm is implemented in a program called *SecStrAnnotator* and itself consists of three steps: structural alignment, secondary structure assignment, and matching the template SSEs with the query SSEs. We will discuss each of these stages in more detail.

The individual steps of the procedure will be demonstrated on the cytochrome P450 family (CYPs). We will reference some scripts in the text, which can be used for easier automation of the workflow. All these scripts are written in programming language Python3 and are available on our website (<https://webchem.ncbr.muni.cz/Wiki/SecStrAnnotator>) together with *SecStrAnnotator* software.

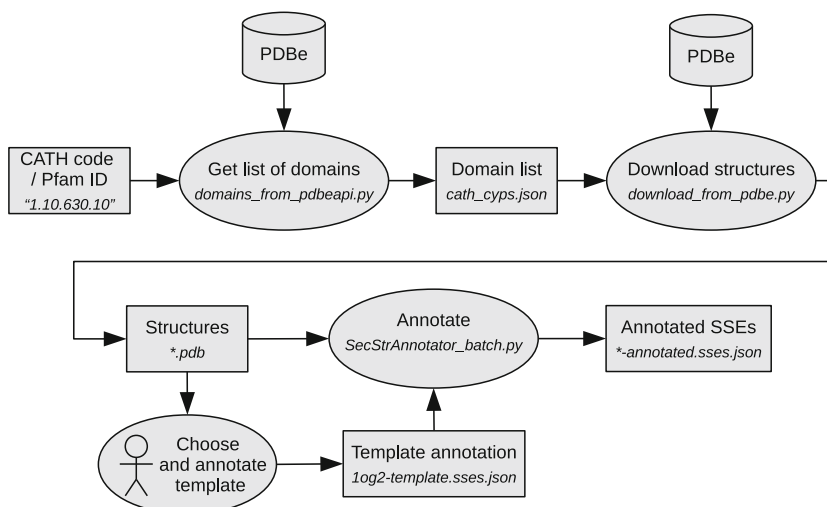


Fig. 2 The overall workflow of SSE annotation performed on a protein family. File names are illustrative (based on CYP family)

3.1 Obtaining the Structures

The list of members of the selected protein family can be acquired from several databases. We will mention two of them, namely, CATH [2] and Pfam [13].

To identify a particular domain, it is necessary to specify the PDB identifier of the structure (PDB ID), chain identifier within the structure, and residue range (or ranges) within the chain. Our notation is best demonstrated on examples:

- Domain 1tqnA00, as defined in CATH, is represented as (1tqn A 28:499), meaning that it is located in the structure with PDB ID 1tqn, in chain A, and it spans residues from 28 to 499.
- Similarly, 1h9rA01 is described as (1h9r A 123:182,255:261), meaning that it is in PDB structure 1h9r and in chain A and consists of two segments, containing residues 123–182 and 255–261.

We use a colon in the residue ranges to avoid confusion between a dash and a minus symbol. In case there are no residues on the chain before or after the domain, the residue numbers in the range can be omitted, e.g., (1tqn A 28:), (1tqn A :499), or (1tqn A :), where the last notation represents the whole chain.

The residue numbers and chain identifier conform to the numbering scheme used in PDB files (`auth_*` numbering scheme). This corresponds to fields `_atom_site.auth_seq_id` and `_atom_site.auth_asym_id` in mmCIF files, rather than `_atom_site.label_seq_id` and `_atom_site.label_asym_id` (`label_*` numbering scheme). It also corresponds to `author_residue_number` and `chain_id` in PDBe REST API.

It is important to be aware of which numbering scheme is used in each moment, because some software tools use `auth_*` while others use `label_*` (*see Note 1*).

3.1.1 List of Domains from CATH

Protein family corresponds to the term *homologous superfamily* used in CATH. At CATH website it is possible to find the selected homologous superfamily and get its CATH code (a four-part numeric identifier such as 1.10.630.10). Alternatively, a cross-reference from a particular structure in PDB can be used.

The list of domains can be obtained programmatically using PDB REST API, specifically *SIFTS mapping* call. An example of such API call is as follows:

```
GET: http://www.ebi.ac.uk/pdbe/api/mappings/1.10.630.10
```

The server response is in a convenient and easy-to-process JSON format. The response can be shown directly in a web browser on the PDB REST API documentation page (<https://www.ebi.ac.uk/pdbe/api/doc/sifts.html>).

A script can be used to call the API, extract all needed information from the response, and write it out into a simplified JSON file (*see Note 2*). Here is an example of calling the script from command line:

```
python3 domains_from_pdbeapi.py 1.10.630.10 > cath_cyps.json
```

The output of the script is a list of domains for each PDB ID in JSON format. Each domain is represented by a three-element array containing domain name, chain identifier, and residue range. The following example of an output contains two domains in PDB entry 1bu7 and one domain in 1tqn:

```
{
  "1bu7": [{"1bu7A00", "A", "1:455"}, {"1bu7B00", "B", "1:455"}],
  "1tqn": [{"1tqnA00", "A", "28:499"}]
}
```

According to CATH convention, domain name 1tqnA00 consists of the PDB identifier 1tqn, chain identifier A, and the number of the domain within the chain (00 is typically used when there is only one domain in the chain; otherwise the domains are numbered 01, 02, etc.). Domain 1tqnA00 would be described as (1tqn A 28:499) in our notation.

3.1.2 List of Domains from Pfam

The website of Pfam database provides a search tool for finding the family of interest. Alternatively, it can be navigated to by a cross-reference from a particular structure in PDB. Once the page of the

family is found, its Pfam ID (a string such as “p450” or “Piwi”) and Pfam accession (such as PF00067 or PF02171) can be obtained.

To obtain the list of domains, PDBe REST API can be used again. The API call is constructed in the same way as with CATH code:

```
GET: http://www.ebi.ac.uk/pdbe/api/mappings/PF00067
```

However, the structure of the response is slightly different than in the case of CATH code; among other things, the domain names are not present. Our script (*see Note 3*) supports both types of response and constructs the missing domain names in a CATH-like manner. The script can be used with Pfam accession code in the same way as with CATH code:

```
python3 domains_from_pdbeapi.py PF00067 > pfam_cyps.json
```

3.1.3 Structures from PDBe

There are many online servers providing PDB structural data. We use PDBe. All needed structures can be downloaded at once, in PDB file format, using script `download_from_pdbe.py`. An example of calling the script:

```
python3 download_from_pdbe.py cath_cyps.json structure_directory
```

3.2 Choice and Annotation of the Template

Our approach to SSE annotation is template-based, meaning that an annotated template domain must be provided to the algorithm. The algorithm then tries to find the annotation of the query domains which well reflects the template annotation. Thus, two tasks are crucial in order to obtain useful annotations for a protein family: selection of the template domain from all domains in the family and preparing the annotation file for this template domain. Our current approach does not include any automated method for fulfilling these tasks; therefore, they must be performed manually. The situation strongly depends on whether an annotation for the selected template domain is available (from literature or other sources). If so, it can be used as the template annotation (possibly with some refinements, described in Subheading 3.2.2). If there is no such annotation, then it must be created from scratch (described in Subheading 3.2.3).

3.2.1 Choice of the Template Domain

There are several requirements for the template domain. An important requirement is availability of SSE annotation in literature. In case of CYPs, the SSE nomenclature is well established, and we based our template annotation on the structure of human CYP 2C9 (PDB ID 1og2) as described by Rowland [5] (*see* Subheading 3.2.2). Unfortunately, sometimes there is no annotation available

in literature. In that case it must be obtained by other procedures, described in *see* Subheading 3.2.3.

The template domain should be a representative of the whole family, so it should be a “typical” or “average” structure rather than an unusual structure which diverges greatly from the rest of the family. If possible, it should contain all the SSEs that are characteristic for the family.

When there are several candidates for the template domain, the quality of the structures should be taken into account—resolution, R values, coverage (i.e., what fraction of residues of the domain are included in the model), and other quality metrics provided by wwPDB structure validation report [16].

In case that a candidate structure contains ligands or other protein chains, it should be checked that these do not induce nonstandard conformation of our domain of interest, as this could have a negative effect on the outcome of the annotation procedure.

An appropriate strategy is also to try out multiple alternatives for the template domain and select the one which performs best. In cases of families with high structural diversity, it might be necessary to divide the family into a few more uniform subgroups and use a separate template for each of them. CATH S35 sequence clusters can serve as a guide to this division.

3.2.2 Refinement of Existing Template Annotation

If some annotation is available, it can be used as it stands (after converting into the required format, described in Subheading 3.2.4). However, it may be appropriate to apply some modifications to this annotation. We will demonstrate these modifications on the annotation of CYP 2C9 (PDB ID 1og2, domain 1og2A00). Individual modification steps are shown in Table 2. For the sake of simplicity, only a few illustrative SSEs are shown (the complete annotation contains more than 30 SSEs).

1. The starting point is the annotation obtained from literature or another source. In case of 1og2, we obtained it from ref. 5. This annotation is shown in column *Original* of Table 2.
2. The best results will be obtained if the secondary structure assignment (SSA; *see* Subheading 1.2) of the template and query domains are obtained by the same method. Therefore, we advise running SSA algorithm on the template domain and making sure that the template annotation is consistent with it. SSA can be run easily by SecStrAnnotator with option `--onlyssa`. Furthermore, the resulting SSA file will be in file format described in Subheading 3.2.4, so it is easier to add SSE names into this file than creating the file manually.

In case of 1og2, we shifted boundaries of helices B, C, and J' by a few residues to make them consistent with the SSA

Table 2
The process of refinement of the template annotation for 1og2

Label	Original →	Consistent SSA →	Additional SSEs →	Artificial SSEs
A	50–61	50–61	50–61	50–61
B	80–89	80–90	80–90	80–90
B''			91–94	91–94
C	117–131	118–131	118–131	118–131
J'	339–342	339–345	339–345	339–345
β 2.1			374–374	374–374
β 2.2			381–381	381–381
β 4.1	472–473	472–473	472–473	472–473
β 4.2	478–479	478–479	478–479	478–479
β 4.3				462–462

SSEs added or changed in each step are shown in bold

algorithm used in SecStrAnnotator. The result is shown in column *Consistent SSA* of Table 2.

3. Sometimes it can be discovered that some SSEs are frequently present in members of a protein family but are not included in the annotation that was obtained from literature. This can be a reason to add these SSEs to the template annotation.

This is illustrated by a short helix found between helices B and C of 1og2. We found out that it is present in more than 80% members of the CYP family and gave it a name B'' (B' is already used). Sheet β 2, consisting of strands β 2.1 and β 2.2, belongs to SSEs with traditionally established names but was missing in the annotation from ref. 5. Column *Additional SSEs* in Table 2 shows the template annotation after adding B'' and sheet β 2.

4. Some SSEs are typical for a protein family (thus worth being annotated) but do not occur in all its members. It is not always possible to find a template domain which would contain all SSEs that we want to annotate in the family. In such case we are forced to use a little trick and add the missing SSEs to the template annotation artificially (even though it is in contradiction with point 2).

As an example, in some CYPs, sheet β 4 consists of three strands (β 4.1, β 4.2, β 4.3), while in others, including 1og2, it is formed only by two strands (β 4.1, β 4.2). We artificially added β 4.3 to the template annotation, in order to allow it to be annotated in those CYPs where it is really present. We

determined approximate position of this artificial β 4.3 based on a few CYPs where β 4.3 is present. The annotation after this step is shown in column *Artificial SSEs* of Table 2.

3.2.3 Creating Template Annotation from Scratch

If there is no available annotation for any member of the family and no naming convention for the SSEs in the family, the template annotation must be created from scratch. This is a nontrivial task, and we have not yet developed any rigorous algorithm to fulfil it. Therefore, we will only describe an intuitive manual method:

1. Run SecStrAnnotator on the template domain with option `--onlyssa`. This will produce secondary structure assignment, which can be used as a template annotation. The individual SSEs will be labelled sequentially and prefixed by the SSE type (e.g., H0, H1, E2, E3, etc.).
2. Try to annotate the family (or a sample of it) using the template annotation from 1 and inspect the results. If there is some unannotated SSE frequently occurring between two particular annotated SSEs, add it to the template annotation (it will be an artificial SSE).
3. If some SSE from the template annotation occurs very rarely in the family, remove it from the template annotation.

Steps 2 and 3 can be performed repeatedly until a satisfactory template annotation is obtained (*see Note 4*). In some cases, it might turn out that the selected template domain is not appropriate, and another domain will serve as a better template.

The procedure can be illustrated on an example of GPCR family (CATH code 1.20.1070.10), shown in Table 3. We randomly selected domain 3pdsA01 as the template domain. SSA yielded eight helices, automatically labelled H0 to H7 (column *Original* in Table 3). After annotation of some other members of the family, we found a two-strand β -sheet occurring between helices H1 and H2 in around 50% of the structures, so we added two artificial strands in the corresponding position (column *Artificial SSEs*). On the other hand, helix H4 was found in less than 20% of the structures, so we removed it from the template annotation (column *Removed SSEs*). Just for transparency, we assigned labels A to G to the remaining seven helices and β 1.1, β 1.2 to the strands (column *Annotation*).

3.2.4 SecStrAnnotator Annotation Format

The remaining task is to convert the template annotation to the format required by SecStrAnnotator. The SecStrAnnotator annotation format is a JSON file, and its structure is illustrated in Fig. 3 (*see Note 5*).

The annotation file contains an object with key-value pairs corresponding to PDB IDs (keys) and annotation data objects (values). Typically, there is only one key-value pair (i.e., annotation

Table 3
The process of creating the template annotation for domain 3pdsA01

Label	Original →	Artificial SSEs →	Removed SSEs →	Annotation
H0	30–61	30–61	30–61	A
H1	67–96	67–96	67–96	B
		97–97	97–97	β1.1
		101–101	101–101	β1.2
H2	103–136	103–136	103–136	C
H3	147–171	147–171	147–171	D
H4	179–187	179–187		
H5	197–229	197–229	197–229	E
H6	267–299	267–299	267–299	F
H7	305–329	305–329	305–329	G

SSEs added or changed in each step are shown in bold

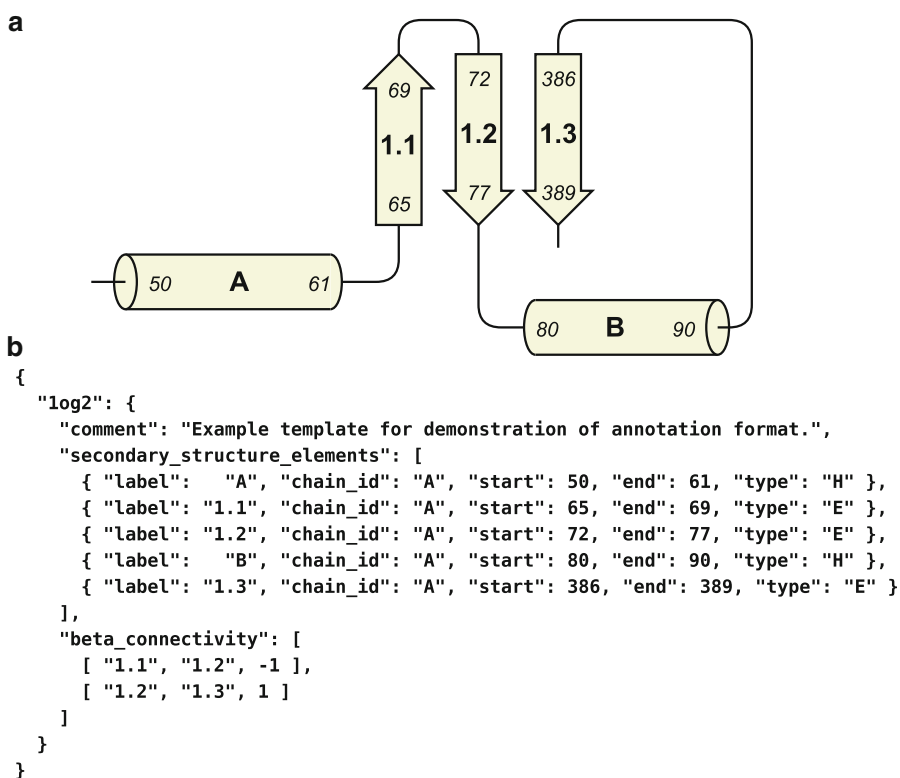


Fig. 3 (a) Topology diagram of an example domain. **(b)** Annotation of the example domain in SecStrAnnotator format. The domain contains two helices, named A and B, and a β -sheet consisting of three strands, named 1.1, 1.2, and 1.3. Strands 1.1 and 1.2 are connected by an antiparallel β -ladder, strands 1.2 and 1.3 by a parallel β -ladder. All the SSEs are located on chain A of structure 1og2

of one structure) in one file. The annotation data object contains keys `secondary_structure_elements` and `beta_connectivity`. Additional information, such as `comment`, can be included but will be ignored by `SecStrAnnotator`.

The value of `secondary_structure_elements` must be an array of objects, each object describing a single SSE. Each of these objects should contain the following:

- `label`—name of the SSE, unique within the domain,
- `chain_id`—chain identifier,
- `start`, `end`—residue number of the first and the last residue in the SSE,
- `type`—type of the SSE. The value can be "H" or "h" for a helix and "E" or "e" for a β -strand. More detailed distinction can be made using DSSP convention [15], i.e., "G" for 3_{10} -helix, "H" for α -helix, "I" for π -helix, "B" for β -strand (with one residue), and "E" for β -strand (with at least two residues); nevertheless, `SecStrAnnotator` will consider them as equivalent to "H" or "E".

The necessity of the `beta_connectivity` section in the template annotation depends on the choice of matching algorithm used in `SecStrAnnotator`. The default algorithm (MOM) takes connectivity of β -strands into account; therefore, the section is required. When using the alternative algorithm (DP), which ignores β -connectivity, this section can be omitted.

The value of `beta_connectivity` contains an array of connection items. Each connection item itself is an array containing two strings and one number and bears information about two β -strands connected by a β -ladder. The two strings are labels of the two connected strands, and the number describes the relative orientation of the strands (1 for parallel, -1 for antiparallel).

3.3 Running `SecStrAnnotator`

In the previous steps, we described how to obtain a list of domains belonging to a protein family, their structures, and an annotation of one of these domains (template domain). Now the annotation algorithm, implemented in `SecStrAnnotator`, can be executed on each domain.

`SecStrAnnotator` finds annotation for a query domain Q , based on the template domain T . Thus, the input consists of the structure of T , structure of Q , and annotation of T . The algorithm consists of three steps. First, it will perform structural alignment of T and Q , so that their corresponding SSEs are located close to each other. Then, it will run secondary structure assignment (SSA) on domain Q . Finally, it will match the template SSEs to the query SSEs, and for each annotated SSE in T , it will select the corresponding SSE in Q .

`SecStrAnnotator` is implemented in C# programming language. It can be downloaded from our website (<https://webchem.ncbr.muni.cz/Wiki/SecStrAnnotator>) together with

`SecStrAnnotator_batch.py`, which is a wrapper for running `SecStrAnnotator` on a batch of domains and collecting the results into one annotation file (*see* **Notes 6** and **7**).

On Windows, `SecStrAnnotator` is executed from command line using the following syntax:

```
SecStrAnnotator.exe [OPTIONS] DIRECTORY TEMPLATE QUERY
```

On Linux, it can be executed using Mono (requires installing `mono-devel` package):

```
mono SecStrAnnotator.exe [OPTIONS] DIRECTORY TEMPLATE QUERY
```

The argument `DIRECTORY` is the directory containing all the input files. The output files will also be saved to this directory. The remaining arguments `TEMPLATE` and `QUERY` describe the domains T and Q . Acceptable formats for these arguments are PDB (e.g., `1tqn`) or `PDB,CHAIN` (e.g., `1tqn,A`) or `PDB,CHAIN,RANGES` (e.g., `1tqn,A,:`). For example, the domain in ranges 123:183, 252:261 on chain B in `1h9r` will be described as `1h9r,B,123:183,252:261`.

The following input files must exist:

- `DIRECTORY/TEMPLATEPDB.pdb` (structure of T in PDB format).
- `DIRECTORY/QUERYPDB.pdb` (structure of Q in PDB format).
- `DIRECTORY/TEMPLATEPDB-template.sses.json` (annotation of T in format described in Subheading 3.2.4).

The output files will be:

- `DIRECTORY/QUERYPDB-detected.sses.json` (SSA of Q).
- `DIRECTORY/QUERYPDB-annotated.sses.json` (annotation of Q).

`SecStrAnnotator` has dependencies on other programs (PyMOL, optionally DSSP) and scripts (`script_align.py`, `script_session.py`). These auxiliary files need to be available in the system, and their location must be specified in the configuration file `SecStrAnnotator_config.json`. The configuration file itself must be in the same directory as `SecStrAnnotator.exe`. Modification of the configuration file might be necessary for successful execution (mainly setting the location of PyMOL on Windows).

Options

The following is an enumeration of the most important command line options. Default values (printed in bold) have been selected to be the most appropriate and robust.

- `--help`
Prints help message and returns.
- `--align METHOD`
Specifies structural alignment method, `METHOD` is one of `align`, `super`, `cealign`, `none` (more in Subheading 3.3.1).
- `--ssa METHOD`
Specifies secondary structure assignment method, `METHOD` is one of `file`, `dssp`, `hbond`, `geom-dssp`, `geom-hbond` (more in Subheading 3.3.2).
- `--onlyssa`
Changes the behavior of `SecStrAnnotator` so that it only runs SSA (no alignment and matching step). In this case it is executed: `SecStrAnnotator.exe [OPTIONS] DIRECTORY QUERY` (i.e., `TEMPLATE` argument is skipped).
- `--limit LIMIT`
Specifies the value of parameter r_0 (in angstroms) in geometrical SSA method, default `1.0` (more in Subheading 3.3.2).
- `--matching METHOD`
Specifies matching method, `METHOD` is one of `dp`, `mom` (more in Subheading 3.3.3).
- `--soft`
Switches on the soft matching variant in MOM algorithm (MOM-soft, more in Subheading 3.3.3).
- `--session`
Creates a PyMOL session visualizing the resulting annotation (*see Note 8*).

3.3.1 Structural Alignment

Structural alignment is realized by calling PyMOL. Option `--align` is used to select which PyMOL's command will be used for alignment:

- *align* (fastest but sequence dependent),
- *super* (slower, sequence independent),
- *cealign* (slowest but very robust, sequence independent CE algorithm [17]).

The default method is *cealign*, and it is preferred unless some performance issues are encountered.

3.3.2 Secondary Structure Assignment (SSA)

`SecStrAnnotator` allows several methods of SSA to be used (*see Note 9*). They can be selected by option `--ssa`. The default and recommended method is *geom-hbond*. However, other methods can be used:

- *file*: The SSA is simply loaded from file `DIRECTORY/QUERY.PDB.sses.json` in format described in Subheading 3.2.4.

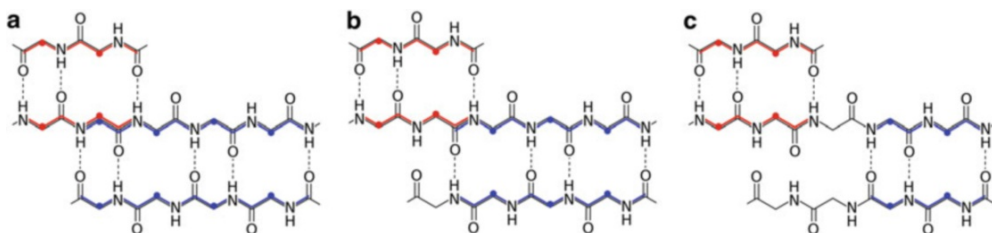


Fig. 4 Marginal situations for relative position of two antiparallel β -ladders: **(a)** the two ladders share four backbone atoms (one residue) and thus constitute single three-strand β -sheet; **(b)** the ladders share two backbone atoms (no residue) and are treated differently by DSSP (as two separate sheets) and our algorithm (as one sheet); **(c)** the ladders share no backbone atoms (no residue) and thus constitute two independent β -sheets. Backbone atoms belonging to each ladder are indicated by red and blue lines. Residues belonging to each ladder are indicated by a red or blue bead on their C^α atom

- *dssp*: DSSP program is executed on the query structure.
- *hbond*: Our modified built-in implementation of DSSP algorithm [15]. The most notable modification is in distinguishing between two β -ladders sharing one strand (thus constituting one sheet) and two independent β -ladders (in two different sheets). The difference between DSSP and our approach can be explained as follows: a backbone atom belongs to a ladder if it lies on a cycle formed by covalent bonds and hydrogen bonds of the ladder. A residue belongs to a ladder if its C^α atom belongs to the ladder (the original paper [15] uses different but equivalent formulation). DSSP considers two ladders to share a strand if they share at least one residue. We consider two ladders to share a strand if they share at least one backbone atom—this is more natural and also consistent with some other software, such as HERA [7] and PROMOTIF [8]. The marginal situations are shown in Fig. 4.
- *geom-hbond*: This is a combined method— β -strands are assigned by *hbond*, while helices are assigned using a geometrical method similar to ref. 18, described in the following text. This method is based purely on the geometry of protein backbone. 3×4 matrix \mathbf{Q}_i contains the coordinates of C^α atoms of residues i , $i + 1$, $i + 2$, and $i + 3$ in a chain. \mathbf{H} denotes “ideal” coordinates of four consecutive C^α in α -helix. This “ideal” coordinates were obtained from α -helices in experimental protein structures. $RMSD_i^H$ is defined as the RMSD between \mathbf{Q}_i and \mathbf{H} (after superimposition). If two or more consecutive values $RMSD_j^H \dots RMSD_k^H$ are below the threshold r_0 , then residues $j + 1$ to $k + 2$ are assigned as a helix. The parameter r_0 can be adjusted using option `--limit`. Lower values of r_0 will lead to stricter assignment with shorter and more regular helices, whereas higher values will tend to assign longer helices which may be curved and contain irregularities (kinks). Its default value 1.0 Å

is quite tolerant to irregularities (compared to DSSP), which is suitable for the purposes of annotation (kinked helices are usually annotated as one helix rather than being divided into two shorter helices). The algorithm does not distinguish between different types of helices (α , 3_{10} , π).

- *geom-dssp*: Another combined method—uses DSSP for β -strands and the geometrical method for helices.

In order to allow easy comparison between SSEs, they are simplified to *line segments*, i.e., start-point and end-point of each SSE is calculated. This is done in by an algorithm related to our geometrical SSA method. Besides the ideal helix geometry \mathbf{H} , it uses $\mathbf{a}_{\mathbf{H}}$, a unit-length (column) vector with the direction of the axis of the ideal helix \mathbf{H} . The axis vector of a real helix spanning residues j to k is then calculated as

$$\mathbf{a} = \sum_{j \leq i \leq k-3} \mathbf{R}_i \mathbf{a}_{\mathbf{H}} \quad (1)$$

where \mathbf{R}_i is the rotation matrix of superimposition of \mathbf{H} onto \mathbf{Q}_i . Center of the real helix is calculated as

$$\mathbf{c} = \frac{1}{k-j+1} \sum_{j \leq i \leq k} \mathbf{r}_i \quad (2)$$

where \mathbf{r}_i is the position of C^α atom of residue i . The axis of the helix is the straight line p which passes through \mathbf{c} and has direction \mathbf{a} . The start-point \mathbf{u} and end-point \mathbf{v} of the helix are then calculated as the projection of its first and last C^α atom onto p :

$$\mathbf{u} = \mathbf{c} + \frac{(\mathbf{r}_j - \mathbf{c}) \cdot \mathbf{a}}{\mathbf{a} \cdot \mathbf{a}} \mathbf{a} \quad (3)$$

$$\mathbf{v} = \mathbf{c} + \frac{(\mathbf{r}_k - \mathbf{c}) \cdot \mathbf{a}}{\mathbf{a} \cdot \mathbf{a}} \mathbf{a} \quad (4)$$

The calculation of the line segment is illustrated in Fig. 5. Line segments for β -strands are calculated in the same manner, except \mathbf{E} and $\mathbf{a}_{\mathbf{E}}$ (C^α coordinates and axis vector of an ideal strand) are used

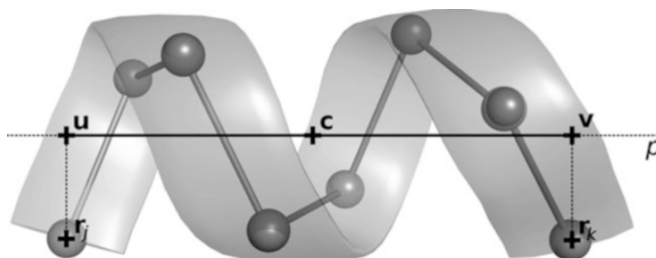


Fig. 5 Calculation of line segment \mathbf{uv} for a helix

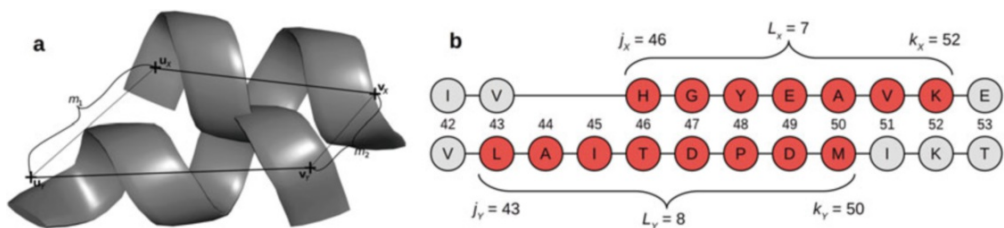


Fig. 6 Calculation of metric μ between two helices, X and Y . **(a)** Spatial part of μ : the first term in Eq. (5) is calculated as $0.5 \times (m_1 + m_2) = 0.5 \times (5.5 + 3.3) = 4.4$. **(b)** Structural alignment-based part of μ : the second term is calculated as $0.5 \times (|46 - 43| + |52 - 50|) = 2.5$ and the third term as $10 \times |7 - 8| / \sqrt{(7 \times 8 + 9^2)} = 0.85$

instead of \mathbf{H} and \mathbf{a}_H . This calculation is used regardless of which methods was used for SSA.

3.3.3 SSE Matching

This is the core part of algorithm. The goal is to find the optimal matching between the SSEs of the domains T and Q . Subsequently, each matched SSEs in Q can be annotated by the same label as the SSE in T it was matched to.

Before the optimal matching can be found, it is necessary to have a measure of similarity between two SSEs, X and Y , belonging to T and Q , respectively. For this purpose, we define metric μ :

$$\begin{aligned} \mu(X, Y) = & c_1(\|\mathbf{u}_X - \mathbf{u}_Y\| + \|\mathbf{v}_X - \mathbf{v}_Y\|) + c_2(|j_X - j_Y| + |k_X - k_Y|) \\ & + c_3 |L_X - L_Y| / \sqrt{L_X L_Y + c_4^2} \end{aligned} \quad (5)$$

where $\mathbf{u}_X \mathbf{v}_X$ ($\mathbf{u}_Y \mathbf{v}_Y$) is the line segment for X (Y), j_X, k_X (j_Y, k_Y) are the positions of the first and last residue of X (Y) in the structural alignment of T and Q , and L_X (L_Y) is the number of residues in X (Y). The values of parameters c_1 to c_4 have been optimized to $c_1 = 0.5$, $c_2 = 0.5$, $c_3 = 10$, and $c_4 = 9$. Higher values of $\mu(X, Y)$ mean bigger difference between X and Y . Calculation of metric μ is illustrated in Fig. 6.

SecStrAnnotator allows choice from two different *matching algorithms*. DP algorithm is fast but ignores β -connectivity. MOM algorithm includes β -connectivity but might be slower under some circumstances (see **Note 10**). The default algorithm is MOM.

DP algorithm (standing for *dynamic programming*) ignores the β -connectivity of the structures and therefore can take advantage of dynamic programming technique [19]. The algorithm is very similar to the well-known Needleman-Wunsch algorithm used for sequence alignment in bioinformatics [20]. Let's denote $X = (X_i)_{1 \leq i \leq m}$, the sequence of template SSEs, i.e., the annotated

SSEs in the template domain T , in the same order as they appear in the primary structure. Similarly, $\Upsilon = (\Upsilon_i)_{1 \leq i \leq n}$ is the sequence of query SSEs, i.e., all SSEs found in the query domain Q .

The score S for matching SSE X_i with SSE $\Upsilon_{i'}$ is defined:

$$\begin{aligned} S(i, i') &= K - \mu(X_i, \Upsilon_{i'}) && \text{for } X_i, \Upsilon_{i'} \text{ of the same SSE type} \\ S(i, i') &= 0 && \text{for } X_i, \Upsilon_{i'} \text{ of different SSE types} \end{aligned}$$

where SSE type refers to two-class distinction (helix vs. strand). The default value of parameter K is set to 30 (*see Note 11*). Higher values of S indicate more similar SSEs.

The goal of the algorithm is to find a matching $M \subseteq \{1 \dots m\} \times \{1 \dots n\}$ which:

- (a) Preserves the order of SSEs:
 $i < j \Leftrightarrow i' < j'$ for each $(i, i') \in M, (j, j') \in M$
- (b) Matches only SSEs with positive score:
 $S(i, i') > 0$ for each $(i, i') \in M$
- (c) Maximizes the total score:
 $S_{\text{total}} = \sum_{(i, i') \in M} S(i, i')$.

The optimal matching M is then found using the dynamic programming technique. The computational complexity is $O(mn)$.

MOM algorithm (standing for *mixed ordered matching*) takes the β -connectivity into account. Unlike DP algorithm, matching helix-to-helix and strand-to-strand, MOM matches helix-to-helix and ladder-to-ladder. Therefore, it requires slightly different formulation of the problem than DP algorithm. Besides X and Υ , we will define the set of template helices $H_X = \{i | X_i \text{ is a helix}\}$, the set of template strands $E_X = \{p | X_p \text{ is a strand}\}$, and the set of template ladders $L_X = \{pq | p, q \in E_X \wedge p < q \wedge X_p \text{ forms a ladder with } X_q\}$. Note that ladders are formally expressed as tuples, e.g., (p, q) , but for better readability, we use shortened notation pq . Sets H_Υ, E_Υ , and L_Υ are defined analogously for the query domain. In the following text, we will keep using indices i and j exclusively for helices and p, q, u , and v for strands; prime symbol will be used with query SSEs.

The goal is to find the optimal matching $M \subseteq H_X \times H_\Upsilon \cup L_X \times L_\Upsilon$, thence the word *mixed* in the name of the algorithm. There are the same three requirements for the matching M as in the case of DP algorithm; however, it is more complicated to express them formally. The matching M must:

- (a) Preserve the order of SSEs:

$$\begin{aligned} o(i, j, i', j') &\text{ for each } (i, i') \in M, (j, j') \in M \\ o(i, p, i', p') \wedge o(i, q, i', q') &\text{ for each } (i, i') \in M, (pq, p'q') \in M \end{aligned}$$

$$o(p, u, p', u') \wedge o(p, v, p', v') \wedge o(q, u, q', u') \wedge o(q, v, q', v') \text{ for each } (pq, p'q') \in M, (uv, u'v') \in M$$

where o is the order consistency predicate, defined as $o(i, j, i', j')$:
 $i < j \Leftrightarrow i' < j'$ (in human words: i with j goes in the same order as i' with j').

(b) Match only SSEs with positive score:

$$\begin{aligned} S(i, i') > 0 & \text{ for each } (i, i') \in M \\ S(p, p') > 0 \wedge S(q, q') > 0 & \text{ for each } (pq, p'q') \in M \end{aligned}$$

(c) Maximize the total score:

$$S_{\text{total}} = \sum_{(i, i') \in M} S(i, i') + \sum_{(pq, p'q') \in M} [S(p, p') + S(q, q')]$$

The problem of finding the matching M can now be easily reduced to the problem of finding a maximum-weight clique in a weighted graph (a clique is a subset of vertices all adjacent to each other). The vertices of the graph are all helix-to-helix and ladder-to-ladder matches with positive score (i.e., fulfilling criterion (b)). The edges of the graph connect only those pairs of vertices which preserve the order of SSEs (i.e., fulfilling criterion (a)). The weight of each vertex is simply the score S , and a clique with maximum total weight is to be found (i.e., fulfilling criterion (c)).

The maximum-weight clique problem can be solved by a backtracking algorithm, which systematically enumerates all inclusion-maximal cliques and selects the one with the best total weight (known as Bron-Kerbosch algorithm [21]). The algorithm can be further improved by using branch-and-bound technique—this means that the current maximum (the best weight found so far) is remembered, and any branch of the algorithm whose maximum expected weight is lower than the current maximum is evaluated as non-perspective and thus is ignored. This modification significantly improved the running time of the algorithm in most tested cases, yet a good worst-case computational complexity cannot be guaranteed. The algorithm will always find an optimal solution. In theoretical case of two cliques having exactly the same weight, the algorithm will find only one of them; nevertheless, in practice this is extremely unlikely. The theoretical computational complexity of MOM algorithm is exponential; however, in most cases the running times are very close to those of DP.

MOM-soft is a slight modification of the MOM algorithm, which allows two ladders which share a strand (see Fig. 4a) to be matched to two ladders whose strands are close to each other (see Fig. 4c)—this is a kind of variation that often occurs in protein

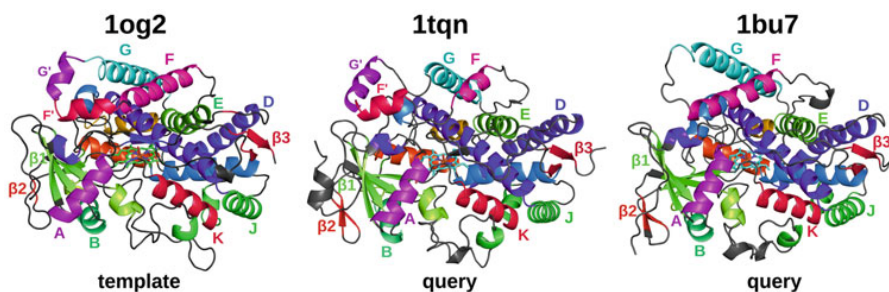


Fig. 7 Example of annotation of two query cytochrome P450 protein domains annotated with 1log2 used as template. For transparency, only visible SSEs are labelled

families, so it may be desirable to allow such matching. The only difference from MOM is that the order consistency predicate is defined as

$$o(i, j, i', j') : (i < j \wedge i' < j') \vee (i > j \wedge i' > j') \vee (i = j \wedge |i' - j'| \leq 1) \vee (|i - j| \leq 1 \wedge i' = j')$$

(see **Note 12**). MOM-soft is switched on by option `--soft`.

The final step, after running MOM or DP matching algorithm, is the transfer of SSE labels from the template to the query domain. In case of MOM, this means that for each pair of matched helices $X_i, Y_{i'}$ (i.e., for each $(i, i') \in M$), helix $Y_{i'}$ is assigned the same label as X_i has. Similarly, for each pair of matched ladders $X_p X_q, Y_{p'} Y_{q'}$ (i.e., for each $(pq, p'q') \in M$), strand $Y_{p'}$ gets the same label as X_p , and strand $Y_{q'}$ gets the same label as X_q . In case of DP, the situation is more straightforward: both the helices and the strands are annotated in the same way as the helices in MOM. Finally, all template SSEs which have been annotated are written into the output file with their newly assigned labels (see **Note 13**).

The results of the annotation algorithm are illustrated on two protein domains in Fig. 7.

4 Notes

1. The `auth_*` numbering scheme allows use of insertion codes, so in some situations, the residue numbering is not straightforward, e.g., 85, 86, 86A, 86B, 87, 88, etc. This complicates the situation even more. The current version of SecStrAnnotator does not support insertion codes and will raise an error if it encounters any. An ugly fix to this is running SecStrAnnotator with `--ignoreinsertions`; however, the structure will then not correspond to the real structure because all residues with insertion codes will be ignored. A better solution is converting an mmCIF file to a PDB file in such way that the `label_*`

numbering is used instead of `auth_*`. This can be done with help of PyMOL, using script `cif_to_pdb_with_label_numbering.py`. Then it is necessary that the domain residue ranges be in the `label_*` numbering as well (can be obtained by running `domains_from_pdbeapi.py` with `--numbering label`).

2. An alternative way of obtaining the list of domains is to download CATH classification file and filter the domains which belong to the selected homologous superfamily. Nevertheless, this has the disadvantage of including obsolete PDB entries in the list (whereas PDBe API excludes obsolete structures).
3. The Pfam database also provides its own API, which might be used as an alternative to PDBe API (with output in XML or tab-delimited format).
4. In the described procedure, the words “frequently” and “rarely” are very subjective and may also depend on the purpose for which the annotation is performed. In some situations, it may be desirable to have a rich template annotation with some SSEs occurring only in a small fraction of the family members. In other cases, it will be suitable to include only the most frequently occurring SSEs in the template annotation even if many rarer SSEs will then stay unannotated.
5. JSON files are sometimes not nicely formatted (without new lines and indentation) and are hard to read. Some web browsers (e.g., Firefox) can visualize such files in a human-friendly interactive form (although installation of extensions may be necessary).
6. SecStrAnnotator cannot guarantee 100% correctness of the provided annotations. Due to the diversity between the structures, there exist twilight-zone cases, in which it is unclear what the correct annotation should be. Therefore, even determining the error rate is very subjective. We performed a manual validation for CYP and GPCR families, and we can claim that the ratio of incorrectly annotated SSEs was under 3% and 0.5%, respectively.
7. Running time of SecStrAnnotator on one domain is typically a few seconds, depending on the size of the structures and available hardware. `SecStrAnnotator_batch.py` can reduce the overall running time for the whole family by running on several CPU cores in parallel (option `--threads`).
8. Visual inspection of the resulting annotation in the automatically created PyMOL session is a simple way of checking the results for possible wrongly annotated SSEs. However, this becomes less convenient as the number of annotated domains gets bigger. Then performing statistics and detection of outliers can be used to uncover wrong annotations.

9. There are many possible criteria for defining secondary structure, and therefore many different SSA methods have been developed [22, 23]. We use *geom-hbond* as the default method because it focuses on the overall shape of helices instead of the details of hydrogen bonding patterns, which are not relevant for the annotation. On the other hand, hydrogen bond approach is used for β -strands, for two reasons: first, connections between strands are vital (a β -strand is not a β -strand without being bound to another β -strand by a ladder), and second, occurrence of β -bulges significantly disrupts the shape of β -strands, so it is hard to describe their geometry universally.
10. The computational complexity of DP algorithm is quadratic with respect to the number of SSEs in T and Q . The theoretical computational complexity of MOM algorithm is exponential; however, in most cases the running times are very close to those of DP. Therefore, MOM is preferred unless serious performance issues are encountered. For structures without β -strands the two algorithms will give identical matching.
11. The value of parameter K can be adjusted using option `--maxmetric`. Decreasing K will result in stricter matching – only very similar SSEs will be allowed to be matched together; the resulting annotation will therefore tend to contain less annotated SSEs but will also be less likely to contain wrong annotations. Increasing K will make the algorithm more tolerant to differences between matched SSEs. This might be necessary in protein families with higher structural diversity. However, it should be done with precaution because too high values of K will cause the algorithm to maximize the number of matched SSEs without focusing on their similarity. High values of K can also slow down MOM algorithm. The option `--maxmetric` can also be used to define K as a linear function of the lengths of SSEs X_i and Y_j and so to put more importance on matching longer SSEs. It is possible to find out the score S for a pair of SSEs from output file `score_matrix.tsv` (when `SecStrAnnotator` is run with `--verbose`). Values of metric μ for matched SSE pairs are included in the output annotation file (field `metric_value`).
12. There is also a constraint that each helix (ladder) can be matched to at most one helix (ladder) – this constraint was not mentioned in DP and pure MOM, because it was a logical consequence of the other constraints.
13. Although obtaining the SSE annotations is the ultimate goal in this chapter, it is often advisable to perform additional statistics on the results over a larger set of proteins from the protein family. Analysis of the distribution of length of the individual

SSEs can provide overview of the general SSE anatomy of the protein family. It can also detect outliers, which may be due to erroneous annotations. Finally, it can help uncover interesting features, like correlation of the structure with source organism or function of the protein.

Acknowledgments

This work was supported by ELIXIR CZ research infrastructure project (MEYS) [LM2015047 to A.M., I.H.V., J.H., K.B., and R.S.V.]; Ministry of Education, Youth and Sports of the Czech Republic under the project CEITEC 2020 [LQ1601 to A.M., R.S.V., and J.K.]; ELIXIR-EXCELERATE project, which received funding from the European Union’s Horizon 2020 research and innovation program [676559]; ELIXIR-CZ: Budování kapacit [CZ.02.1.01/0.0/0.0/16_013/0001777]; Ministry of Education, Youth and Sports of the Czech Republic [project CZ.02.1.01/0.0/0.0/16_019/0000754 to V.N. and K.B.]; and Palacky University Olomouc [IGA_PrF_2018_032 to V.N.]. A.M. is a “Brno Ph.D. Talent” scholarship holder funded by Brno City Municipality.

References

- Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H et al (2000) The protein data bank. *Nucleic Acids Res* 28 (1):235–242. <https://doi.org/10.1093/nar/28.1.235>
- Sillitoe I, Lewis TE, Cuff A, Das S, Ashford P, Dawson NL et al (2015) CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res* 43(D1):D376–D381. <https://doi.org/10.1093/nar/gku947>
- Fox NK, Brenner SE, Chandonia JM (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>
- Poulos TL, Finzel BC, Howard AJ (1987) High-resolution crystal structure of cytochrome P450cam. *J Mol Biol* 195 (3):687–700. [https://doi.org/10.1016/0022-2836\(87\)90190-2](https://doi.org/10.1016/0022-2836(87)90190-2)
- Rowland P, Blaney FE, Smyth MG, Jones JJ, Leydon VR, Oxbrow AK et al (2006) Crystal structure of human cytochrome P450 2D6. *J Biol Chem* 281(11):7614–7622. <https://doi.org/10.1074/jbc.M511232200>
- Cojocaru V, Winn PJ, Wade RC (2007) The ins and outs of cytochrome P450s. *Biochim Biophys Acta* 1770(3):390–401. <https://doi.org/10.1016/j.bbagen.2006.07.005>
- Hutchinson EG, Thornton JM (1990) HERA—a program to draw schematic diagrams of protein secondary structures. *Proteins* 8(3):203–212. <https://doi.org/10.1002/prot.340080303>
- Hutchinson EG, Thornton JM (1996) PRO-MOTIF—a program to identify and analyze structural motifs in proteins. *Protein Sci* 5 (2):212–220. <https://doi.org/10.1002/pro.5560050204>
- Stivala A, Wybrow M, Wirth A, Whisstock JC, Stuckey PJ (2011) Automatic generation of protein structure cartoons with Pro-origami. *Bioinformatics* 27(23):3315–3316. <https://doi.org/10.1093/bioinformatics/btr575>
- Svobodova Varekova R, Midlik A, Hutarova Varekova I, Hutar J, Navratilova V, Koca J et al (2018) Secondary structure elements—annotations and schematic 2D visualizations stable for individual protein families. *Biophys J* 114(3):46a–47a. <https://doi.org/10.1016/j.bpj.2017.11.307>

11. Berman H, Henrick K, Nakamura H (2003) Announcing the worldwide Protein Data Bank. *Nat Struct Biol* 10(12):980–980. <https://doi.org/10.1038/nsb1203-980>
12. Velankar S, Dana JM, Jacobsen J, van Ginkel G, Gane PJ, Luo J et al (2013) SIFTS: structure integration with function, taxonomy and sequences resource. *Nucleic Acids Res* 41(D1):D483–D489. <https://doi.org/10.1093/nar/gks1258>
13. Finn RD, Coghill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL et al (2016) The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Res* 44(D1):D279–D285. <https://doi.org/10.1093/nar/gkv1344>
14. The PyMOL Molecular Graphics System, Version 2.0 Schrodinger, LLC
15. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22(12):2577–2637. <https://doi.org/10.1002/bip.360221211>
16. Gore S, Sanz Garcia E, Hendrickx PMS, Gutmanas A, Westbrook JD, Yang H et al (2017) Validation of structures in the Protein Data Bank. *Structure* 25(12):1916–1927. <https://doi.org/10.1016/j.str.2017.10.009>
17. Shindyalov IN, Bourne PE (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng* 11(9):739–747. <https://doi.org/10.1093/protein/11.9.739>
18. Mitchell EM, Artymiuk PJ, Rice DW, Willett P (1990) Use of techniques derived from graph theory to compare secondary structure motifs in proteins. *J Mol Biol* 212(1):151–166. [https://doi.org/10.1016/0022-2836\(90\)90312-A](https://doi.org/10.1016/0022-2836(90)90312-A)
19. Eddy SR (2004) What is dynamic programming? *Nat Biotechnol* 22:909. <https://doi.org/10.1038/nbt0704-909>
20. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 48(3):443–453. [https://doi.org/10.1016/0022-2836\(70\)90057-4](https://doi.org/10.1016/0022-2836(70)90057-4)
21. Bron C, Kerbosch J (1973) Algorithm 457: finding all cliques of an undirected graph. *Commun ACM* 16(9):575–577. <https://doi.org/10.1145/362342.362367>
22. Anderson CA, Rost B (2009) Secondary structure assignment. In: Gu J, Bourne PE (eds) *Structural bioinformatics*, 2nd edn. Wiley, Hoboken
23. Cao C, Xu ST, Wang LC (2015) An algorithm for protein helix assignment using helix geometry. *PLoS One* 10(7):20. <https://doi.org/10.1371/journal.pone.0129674>

Uncovering of cytochrome P450 anatomy by SecStrAnnotator

Adam Midlik^{1,2}, Veronika Navrátilová³, Taraka Ramji Moturu^{1,2}, Jaroslav Koča^{1,2}, Radka Svobodová^{1,2}, Karel Berka³

¹ CEITEC – Central European Institute of Technology, Masaryk University, Kamenice 5, 625 00 Brno, Czech Republic

² National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Kamenice 5, 625 00 Brno, Czech Republic

³ Department of Physical Chemistry, Faculty of Science, Palacký University, 17. listopadu 1192/12, 771 46 Olomouc, Czech Republic

Scientific Reports, 11: 12345, **2021**.

<https://doi.org/10.1038/s41598-021-91494-8>



OPEN

Uncovering of cytochrome P450 anatomy by SecStrAnnotator

Adam Midlik^{1,2}, Veronika Navrátilová³, Taraka Ramji Moturu^{1,2}, Jaroslav Koča^{1,2}, Radka Svobodová^{1,2} & Karel Berka³

Protein structural families are groups of homologous proteins defined by the organization of secondary structure elements (SSEs). Nowadays, many families contain vast numbers of structures, and the SSEs can help to orient within them. Communities around specific protein families have even developed specialized SSE annotations, always assigning the same name to the equivalent SSEs in homologous proteins. A detailed analysis of the groups of equivalent SSEs provides an overview of the studied family and enriches the analysis of any particular protein at hand. We developed a workflow for the analysis of the secondary structure anatomy of a protein family. We applied this analysis to the model family of cytochromes P450 (CYPs)—a family of important biotransformation enzymes with a community-wide used SSE annotation. We report the occurrence, typical length and amino acid sequence for the equivalent SSE groups, the conservation/variability of these properties and relationship to the substrate recognition sites. We also suggest a generic residue numbering scheme for the CYP family. Comparing the bacterial and eukaryotic part of the family highlights the significant differences and reveals a well-known anomalous group of bacterial CYPs with some typically eukaryotic features. Our workflow for SSE annotation for CYP and other families can be freely used at address <https://sestra.ncbr.muni.cz>.

Secondary structure elements (SSEs) are defined by the repetitive pattern of hydrogen bonds and geometric arrangement. The most well-known SSE types are the α -helix and the β -strand. SSEs have been used to analyze protein structures since their first observation by Linus Pauling^{1,2}. They define the structural folds of individual protein structural families as classified by CATH³ or SCOPe⁴ databases. Structural folds, defined by SSEs, may reveal a possible subset of typical biochemical functions of proteins from those protein families⁵. SSEs can also serve as guides for orientation in the protein structures within scientific communities.

In order to compare similar structures, the communities around several proteins families have developed specialized nomenclatures for annotation (labeling) of the SSEs in the members of the family. Such nomenclatures assign the same label to the equivalent SSEs from different proteins in the family. We will refer to a group of equivalent SSEs as an *SSE class*.

Adoption of such SSE nomenclatures is typical for well-studied protein families with a large number of available structures with a common fold but large sequence variations, such as esterases^{6,7}, G-protein coupled receptors (GPCRs)⁸, immunoglobulins⁹, cytochromes P450 (CYPs)¹⁰ and others. These traditional nomenclatures prove to be particularly useful when comparing existing structures, describing new ones or generalizing observations over the whole family.

Annotated SSEs can be used as reference points to describe the position of key regions, such as catalytic sites, selectivity filters, channels, or protein–protein interfaces. A nice illustrative example is again the CYP family (Fig. 1), with a well-established classification of multiple different channels based on their position relative to the annotated SSEs^{11–13} and with substrate selectivity defining residues on several SSEs^{14–18}. The channels of the cytochromes P450 represent a network of the ins and outs which provide the ways for the metabolites to enter the deeply buried active site with the heme cofactor as well as the exit paths for the metabolite output. Channels are named according to their spatial location in respect to SSEs lining each pathway and are summarized in the general nomenclature for cytochrome P450 channels introduced by Cojocar et al.¹¹. These SSEs relate to the regions important for substrate recognition—SRS-1, SRS-2, SRS-3, and SRS-5¹⁷. Moreover, amino acids located in particular SSEs (e.g. F-G loop) play a crucial role in the substrate egress through the channels. In summary, the spatial arrangement of SSEs may differ from one cytochrome P450 to another, which may result in variations

¹CEITEC – Central European Institute of Technology, Masaryk University, Brno 625 00, Czech Republic. ²National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Brno 625 00, Czech Republic. ³Department of Physical Chemistry, Faculty of Science, Palacký University, Olomouc 771 46, Czech Republic. ✉email: radka.svobodova@ceitec.muni.cz; karel.berka@upol.cz

5. MAIN PUBLICATIONS

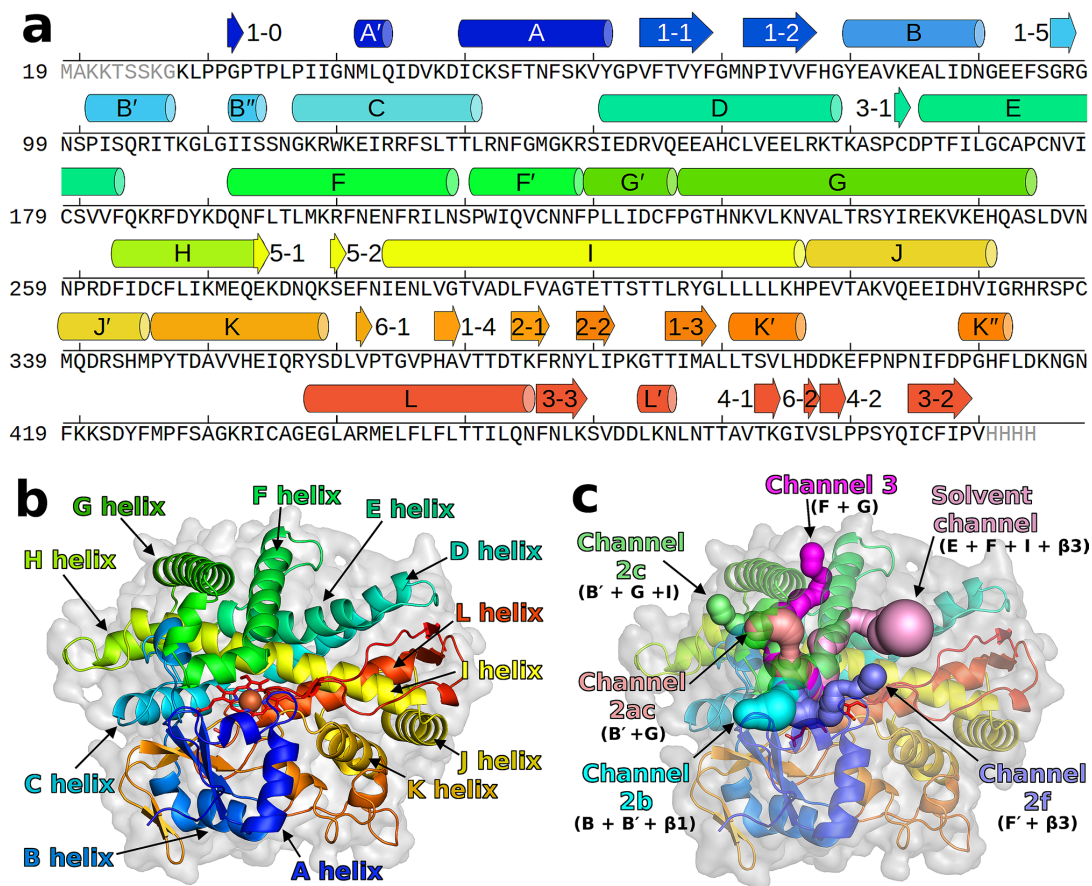


Figure 1. SSE annotation for the CYP family. (a) Annotation shown on the sequence (PDB ID 2nnj), (b) annotation of the major helices shown on the structure, (c) annotation of protein channels defined with respect to the annotated SSEs. Some SSEs (B', β 1-0, β 5-1, β 5-2, β 6-1, β 6-2) are not really present in 2nnj but they are shown to illustrate their location in other CYPs. The table of residue ranges of the SSEs is included in Supplementary Table S1. These figures were created using PyMOL 2.3¹⁹ and GIMP 2.10.18 (<https://www.gimp.org/>).

of channel opening. Annotation of the SSEs in various structures of cytochrome P450 is useful for identifying these channels and thus elucidating the channel preferences of individual cytochromes P450 and its substrates.

In some protein families, the communities have extended the annotation down to the residue level and introduced generic numbering schemes. Such schemes assign the same generic number to the equivalent residue positions in homologous proteins, which facilitates comparisons of mutation effects, ligand interactions, structural motifs etc. The generic numberings may be based on the sequence information (e.g. immunoglobulins²⁰) or may combine the SSE annotations with the sequence information (e.g. GPCRs⁹).

Annotation of SSEs can be valuable even in protein families without defined traditional nomenclature since it provides the equivalence between the SSEs from the individual members of the family (the SSEs from the same SSE class are annotated by the same label, even if the labels are arbitrarily created).

Previously, we presented methods for automated annotation of SSEs in protein families, implemented in tool SecStrAnnotator²¹. Automated annotation opens the possibility to focus on any protein family and describe its general SSE anatomy, which can bring a valuable insight to the understanding of its function—SSEs typically present, their occurrence, typical length, position and amino acid composition and variation for individual SSE classes.

In this paper, we propose a procedure for analysis of the general SSE anatomy of a protein structural family, based on and extending SecStrAnnotator. We demonstrate this type of analysis on the cytochromes P450, a biologically important family with a long tradition of SSE annotation and well-established SSE nomenclature. We also suggest a generic residue numbering scheme for the CYP family. The SSE annotations for the CYP family are accessible online through SecStrAPI and can be easily visualized via a dedicated PyMOL plugin—all

freely available at the SecStrAnnotator website (<https://webchem.ncbr.muni.cz/Wiki/SecStrAnnotator>). New CYP structures can be annotated by SecStrAnnotator Online (<https://sestra.ncbr.muni.cz>).

Traditional SSE nomenclature in the CYP family. The common fold of the CYP family has a triangular prism shape and consists mostly of α -helices combined with several β -sheets and a heme cofactor, which forms the catalytic center of the enzyme^{14,22,23}. The traditional SSE nomenclature in the CYP family is based on the labels used by Poulos et al.²⁴ for the first experimentally determined CYP structure (P450cam) with 12 helices, labeled A–L, and 5 sheets, β 1– β 5. The publication of the refined structure²⁵ mentioned a new helix B' and also several shorter helices and strands without labels.

Ravichandran et al.²⁶ on P450 BM3 changed the labeling scheme for β -sheets to the form which later became widely used in the community: sheets β 1 (5 strands, previously labeled β 1 + β 3), β 2 (2 strands, previously β 4), β 3 (3 strands, previously β 5) and a new sheet β 4 (2 strands). The strands within each sheet can be referred to individually, using a hyphen (e.g. β 1-1, β 1-2). They also added annotation of two new helices – J' (between J and K) and K' (after strand β 1-3).

As more and more structures emerged in the following years, new labels were needed for the newly observed SSE classes: helices A'²⁷, L'^{28,29}, F', G'³⁰, K''²⁹ and B''³¹; sheets β 5²⁷ (corresponding to β 2 in Poulos et al.²⁵) and β 6³²; and strand β 1-0³³. Many other SSEs have been mentioned and labeled in literature but these are either very rare or can be treated as a part of a longer SSE (e.g. helix D' in Park et al.²⁸ can be understood as an N-terminal part of helix D).

Unfortunately, the labeling is not always consistent, sometimes even in the papers by the same author. The same SSE class can be assigned different labels (e.g. helix L' in Scott et al.²⁹ is helix M in Pylypenko et al.³³) or one label can be assigned to different SSE classes (e.g. helix A'' is located between β 1-1 and β 1-2 in Pylypenko et al.³³ but before A' in Williams et al.³⁴). Throughout this paper we will use the nomenclature as shown in Fig. 1 and specified in Supplementary Table S1.

Results and discussion

Structures of proteins from the CYP family typically contain at least 14 helices—A, B, B', C, D, E, F, G, H, I, J, K, K', L – and 4 sheets— β 1 (5 strands), β 2 (2 strands), β 3 (3 strands), β 4 (2 strands). Additional helices A', B'', F', G', J', K'' and L' are often present, as well as two sheets β 5 and β 6 (with two strands in each). Sheet β 1 often contains an extra strand β 1-0 (bonded to β 1-1). The sequential order and 3D position of these SSE classes is shown in Fig. 1. Other SSEs may appear in the structures, but they are not characteristic to the family.

For convenience, we divide all annotated SSE classes into three groups throughout the paper: *major helices* (helices A–L, typically longer than 8 residues), *minor helices* (all the remaining helices, typically shorter than 8 residues) and *strands*.

In the following text, we analyze these SSE classes in terms of frequency of occurrence, length (number of residues) and amino acid composition. We also discuss differences between eukaryotic and bacterial structures. In Supplementary Note we provide a dedicated analysis of the structural irregularities.

Regions of variable secondary structure. There are several regions in CYP structures which are structurally very variable. These regions are mainly:

- Region before helix A (containing β 1-0 and A')—this region usually contains at least one helix (annotated as A'), but often there are more short helices. Furthermore, in eukaryotic CYPs this region should contain membrane anchor, but it is missing in most experimental structures, because it complicates crystallization. Therefore, the structures might be biologically irrelevant in this region (though helix A' was observed also in molecular dynamics simulations on membranes³⁵).
- Region between β 1-5 and B'' (containing helix B'), is a part of so-called BC-loop. Usually it contains one helix, which is annotated as B', but often there is more than one helix (especially in bacteria).
- Region between K' and L (containing helix K'')—this region can contain several short helices with variable positions.

Each of these regions can contain more than one helix and the position of those helices varies from structure to structure. As a result, there is large uncertainty in the annotation of these regions. Therefore, the results for β 1-0, A', B', K'' should be interpreted with caution.

Frequency of occurrence of the SSE classes. This section describes the frequency of occurrence of each SSE class, i.e. in what fraction of the structures the particular SSE is present. The results are shown in Fig. 2.

Major helices. All major helices occur in more than 97% of the structures. In cases where they are missing, it can be attributed to the experiment (bad quality, residue coverage or resolution of the structure), rather than not being formed in the structure.

Minor helices. The most frequent minor helices are K' (100%) and B' (83%), followed by A' (63%), B'' (61%), K'' (46%), J' (35%), L' (30%), F' (28%) and G' (25%). Helices A', B'', K'', L' are very short, so it can often happen that they are not formed at all. On the other hand, the low occurrence of helices F', G', J' can be explained by their absence in bacterial CYPs (roughly 2/3 of all structures)—for more details see section “Comparison of bacterial and eukaryotic CYPs”. Furthermore, the flexible F'G'-loop is often not modeled in the experimental structures.

5. MAIN PUBLICATIONS

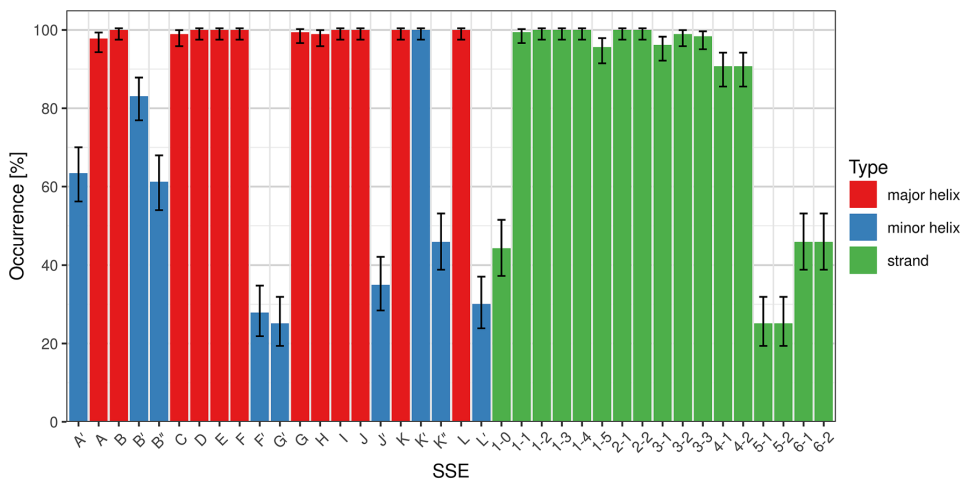


Figure 2. Frequency of occurrence of individual SSE classes. Error bars show confidence intervals calculated by the Agresti-Coull method for $\alpha=0.05$ ³⁶.

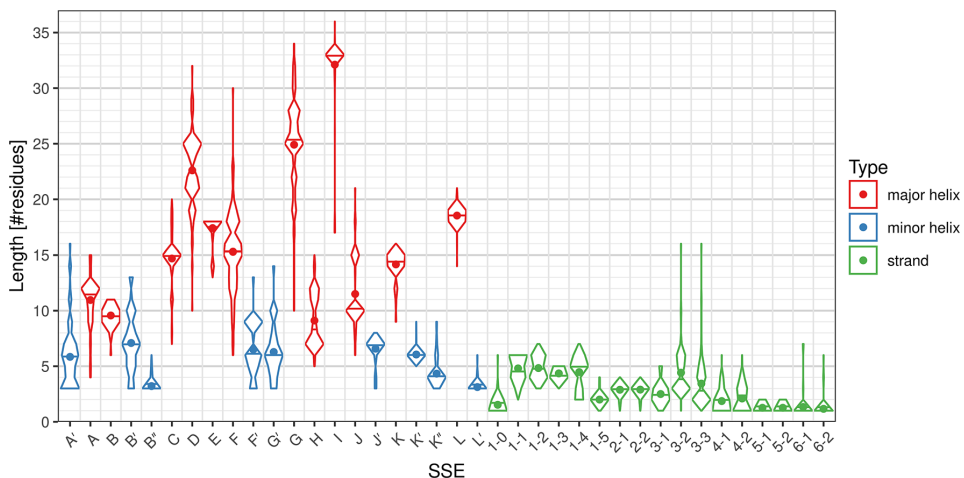


Figure 3. Distributions of length of individual SSE classes. The dots in the violin plot represent the mean, horizontal lines represent the median of the distribution. Non-existing SSEs are not included in the distribution.

Strands. In sheet $\beta 1$, strands $\beta 1-1$, $\beta 1-2$, $\beta 1-3$, $\beta 1-4$ are always present, $\beta 1-5$ is found in 96% of the structures, $\beta 1-0$ in 44%. Strand $\beta 1-0$ is less frequent because it is very short, and it is in the region of variable secondary structure before helix A. Sheet $\beta 2$ is always present. In sheet $\beta 3$, strands $\beta 3-2$, $\beta 3-3$ are present in more than 98% of structures (if they are missing, it is because the corresponding residues are not modeled in the experimental structure). $\beta 3-1$ is sometimes not formed (present in 96%). Sheet $\beta 4$ is found in 91% of structures—sometimes it is not formed. The remaining two sheets are much less frequent— $\beta 5$ (25%) and $\beta 6$ (46%)—which can be related to their very short length.

Length of the SSEs. The typical length of individual helix classes varies substantially, from the minimal possible value of 3 residues (helices B", L') to 33 residues (helix I). On the other hand, β -strands in CYP structures are much shorter and range from 1 residue (strands $\beta 5-1$, $\beta 5-2$, $\beta 6-1$, $\beta 6-2$) to 5 residues on average (strands $\beta 1-1$ through $\beta 1-4$). The distribution of length of each SSE class is visualized by a violin plot in Fig. 3.

Helices. The helix SSE classes differ not only by their average length but there are also great differences in the length variability. Helices with the most uniform length are the helices in the core of the structure (C, E, I, K, L),

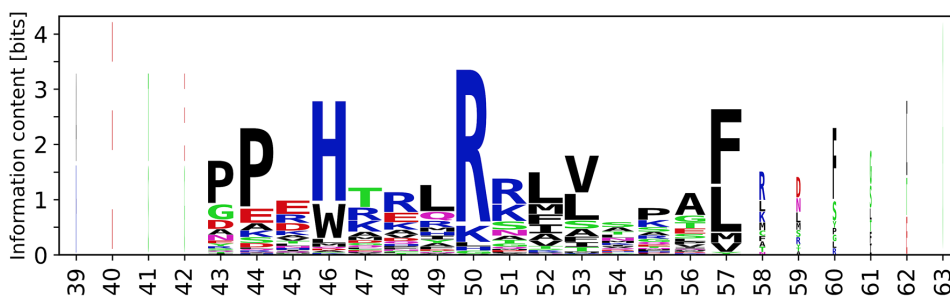


Figure 4. Sequence logo of helix C with generic numbering of residues. The remaining sequence logos are available in Supplementary Figures S3 and S4.

helices J' and K', and some very short helices on the edge of detection (B'', L', if they exist, they are very short (3–4 residues), so there is no space for variability).

Helices with the most variable length are the helices in the regions of variable secondary structure (A', A, B', F, F', G', G, K'').

Some helices have a bimodal distribution:

- For helix J, there are two peaks in the length distribution at 10 and 15 residues, with almost no samples in-between. This is because the length is different in bacteria (10 residues) and eukaryotes (15 residues); within these groups the length is very uniform (for details see section “Comparison of bacterial and eukaryotic CYPs”).
- For helix H, there are two peaks at 7–8 residues and 11–12 residues (not as nicely separated as in the case of helix J). In this case it cannot be easily explained as in the case of helix J (although there is a preference for shorter length in bacteria). The difference between the peaks roughly corresponds to one turn of α -helix; the lengths between the two peaks must be unstable, because in such case the protein chain would have to continue in the opposite direction.
- Some other helices (e.g. D, F) also have complex length distributions.

Strands. Sheet β 1 has around 5 residues in each strand, with the exception of the last strand (β 1-5), which has only 2 residues. Strand β 1-0, if present, has 1–2 residues.

Sheet β 2 has quite uniform length, in most cases 3 residues in each strand.

Sheet β 3 has more variable length, because of the differences between bacterial and eukaryotic structures (for details see section “Comparison of bacterial and eukaryotic CYPs”). In both cases, the middle strand β 3-2 is slightly longer, because it bridges β 3-1 and β 3-3.

Sheet β 4 is quite short (usually 1–3 residues per strand).

The remaining sheets are very short— β 5 usually contains 1–2 residues per strand, β 6 only 1 residue per strand.

Multiple sequence alignment and generic residue numbering in SSEs. For SSE classes with sufficient sequence conservation, we created sequence logos (see Fig. 4) and selected the most conserved residue as the reference residue. Based on the reference residue, we established a generic residue numbering scheme similar to the schemes used for GPCRs (described by Isberg et al.⁸ and used throughout GPCRdb³⁷) or immunoglobulins²⁰.

The reference residue is always numbered as @X.50, where X is the SSE label (character @ is added to avoid confusion in line notation). The remaining residues are then numbered correspondingly. An example of such residue identification is W120@C.46 in structure 2nnj (or generically @C.46), denoting the tryptophan residue 120 in helix C four positions before the reference residue, which is R124@C.50. Residue mutations can be also specified, e.g. W120A@C.46 (Fig. 4). Several examples of the usage of generic residue numbers can be found in section “Comparison of bacterial and eukaryotic CYPs”, demonstrating their usefulness.

We established the generic numbering for those SSEs, that contain at least one column in their logo with area (c_i) greater than 2 bits: major helices B, C, E, H, I, J, K, L, minor helices J', K', K'', and strands β 1-1, β 1-2, β 1-3, β 1-4, β 1-5, β 2-2, β 3-1, β 3-2. Other SSE classes have insufficient sequence conservation and/or sequence alignment does not correspond to structure alignment (i.e. reference residues do not align in 3D). Therefore, it is impossible to establish a meaningful generic numbering for these SSE classes. All sequence logos are available in Supplementary Figures S3 and S4.

These logos are computed only from the dataset of the available structures (Set-NR) and thus will differ from logos obtained from the alignment of all available sequences. Still, when compared to Pfam³⁸ representative proteome alignment (Supplementary Figure S5), they hold the key conserved residues and thus can be used for the definition of generic residue numbering that unlike Pfam takes into consideration generic spatial arrangement within SSE class.

5. MAIN PUBLICATIONS

SSE label	Residue range (in 2nnj)	Heme	Average information content	Gotoh 1992	Zawaira et al. 2011
K''	409–412		2.18		
β 1-5	96–97	✓	2.06		
β 6-1	362		1.98		
K	346–359	✓	1.98	SRS 5	SRS 5
β 6-2	477		1.87		
β 3-1	164		1.84		
L'	464–466		1.77		
L	438–455	✓	1.76		
J	317–331		1.76		
K'	391–396		1.70		
J'	339–345		1.67		
C	117–131		1.67		SRS 1
β 5-1	274		1.61		
β 1-4	368–369	✓	1.60	SRS 5	SRS 5
β 3-3	456–459		1.55		
β 2-1	374–376		1.55		
I	284–316	✓	1.54	SRS 4	SRS 4
β 5-2	280		1.53		
β 1-2	72–77		1.51		SRS 1'b
β 2-2	379–381		1.49		
G'	220–226		1.47		SRS 2
B	80–90		1.43		
β 3-2	485–489		1.41		
F'	211–219		1.38		SRS 2
H	263–274		1.37		
β 1-3	386–389		1.35		
β 1-1	64–69		1.31		SRS 1'b
E	166–183		1.22		
β 1-0	32		1.20		
D	141–159		1.20		
B''	112–114	✓	1.20		
β 4-2	478–479		1.18		SRS 6
A'	42–44		1.18		SRS 1'a
A	50–61		1.12		
G	227–254		1.04	SRS 3	SRS 3
F	192–209		0.86	SRS 2	SRS 2
B'	101–107	✓	0.73	SRS 1	SRS 1
β 4-1	473–474		0.70	SRS 6	SRS 6

Table 1. Comparison between the sequence conservation of the SSE classes (quantified by average information content) and the position of the substrate recognition sites from references^{17,18}. The SSEs are sorted from the most conserved to the most variable. The “Heme” column marks the SSEs with any atom within 8 Å from the heme cofactor.

Comparison with substrate recognition sites. We compared the sequence variability of the SSEs with the positions of the substrate recognition sites (SRS) in CYPs reported by Gotoh¹⁷ and Zawaira et al.¹⁸ (see Table 1). From this comparison, we observe that most SRS sites (SRS1, SRS2, SRS3, SRS6) are located in the SSEs with the most variable sequence. CYPs are known for high substrate variability and Table 1 shows that the substrate recognition sites are mirrored in the structural variability of their SSEs.

Two exceptions to this observation are SRS4 and SRS5, located in the highly or moderately conserved SSEs. This can be explained by their proximity to the heme cofactor—stabilization of the heme requires highly conserved amino acids in the neighboring SSEs.

Comparison of bacterial and eukaryotic CYPs. We compared the frequencies of occurrence of individual SSE classes between bacterial and eukaryotic structures (Fig. 5) as well as their distributions of length (Fig. 6).

Helices A', B', F', G', J' and K'' are significantly more frequent in eukaryotes. This is in agreement with literature^{14,23} reporting that F' and G' are typically not present in bacteria. However, sometimes a short helix

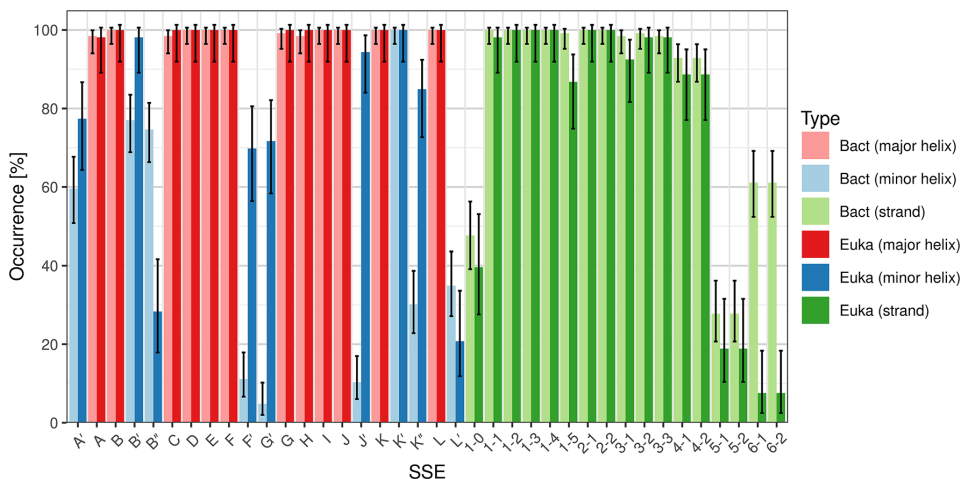


Figure 5. Comparison of SSE class occurrence in bacterial (Bact) and eukaryotic (Euka) CYP structures.

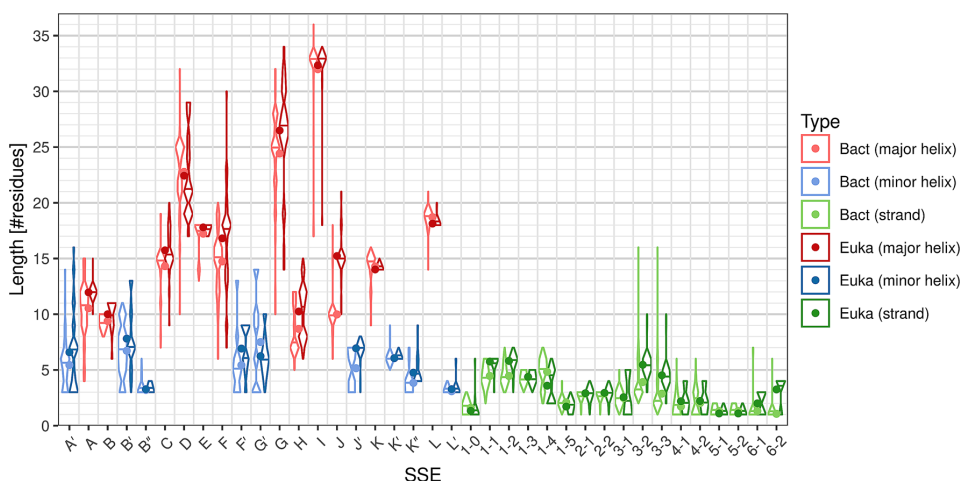


Figure 6. Comparison of SSE class length distribution in bacterial (Bact) and eukaryotic (Euka) CYP structures. The absent SSEs are not included in the distributions.

can be formed between F and G in bacterial structures, which is then automatically annotated as F' or G', so the observed occurrence is not equal to zero even in bacteria.

Conversely, helix B'', sheet β_6 and strand β_{1-5} occur more frequently in bacteria.

Helices A, A', C, F, G, H, J, J' and strands β_{1-1} , β_{1-2} , β_{3-2} , β_{3-3} , β_{6-2} are longer in eukaryotes, while strand β_{1-4} tends to be longer in bacteria. The case of β_{6-2} can be explained by the merging of β_{6-2} and β_{4-2} in some eukaryotic structures. The Kolmogorov–Smirnov test reports significant length difference for a few more SSE classes (B, K'' towards eukaryotes; D, K, L, β_{1-0} , β_{1-5} towards bacteria), however in these cases the mean length differs by less than one residue.

Complete results of the statistical tests, including the p -values, can be found in Supplementary Tables S2 and S3.

A notable difference is visible in the region of J and J' helices. Their typical length in eukaryotes is 15 and 7 residues, respectively, while in bacteria helix J has typically 10 residues and J' is not present at all. However, there is an anomalous group of 5 bacterial CYPs whose helices J and J' have lengths 15 and 7 residues, exactly as observed in eukaryotic CYPs. This group includes:

5. MAIN PUBLICATIONS

- the heme domain of the flavocytochrome P450 BM3 (*Bacillus megaterium*, PDB IDs 3kx3, 6h11 and 6h1t (these map to different UniProt IDs but have the same sequence of the heme domain)), where helix J' plays a role in the interaction with redox flavodomain,
- CYP 51 (*Mycobacterium tuberculosis*, PDB ID 2ci0),
- CYP 51 (*Methylococcus capsulatus*, PDB ID 6mcw),
- CYP 120A1 (*Synechocystis sp.*, PDB ID 3ve3),
- CYP 170A1 (*Streptomyces coelicolor*, PDB ID 3dbg).

We further investigated this anomalous bacterial group and discovered more similarities to the eukaryotic structures:

- The sequence of their helix J resembles the sequence of eukaryotic helixJ—most notably the glutamic acid E@J.59 (see section “[Multiple sequence alignment](#) and generic residue numbering in SSEs”) is highly conserved in both the eukaryotic CYPs (98%) and the anomalous group (100%) due to interactions with backbone amide groups of residues @K.40 and @K.41, while being much less conserved in the rest of the bacterial CYPs (7%), where this interaction is not observed.
- The region between helix K' and the heme binding site is approximately 9 residues longer in the eukaryotic and anomalous CYPs compared to the regular bacterial CYPs (roughly 39 residues in the eukaryotic and anomalous CYPs, 30 residues in the regular bacterial CYPs). Furthermore, in eukaryotic and anomalous CYPs, this region typically contains helix K'' (in 85% of the eukaryotic and in all anomalous CYPs), which is usually not present in the regular bacterial CYPs (only in 26%).

These deviations are concentrated in the region which has been reported as the interface for binding of the redox partner (for P450 BM3³⁹ and human mitochondrial CYP11A1⁴⁰). This suggests that these deviations may be functionally related to the interaction with the redox partner.

Generally, the bacterial and mitochondrial eukaryotic CYPs receive electrons from a small iron-sulfur protein, while the microsomal eukaryotic CYPs receive electrons from a flavoprotein, like NADPH-cytochrome P450 oxidoreductase (CPR)³⁹. Three of the five anomalous bacterial CYPs violate this interaction pattern: *Bacillus megaterium* P450 BM3 (aka CYP 102A1) contains flavodomain on C-terminus as a part of its sequence; *Mycobacterium tuberculosis* CYP 51 interacts with NADPH-hemoprotein reductase⁴¹; *Streptomyces coelicolor* CYP 170A1 is also known to be reduced by NADPH⁴². However, we have not found similar interactions with flavodomain for *Methylococcus capsulatus* CYP 51, but it belongs to a specific class containing Fe-4S ferredoxin-type on its C-terminus⁴³. We have found no information about interactions with redox partners for putative *Synechocystis sp.* CYP 120A1, but from the anomalous motif we can hypothesize interactions with some NADPH-hemoprotein reductase.

In some other aspects the anomalous CYPs behave as typical bacterial CYPs – there is no F' and G' helix in the FG-loop; the A-propionate side chain of the heme is oriented to the distal site (towards the substrate binding pocket). We can therefore hypothesize that this group represents evolutionary transition towards eukaryotic CYPs – this is also supported by the fact that the anomalous bacterial CYPs group with the eukaryotic sequences in the phylogenetic tree from Set-NR (see Supplementary Fig. S6).

SecStrAPI: how to get to our annotations. All annotations which are mentioned in this paper are publicly available through SecStrAPI at web address <https://webchem.ncbr.muni.cz/API/SecStr>.

The annotations can be downloaded directly (in JSON format, described in detail on the website) or can be accessed through PyMOL plugin *secstrapi_plugin.py*, which is available on the website and serves for simple and quick visualization of the SSE annotations.

Any cytochrome P450 structure, including new structures not included in our dataset, can be uploaded and annotated in our web application SecStrAnnotator Online at web address <https://sestra.ncbr.muni.cz>.

Limitations of the method. The presented methodology is in principle applicable to any protein family of interest. The workflow is almost fully automated—the bottleneck is the preparation of the annotation template. An appropriate template domain must be selected, and its annotation must be found in literature or created from scratch (especially in the families where no annotation conventions exist). However, we are currently developing software for automatic template generation, which will allow generalization of the annotation pipeline over all CATH protein families.

Another limitation is of course the fact that the inputs are experimental protein structures which are often incomplete. If an SSE is located in region which is not modeled in the experimental structure, then it will not be annotated and its observed occurrence in the family will be lower than its real occurrence. In the same way, this can affect the observed SSE length. This happens most often in the peripheral parts of the structure, in case of CYPs the FG-loop, BC-loop, HI-loop, JK-loop, sheet β 3, and N-terminal part. The experimental setup (e.g. crystallization conditions, resolution, refinement procedure) can also induce small structural variations, which may influence the exact length of the detected SSEs.

The natural diversity of the structures within a family can make it difficult to find the correct SSE annotation. For example, when a region in the template protein is occupied by a single helix but the equivalent region in another protein contains two helices, it might not be clear which of the two helices should be annotated. SecStrAnnotator will base the annotation on optimization of the overall score, which takes into account purely the structural information. In extreme cases, this automatic annotation can be different from the annotation

by an expert. We list the regions with the most uncertain annotation in section “Regions of variable secondary structure”. The annotation files contain the metric value for each SSE (higher values imply greater difference from the template and thus lower confidence of annotation).

A similar issue is related to the secondary structure assignment—an SSE can be so strongly deformed (kinked) that the two parts of the SSE will be assigned as two separate SSEs and only one part will be correctly annotated. This can be seen especially in the case of helix I (which is known to contain a kink¹⁴)—its typical length is 33 residues but in some structures we observe a length of 17–18 residues (i.e. only one part of the broken helix I).

Still all these complications are limited to a small number of marginal cases and they do not significantly affect the overall view of the family.

Conclusions

We presented a workflow for description of the secondary structure anatomy of a protein structural family—automatic annotation of secondary structure elements, analysis of their frequency of occurrence, typical length, position, amino acid composition and the variability of these properties.

We demonstrated these methods in the case study of the Cytochromes P450 (CYP) family. The characteristic SSEs of the family are 14 helices A, B, B', C, D, E, F, G, H, I, J, K, K', L and 4 sheets β 1, β 2, β 3, β 4, which occur nearly in all structures. Optional SSEs include helices A', B'', F', G', J', K'', L', sheets β 5, β 6, and strand β 1-0. Some of these SSE classes are very uniform in length (the core helices C, E, I, K, L, but also J', K'), while some show extensive length variation (A', A, B', D, F, F', G', G, H, J, K'', β 3-2, β 3-3). The shortest helices B'', L' and sheets β 5, β 6 are on the edge of detection.

For the SSE classes with sufficient sequence conservation, we have established a generic residue numbering scheme (e.g. W120^{6C.46}) similar to that used for the GPCR family. Unfortunately, this was not possible in the variable regions (A, B', D, F, G), which are responsible for substrate uptake and substrate recognition.

We also compared the eukaryotic and bacterial members of the CYP family. The most substantial difference is the absence of helices F', G', J' in bacteria. Helices A', B', K'' are also rarer in bacteria, while B'', β 6 and β 1-5 are more common in bacteria than in eukaryotes. Many SSEs tend to be longer in either eukaryotes (A, A', E, G, H, J, J', β 1-1, β 1-2, β 3-2, β 3-3, β 6-2) or bacteria (β 1-4).

Strikingly, we also identified a small group of 5 bacterial CYPs with typical eukaryotic features in the region of helices J–L, which can be explained by the interaction with the redox partner.

Automatic annotation of CYP SSEs allows not only the orientation in the structure but also among its channels, which play a crucial role in substrate recognition and product egress.

All the utilized software tools and the obtained data, including secondary structure annotations and generic residue numbers, are available at our website (<https://webchem.ncbr.muni.cz/Wiki/SecStrAnnotator>). The annotations can be easily visualized with a PyMOL plugin. CYP structures can be also annotated by SecStrAnnotator Online at <https://sestra.ncbr.muni.cz>. While the annotations are now only specific for the CYP family, the software is applicable in principle to all protein families with defined annotations. We are currently working on the generalization of the annotation pipeline over all CATH protein families, but we are open for submissions of annotation templates for other protein families from the community.

Methods

Datasets. *Set-NR.* A list of protein domains annotated as Cytochrome P450 was obtained from SIFT's resource⁴⁴ via PDB REST API (<https://www.ebi.ac.uk/pdbe/api/>, /mappings endpoint) on 7 July 2020. More specifically, annotations originating from databases CATH and Pfam (accessions 1.10.630.10 and PF00067) were merged to obtain 1855 protein domains located in 1012 PDB entries. The information about residue ranges was discarded and whole chains were taken instead (this was necessary because Pfam often wrongly annotates only a small portion of the chain). The domains were mapped to UniProt IDs and the best-quality domain was selected for each UniProt ID. The quality was measured by “overall_quality” obtained from PDB REST API (/validation/summary_quality_scores/entry endpoint). The domains which map to no UniProt ID were excluded. The resulting *Set-NR* (non-redundant) contains 183 protein domains.

Set-NR-Bact and *Set-NR-Euka.* Domains from *Set-NR* were mapped to their source organism using PDB REST API (/pdb/entry/molecules endpoint) and divided into four subsets based on their superkingdom using NCBI Taxonomy⁴⁵: *Set-NR-Bact* (Bacteria, 126 structures), *Set-NR-Euka* (Eukaryota, 53 structures), *Set-NR-Arch* (Archaea, 3 structures) and *Set-NR-Viru* (Viruses, 1 structure). However, *Set-NR-Arch* and *Set-NR-Viru* were not analyzed separately because of their small size.

All data, including the lists of PDB IDs and UniProt IDs for each dataset, are available in the Zenodo repository⁴⁶.

Template annotation. Since SecStrAnnotator requires an annotated template structure, we have chosen a template domain based on multiple selection criteria.

First, the template should contain all SSE classes. Thus, we considered only the eukaryotic structures (*Set-NR-Euka*).

Second, the template structure should be an “average” structure which is as similar to all the others as possible. Therefore, we compared each pair of structures in *Set-NR-Euka* by *cealign* command in PyMOL and calculated pairwise Q-scores⁴⁷. Then for each structure we calculated the average Q-score against all the other structures Q_{avg} . We selected the structure with the highest Q_{avg} as the template domain, which was 2nnjA (human CYP 2C8).

5. MAIN PUBLICATIONS

Third, template structure should have sufficient resolution, quality and should not contain unmodeled loops etc. The selected domain 2nnjA meets these criteria (resolution 2.28 Å, overall quality 42.46, observed residue range 10–472 covers the whole region of interest).

Secondary structure annotation was mapped from the annotation of CYP 2C9 by Rowland et al.¹⁰ with several added SSE classes, as described in section “Traditional SSE nomenclature in the CYP family” and shown in Fig. 1a. General methods for the selection and annotation of the template with and without any prior knowledge can be found our earlier publication²¹.

Annotation procedure. The annotation was performed using our software SecStrAnnotator. The current version 2.2 has been improved since the original publication²¹ of SecStrAnnotator 1.0, the most significant changes being the support for mmCIF files, label_* numbering, revised secondary structure assignment method (geom-hbond2), and detection of structural irregularities within SSEs. Switching from .NET Framework to .NET Core enabled more consistent usage across operating systems. Furthermore, many additional scripts have been added, thus creating the SecStrAnnotator Suite and facilitating the automation of the whole analysis pipeline (including automatic selection of the non-redundant set, sequence alignment, generic residue numbering, sequence logo visualization, statistical evaluation, visualization through a PyMOL plugin). To overcome the need of installation, we also introduced the SecStrAnnotator Online and SecStrAPI with precomputed annotation results.

The annotation algorithm consists of three main steps: secondary structure assignment, structural alignment and SSE matching. Detailed description is provided in Midlik et al.²¹. SecStrAnnotator was run with these settings: `-ssa geom-hbond2 -align cealign -matching mom -soft -maxmetric 25,0.5,0.5 -label2auth -verbose`.

Generation of user-defined template is possible either running desktop version with `-onlyssa` option or by using SecStrAnnotator Online in SSA-only mode (by selecting Template: None). The output of this mode is a detected.sses.json file which contains only generically labelled SSEs (e.g. H0, H1, H2, E3, E4...), but can in fact be uploaded and used as a template for other proteins from the same family. The user can also manually change the labels in this file and/or remove unwanted SSEs (when renaming/removing β -strands, please pay attention to also changing the β -connectivity section) or even further improve this template as described in Midlik et al.²¹.

Statistical evaluation. All statistical tests were performed in R (version 3.4.4-1ubuntu1) using the *stats* library. The plots were generated in R using the *ggplot2* library.

Comparison of bacterial and eukaryotic dataset. The occurrence of each SSE class in Set-NR-Bact and Set-NR-Euka was modeled as a binomial distribution, and the two datasets were compared by the test of equal proportions (*prop.test*) with $\alpha=0.05$.

The distribution of length (number of residues) of each SSE class was compared between Set-NR-Bact and Set-NR-Euka. Where the medians of the eukaryotic and bacterial distribution were not equal, the two-sample Kolmogorov–Smirnov test (*ks.test*) with $\alpha=0.05$ was used to decide if the difference between the distributions is significant. Non-existing SSEs were not included in the length distributions.

Multiple sequence alignment. The amino acid sequences of the individual SSE classes were extracted from Set-NR and aligned using an in-house algorithm NoGapAligner which allows gaps only at the beginning and at the end but not within a sequence (this is necessary in order to establish generic residue numbering). Substitution matrix was BLOSUM62 and the gap penalty was set to 10. Sequence logos were rendered using *logomaker* module for Python⁴⁸.

For every position i in each multiple sequence alignment, the information content R_i and the conservation measure c_i were calculated as follows:

$$R_i = \log_2 20 + \sum_{a \in A} \frac{p_i^a}{p_i} \log_2 \frac{p_i^a}{p_i} \quad (1)$$

$$c_i = p_i R_i \quad (2)$$

where A is the set of 20 standard amino acids, p_i^a is the fraction of sequences having amino acid a at position i , p_i is the fraction of sequences having any amino acid (not a gap) at position i ⁴⁹. In the visual form of a logo, p_i and R_i correspond to the width and height of the i -th column, thus c_i corresponds to the area of the column. R_i is expressed in bits and its values can range from 0, for a position with 20 equiprobable amino acids, to approximately 4.3 ($\log_2 20$), for a position with one perfectly conserved amino acid. The position with the greatest c_i within the alignment was selected as the reference residue of the SSE class.

To be able to compare the overall conservation of individual SSE classes, we computed the average information content R_{avg} (i.e. average column height) of each logo as:

$$R_{\text{avg}} = \frac{\sum_{i=1}^n p_i R_i}{\sum_{i=1}^n p_i} \quad (3)$$

where n is the number of positions in the logo.

Data availability

The datasets generated and analyzed during the current study are available in the Zenodo repository (<https://doi.org/10.5281/zenodo.3939133>)⁴⁶ and as a part of the SecStrAnnotator manual (<https://webchem.ncbr.muni.cz/Wiki/SecStrAnnotator>).

Received: 30 September 2020; Accepted: 21 May 2021

Published online: 11 June 2021

References

- Pauling, L., Corey, R. B. & Branson, H. R. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci.* **37**, 205–211 (1951).
- Pauling, L. & Corey, R. B. Configurations of polypeptide chains with favored orientations around single bonds: two new pleated sheets. *Proc. Natl. Acad. Sci. U. S. A.* **37**, 729–740 (1951).
- Sillitoe, I. *et al.* CATH: increased structural coverage of functional space. *Nucleic Acids Res.* **49**, D266–D273 (2021).
- Fox, N. K., Brenner, S. E. & Chandonia, J.-M. SCOPe: Structural Classification of Proteins-extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res.* **42**, D304–309 (2014).
- Thornton, J. M., Todd, A. E., Milburn, D., Borkakoti, N. & Orengo, C. A. From structure to function: approaches and limitations. *Nat. Struct. Biol.* **7**(Suppl), 991–994 (2000).
- Krejci, E., Duval, N., Chatonnet, A., Vincens, P. & Massoulié, J. Cholinesterase-like domains in enzymes and structural proteins: functional and evolutionary relationships and identification of a catalytically essential aspartic acid. *Proc. Natl. Acad. Sci. U. S. A.* **88**, 6647–6651 (1991).
- Lenfant, N. *et al.* ESTHER, the database of the α/β -hydrolase fold superfamily of proteins: tools to explore diversity of functions. *Nucleic Acids Res.* **41**, D423–429 (2013).
- Isberg, V. *et al.* Generic GPCR residue numbers—aligning topology maps while minding the gaps. *Trends Pharmacol. Sci.* **36**, 22–31 (2015).
- Ehrenmann, F., Kaas, Q. & Lefranc, M.-P. IMGT/3Dstructure-DB and IMGT/DomainGapAlign: a database and a tool for immunoglobulins or antibodies, T cell receptors, MHC, IgSF and MhcSF. *Nucleic Acids Res.* **38**, D301–307 (2010).
- Rowland, P. *et al.* Crystal structure of human cytochrome P450 2D6. *J. Biol. Chem.* **281**, 7614–7622 (2006).
- Cojocar, V., Winn, P. J. & Wade, R. C. The ins and outs of cytochrome P450s. *Biochim. Biophys. Acta* **1770**, 390–401 (2007).
- Hendrychova, T., Berka, K., Navratilova, V., Anzenbacher, P. & Otyepka, M. Dynamics and hydration of the active sites of mammalian cytochromes P450 probed by molecular dynamics simulations. *Curr. Drug Metab.* **13**, 177–189 (2012).
- Yu, X., Cojocar, V. & Wade, R. C. Conformational diversity and ligand tunnels of mammalian cytochrome P450s. *Biotechnol. Appl. Biochem.* **60**, 134–145 (2013).
- Otyepka, M., Skopalik, J., Anzenbacherová, E. & Anzenbacher, P. What common structural features and variations of mammalian P450s are known to date?. *Biochim. Biophys. Acta* **1770**, 376–389 (2007).
- Otyepka, M., Berka, K. & Anzenbacher, P. Is there a relationship between the substrate preferences and structural flexibility of cytochromes P450?. *Curr. Drug Metab.* **13**, 130–142 (2012).
- Urban, P., Lautier, T., Pompon, D. & Truan, G. Ligand access channels in cytochrome P450 enzymes: a review. *Int. J. Mol. Sci.* **19**, 1617 (2018).
- Gotoh, O. Substrate recognition sites in cytochrome P450 family 2 (CYP2) proteins inferred from comparative analyses of amino acid and coding nucleotide sequences. *J. Biol. Chem.* **267**, 83–90 (1992).
- Zawaira, A., Ching, L. Y., Coulson, L., Blackburn, J. & Wei, Y. C. An expanded, unified substrate recognition site map for mammalian cytochrome P450s: analysis of molecular interactions between 15 mammalian CYP450 isoforms and 868 substrates. *Curr. Drug Metab.* **12**, 684–700 (2011).
- Schrödinger, LLC. The PyMOL Molecular Graphics System, Version 2.3. (2015).
- Dunbar, J. & Deane, C. M. ANARCI: antigen receptor numbering and receptor classification. *Bioinformatics* **32**, 298–300 (2016).
- Midlik, A. *et al.* Automated family-wide annotation of secondary structure elements. *Methods Mol. Biol.* **1958**, 47–71 (2019).
- Peterson, J. A. & Graham, S. E. A close family resemblance: the importance of structure in understanding cytochromes P450. *Structure* **6**, 1079–1085 (1998).
- Johnson, E. F. & Stout, C. D. Structural diversity of eukaryotic membrane cytochrome p450s. *J. Biol. Chem.* **288**, 17082–17090 (2013).
- Poulos, T. L., Finzel, B. C., Gunsalus, I. C., Wagner, G. C. & Kraut, J. The 2.6-Å crystal structure of *Pseudomonas putida* cytochrome P-450. *J. Biol. Chem.* **260**, 16122–16130 (1985).
- Poulos, T. L., Finzel, B. C. & Howard, A. J. High-resolution crystal structure of cytochrome P450cam. *J. Mol. Biol.* **195**, 687–700 (1987).
- Ravichandran, K. G., Boddupalli, S. S., Hasermann, C. A., Peterson, J. A. & Deisenhofer, J. Crystal structure of hemoprotein domain of P450BM-3, a prototype for microsomal P450s. *Science* **261**, 731–736 (1993).
- Hasemann, C. A., Ravichandran, K. G., Peterson, J. A. & Deisenhofer, J. Crystal structure and refinement of cytochrome P450terp at 2.3 Å resolution. *J. Mol. Biol.* **236**, 1169–1185 (1994).
- Park, S. Y. *et al.* Crystal structure of nitric oxide reductase from denitrifying fungus *Fusarium oxysporum*. *Nat. Struct. Biol.* **4**, 827–832 (1997).
- Scott, E. E. *et al.* An open conformation of mammalian cytochrome P450 2B4 at 1.6-Å resolution. *Proc. Natl. Acad. Sci. U. S. A.* **100**, 13196–13201 (2003).
- Wester, M. R. *et al.* Structure of a substrate complex of mammalian cytochrome P450 2C5 at 2.3 Å resolution: evidence for multiple substrate binding modes. *Biochemistry* **42**, 6370–6379 (2003).
- Ouellet, H., Podust, L. M. & de Montellano, P. R. O. Mycobacterium tuberculosis CYP130: crystal structure, biophysical characterization, and interactions with antifungal azole drugs. *J. Biol. Chem.* **283**, 5069–5080 (2008).
- Hasemann, C. A., Kurumbail, R. G., Boddupalli, S. S., Peterson, J. A. & Deisenhofer, J. Structure and function of cytochromes P450: a comparative analysis of three crystal structures. *Structure* **3**, 41–62 (1995).
- Pylypenko, O., Vitali, F., Zerbe, K., Robinson, J. A. & Schlichting, I. Crystal structure of OxyC, a cytochrome P450 implicated in an oxidative C-C coupling reaction during vancomycin biosynthesis. *J. Biol. Chem.* **278**, 46727–46733 (2003).
- Williams, P. A. *et al.* Crystal structures of human cytochrome P450 3A4 bound to metyrapone and progesterone. *Science* **305**, 683–686 (2004).
- Berka, K., Palončyová, M., Anzenbacher, P. & Otyepka, M. Behavior of human cytochromes P450 on lipid membranes. *J. Phys. Chem. B* **117**, 11556–11564 (2013).
- Agresti, A. & Coull, B. A. Approximate is better than “exact” for interval estimation of binomial proportions. *Am. Stat.* **52**, 119–126 (1998).
- Pándy-Szekerés, G. *et al.* GPCRdb in 2018: adding GPCR structure models and ligands. *Nucleic Acids Res.* **46**, D440–D446 (2018).
- Mistry, J. *et al.* Pfam: The protein families database in 2021. *Nucleic Acids Res.* **49**, D412–D419 (2021).

39. Sevrioukova, I. F., Li, H., Zhang, H., Peterson, J. A. & Poulos, T. L. Structure of a cytochrome P450-redox partner electron-transfer complex. *Proc. Natl. Acad. Sci. U. S. A.* **96**, 1863–1868 (1999).
40. Strushkevich, N. *et al.* Structural basis for pregnenolone biosynthesis by the mitochondrial monooxygenase system. *Proc. Natl. Acad. Sci. U. S. A.* **108**, 10139–10143 (2011).
41. Bellamine, A., Mangla, A. T., Nes, W. D. & Waterman, M. R. Characterization and catalytic properties of the sterol 14 α -demethylase from *Mycobacterium tuberculosis*. *Proc. Natl. Acad. Sci.* **96**, 8937–8942 (1999).
42. Zhao, B. *et al.* Biosynthesis of the sesquiterpene antibiotic albaflavenone in *Streptomyces coelicolor* A3(2). *J. Biol. Chem.* **283**, 8183–8189 (2008).
43. Jackson, C. J. *et al.* A novel sterol 14 α -demethylase/ferredoxin fusion protein (MCCYP51FX) from *Methylococcus capsulatus* represents a new class of the cytochrome P450 superfamily. *J. Biol. Chem.* **277**, 46959–46965 (2002).
44. Dana, J. M. *et al.* SIFTS: updated structure integration with function, taxonomy and sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Res.* **47**, D482–D489 (2019).
45. Sayers, E. W. *et al.* Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **37**, D5–15 (2009).
46. Midlik, A. *et al.* Annotation and analysis of the secondary structure elements in the Cytochrome P450 protein family. *Zenodo* (2020). <https://doi.org/10.5281/zenodo.3939133>.
47. Krissinel, E. & Henrick, K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr. D Biol. Crystallogr.* **60**, 2256–2268 (2004).
48. Tareen, A. & Kinney, J. B. Logomaker: beautiful sequence logos in Python. *Bioinformatics* **36**, 2272–2274 (2020).
49. Schneider, T. D. & Stephens, R. M. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res.* **18**, 6097–6100 (1990).

Acknowledgements

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the project CEITEC 2020 [LQ1601]; ELIXIR-CZ research infrastructure project including access to computing and storage facilities [LM2018131]; European Regional Development Fund-projects ELIXIR-CZ [CZ.02.1.01/0.0/0.0/16_013/0001777]. A.M. was also supported by Brno Ph.D. Talent Scholarship funded by Brno City Municipality. We would like to thank Veronika Bendová for her valuable advice on the statistical procedures.

Author contributions

R.S., K.B., and J.K. conceived, designed, and cooperated the study. A.M. created the software, analyzed data and made almost all Figures. T.R.M. and V.N. tested the software. T.R.M. performed sequence analyses and Figure S6. A.M., V.N., K.B., and R.S. interpreted the results. A.M., R.S., and K.B. wrote the manuscript. All authors reviewed and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-021-91494-8>.

Correspondence and requests for materials should be addressed to R.S. or K.B.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021

SUPPLEMENTARY INFORMATION**Uncovering of cytochrome P450 anatomy by SecStrAnnotator**

Adam Midlik^{1,2}, Veronika Navrátilová^{3,#}, Taraka Ramji Moturu^{1,2}, Jaroslav Koča^{1,2}, Radka Svobodová^{1,2,*}, Karel Berka^{3,*}

¹ CEITEC – Central European Institute of Technology, Masaryk University, Brno 625 00, Czech Republic

² National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Brno 625 00, Czech Republic

³ Department of Physical Chemistry, Faculty of Science, Palacký University, Olomouc 771 46, Czech Republic

Current address: Pharmacovigilance department, State Institute for Drug Control, Praha 100 41, Czech Republic

* Corresponding authors

E-mail: radka.svobodova@ceitec.muni.cz, karel.berka@upol.cz

Supplementary Table S1. Residue ranges of the SSEs in the PDB entry 2nnj (template annotation)

SSE label	Residue range	SSE group
A'	42–44	minor helix
A	50–61	major helix
B	80–90	major helix
B'	101–107	minor helix
B''	112–114	minor helix
C	117–131	major helix
D	141–159	major helix
E	166–183	major helix
F	192–209	major helix
F'	211–219	minor helix
G'	220–226	minor helix
G	227–254	major helix
H	263–274	major helix
I	284–316	major helix
J	317–331	major helix
J'	339–345	minor helix
K	346–359	major helix
K'	391–396	minor helix
K''	409–412	minor helix
L	438–455	major helix
L'	464–466	minor helix
β1-0	32	strand
β1-1	64–69	strand
β1-2	72–77	strand
β1-3	386–389	strand
β1-4	368–369	strand
β1-5	96–97	strand
β2-1	374–376	strand
β2-2	379–381	strand
β3-1	164	strand
β3-2	485–489	strand
β3-3	456–459	strand
β4-1	473–474	strand
β4-2	478–479	strand
β5-1	274	strand
β5-2	280	strand
β6-1	362	strand
β6-2	477	strand

Supplementary Note: Structural irregularities

Introduction

Helices found in protein structures are traditionally distinguished into three types: 3_{10} -helix, α -helix and π -helix, characterized by repetitive $i+3 \rightarrow i$, $i+4 \rightarrow i$ and $i+5 \rightarrow i$ backbone hydrogen bonds, respectively. However, the 3_{10} and π -helices are much less common than the α -helix and rarely span more than a few residues. Combinations of the hydrogen bonding patterns commonly occur in a single helical segment, such as 3_{10} - α - 3_{10} or α - π - α ¹. Therefore, we can understand the α -helix as the standard structural pattern and the 3_{10} and π -helices as structural irregularities within this pattern.

A β -bulge is a region of irregularity in a β -sheet formed by two or more residues on one strand (long side) opposite a single residue on the other strand (short side). β -bulges are relatively frequent (on average two instances per protein) and occur primarily between antiparallel strands^{2,3}.

Detection of structural irregularities

The traditional (DSSP) distinction of helix types is based on the type of hydrogen bonds stabilizing the helix (type 3_{10} : $i+3 \rightarrow i$, type α : $i+4 \rightarrow i$, type π : $i+5 \rightarrow i$ bonds). A DSSP helix is detected when there are at least two consecutive hydrogen bonds of the same type⁴.

SecStrAnnotator uses a method for helix detection which focuses on the geometry of the protein backbone and allows abstraction from these hydrogen bonding patterns. However, it also reports the hydrogen bonds found in each helix (when run with `--verbose`).

The *contained types* of such helix are then determined by the occurrence of two consecutive hydrogen bonds of the same type (i.e. a DSSP helix) within the helix. A helix may contain 3_{10} , α , π , or any combination of these types (helices not containing any type are rejected).

It should be kept in mind that all obtained results are based on the DSSP definition of a hydrogen bond, which is approximate and quite benevolent.

Occurrences of irregularities

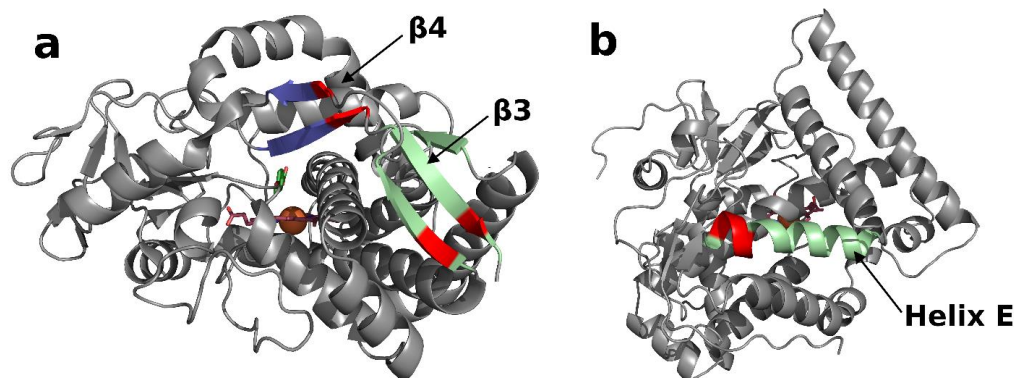
Beta-bulges

We analysed the frequency of occurrence of β -bulges in individual β -sheets and found out that they are not distributed randomly but occur mostly in sheets $\beta 3$ and $\beta 4$ (see Supplementary Fig. S1). The most common are:

- classic β -bulge on sheet $\beta 4$, with the long side in $\beta 4$ -2 and the short side in $\beta 4$ -1 (in 20.8% of the structures)
- classic β -bulge on sheet $\beta 3$, with the long side in $\beta 3$ -3 and the short side in $\beta 3$ -2 (in 8.2% of the structures)

Bulges of other types occur rarely (less than 5% structures for each type). Bulges in sheet $\beta 1$ occur in less than 5% structures; they are never found in sheets $\beta 2$, $\beta 5$, $\beta 6$.

The β -bulges are much more common in the bacterial than the eukaryotic structures. Namely, the bulge on sheet $\beta 4$ is found in 28.6% bacterial and 1.9% eukaryotic structures; the bulge on sheet $\beta 3$ is found in 8.7% bacterial and 3.8% eukaryotic structures. In archaeal structures, sheets $\beta 3$ and $\beta 4$ are usually merged into a single sheet containing two or more bulges.



Supplementary Figure S1. Location of the most common structural irregularities. (a) The β -bulges in the bacterial CYP199A2 (PDB ID 4dnj); sheet $\beta 3$ is shown in green, sheet $\beta 4$ in blue, the bulges are highlighted in red. (b) The π -helix within helix E in the bacterial CYP142A2 (PDB ID 4uax); helix E is shown in green, the π -helix is highlighted in red.

3₁₀-helices and π -helices

A helix may consist of a single helix type (3_{10} , α , π) or may contain any combination of these basic types. We studied how often each of these types occurs in individual annotated helices.

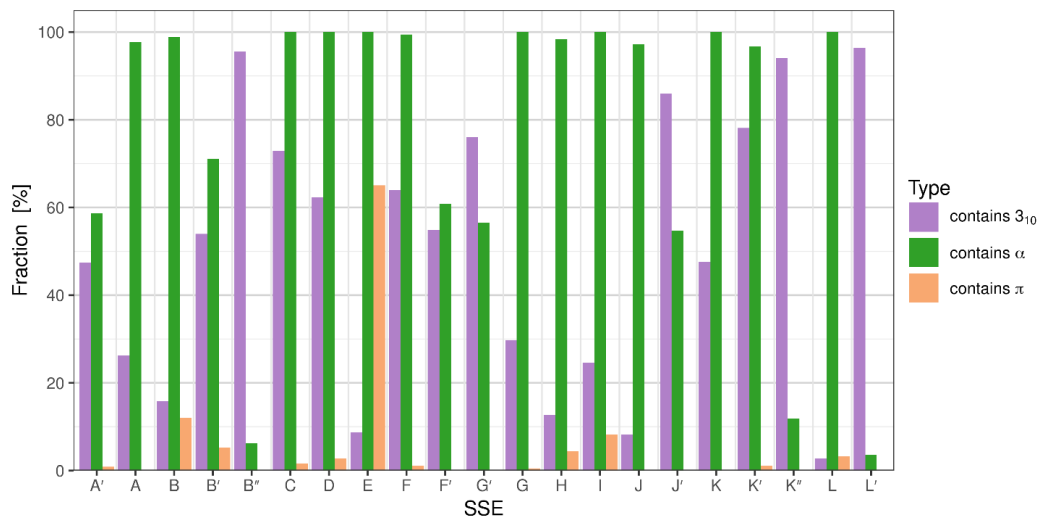
The α -helix is of course the most abundant type and is present in 86.4% of all studied helices.

43.3% of all studied helices contain a 3_{10} -helix. Most of these 3_{10} -helices occur in the shortest minor helices, which are typically pure 3_{10} (L', B'', K''), followed by J', K', and G'. Major helices with the highest content of 3_{10} -helices are C, D, and F. In contrast, helices with the lowest occurrence of 3_{10} -helical parts are L, E, and J (under 10%).

The π -helices are far less abundant than 3_{10} -helices – only 6.3% of all helices contain a π -helix. The π -helices very often occur as a part of helix E (in 65.0% cases), followed by helix B (12.0%), helix I (8.2%), and helix B' (5.3%). In other helices, their occurrence is under 5% (see Supplementary Fig. S2).

It is an interesting discovery that in 65.0% structures helix E contains a π -helix, and this fact might be related to the function or stability of the structures. This π -helix is typically located near the N-terminus of helix E (see Supplementary Fig. S1), and its occurrence is much higher in bacteria (87.3%) than in eukaryotes (9.4%). In the case of helix B this tendency is reversed (41.5% in eukaryotes, 0.0% in bacteria).

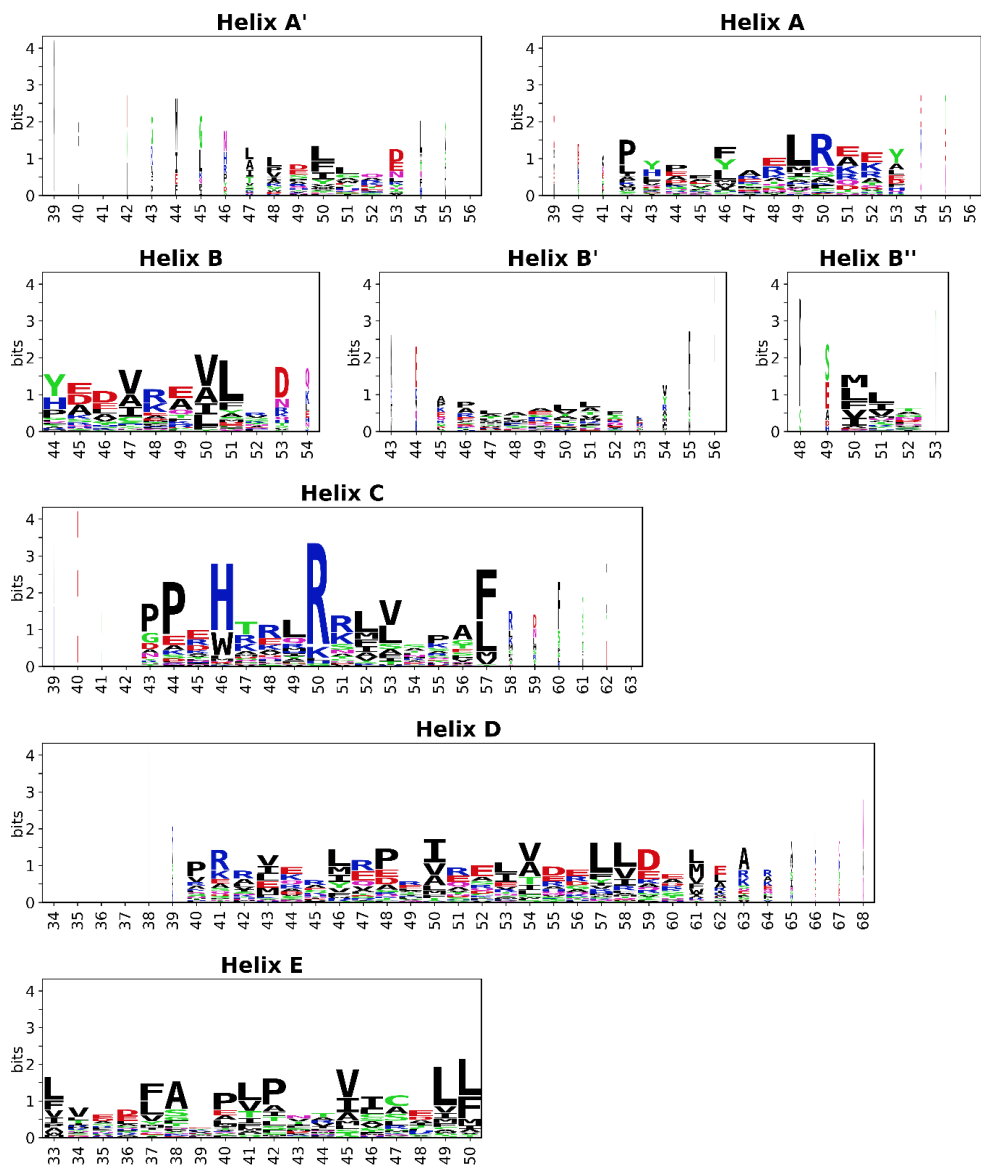
5. MAIN PUBLICATIONS



Supplementary Figure S2. Percentage of helix types – 3₁₀, α and π.

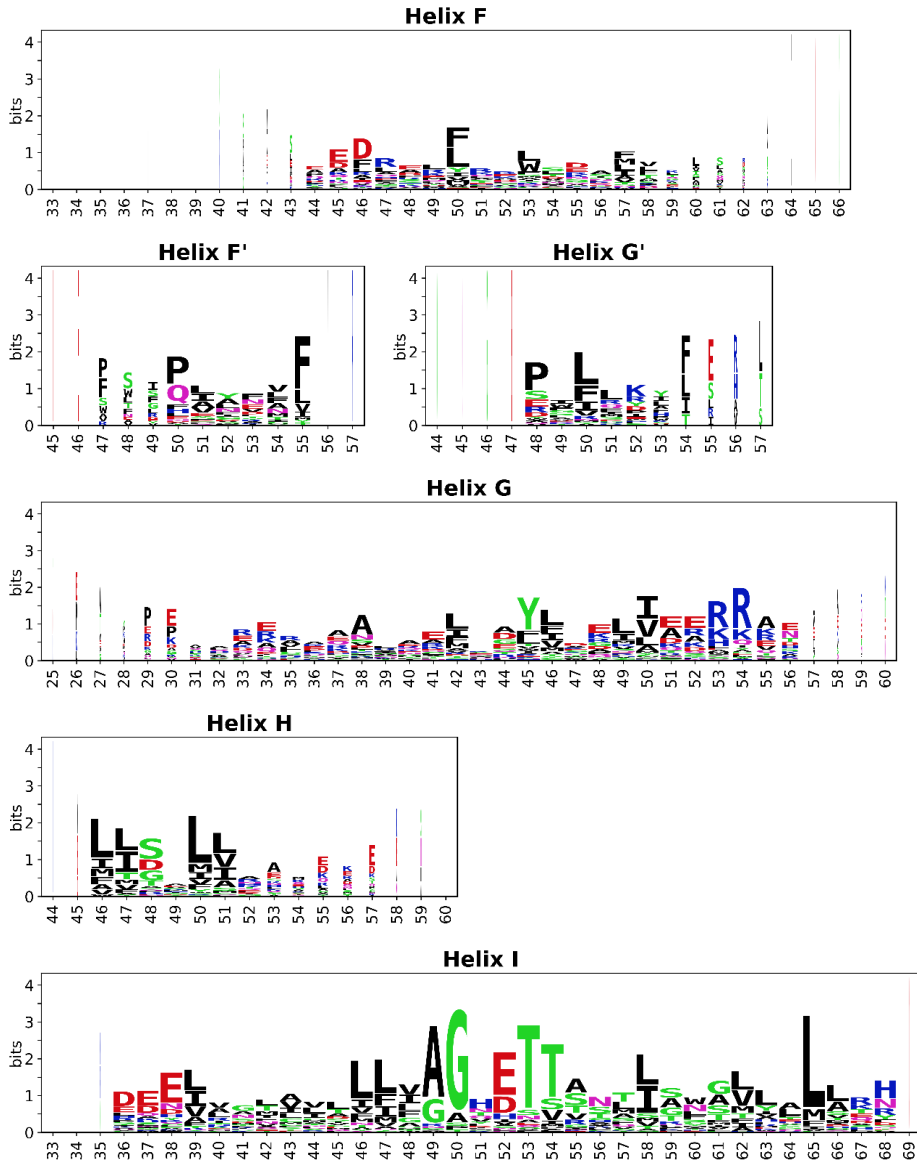
Sequence logos

Helices

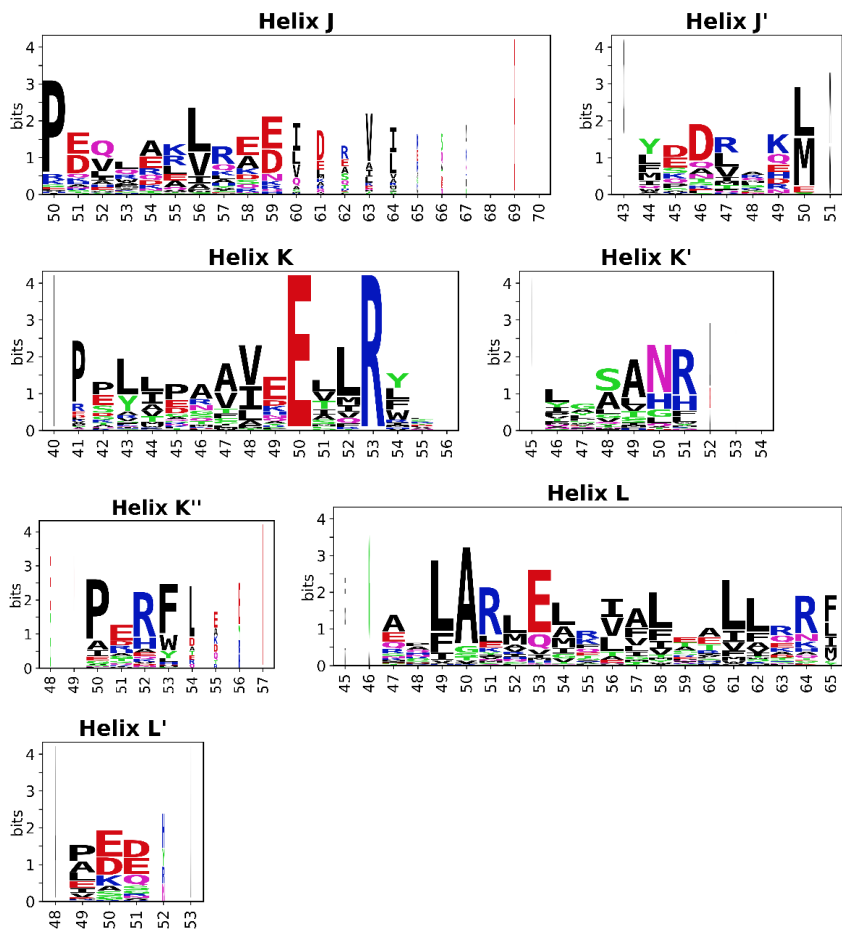


Supplementary Figure S3. Sequence logos for the helices. (Continues on the next page.)

5. MAIN PUBLICATIONS

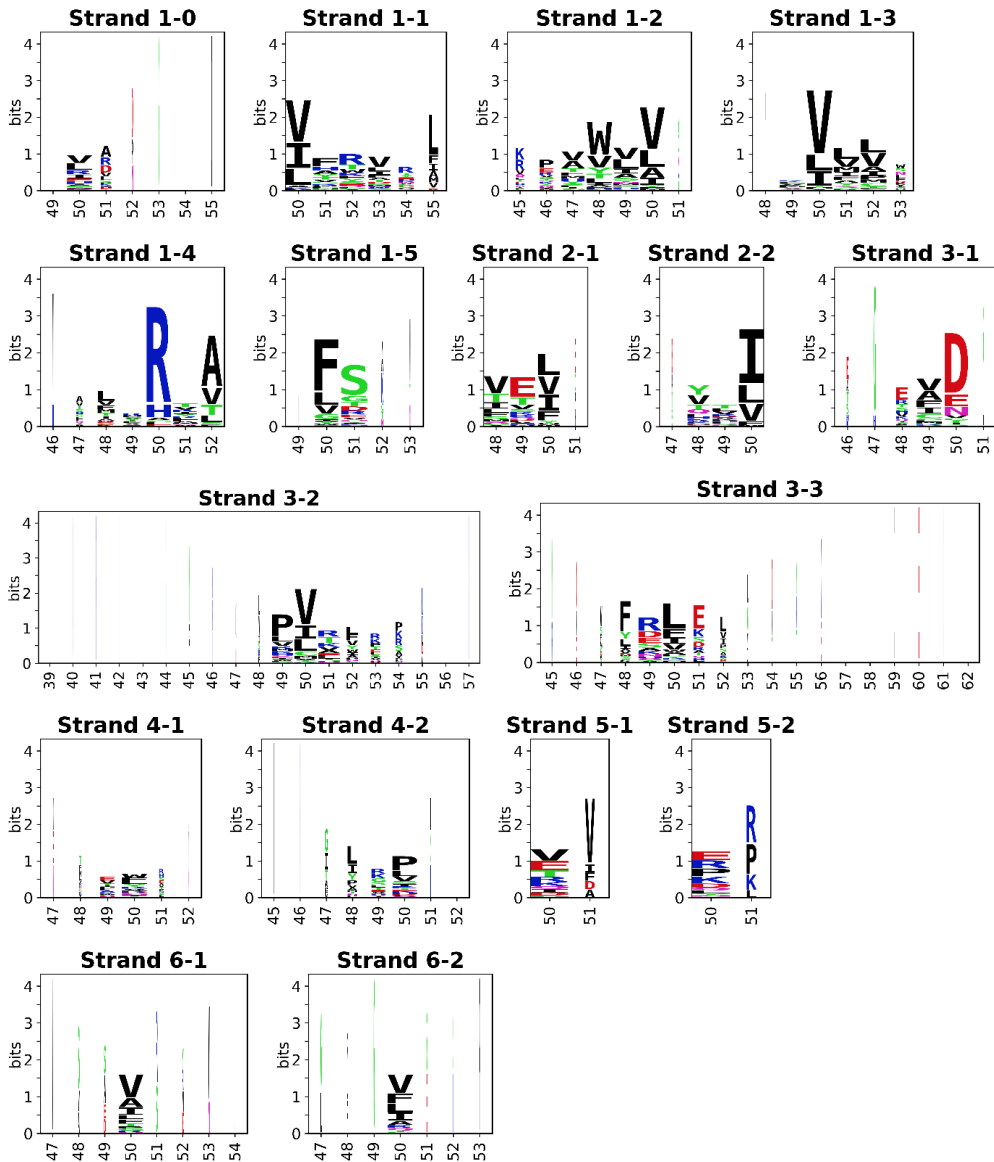


Supplementary Figure S3 (continued). Sequence logos for the helices. (Continues on the next page.)



Supplementary Figure S3 (continued). Sequence logos for the helices.

Strands



Supplementary Figure S4. Sequence logos for the β -strands.

Supplementary Table S2. Comparison of the SSE occurrences in the bacterial and eukaryotic CYP structures.

Results of the test of equal proportions comparing SSE occurrences in Set-NR-Bact vs Set-NR-Euka. The column "Comparison" marks the significant differences at confidence level $\alpha = 0.05$ (>: higher occurrence in Bacteria, <: higher occurrence in Eukaryota).

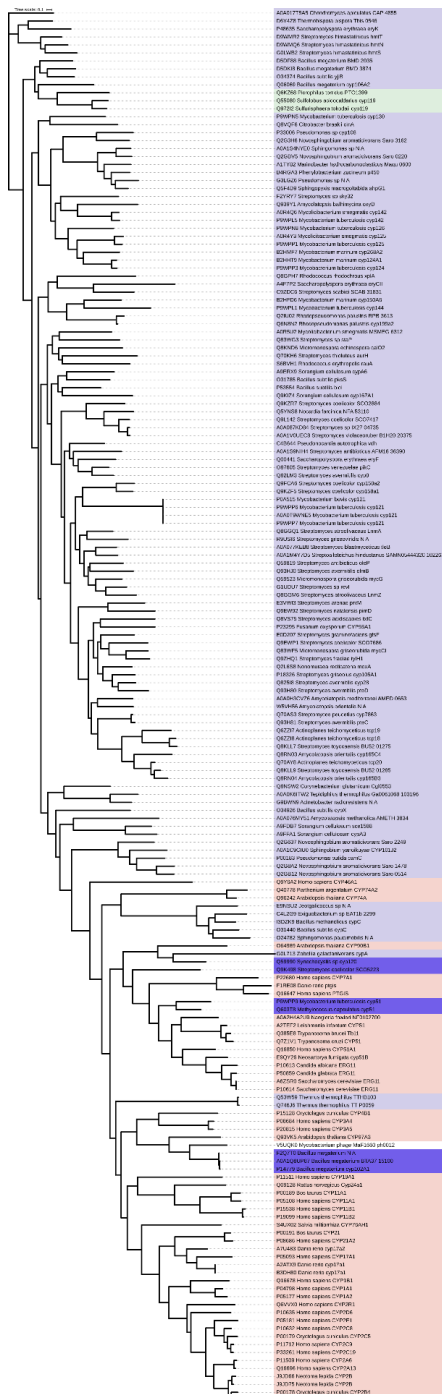
SSE label	Occurrence (Set-NR-Bact)	Comparison	Occurrence (Set-NR-Euka)	Occurrence difference	p-value
A'	0.595	<	0.774	0.180	0.035
A	0.984		0.981	-0.003	1
B	1.000		1.000	0.000	NaN
B'	0.770	<	0.981	0.210	0.0012
B''	0.746	>	0.283	-0.460	1.8E-08
C	0.984		1.000	0.016	0.89
D	1.000		1.000	0.000	NaN
E	1.000		1.000	0.000	NaN
F	1.000		1.000	0.000	NaN
F'	0.111	<	0.698	0.590	8.4E-15
G'	0.048	<	0.717	0.670	1.3E-20
G	0.992		1.000	0.008	1
H	0.984		1.000	0.016	0.89
I	1.000		1.000	0.000	NaN
J	1.000		1.000	0.000	NaN
J'	0.103	<	0.943	0.840	3.9E-26
K	1.000		1.000	0.000	NaN
K'	1.000		1.000	0.000	NaN
K''	0.302	<	0.849	0.550	6.1E-11
L	1.000		1.000	0.000	NaN
L'	0.349		0.208	-0.140	0.09
β 1-0	0.476		0.396	-0.080	0.41
β 1-1	1.000		0.981	-0.019	0.65
β 1-2	1.000		1.000	0.000	NaN
β 1-3	1.000		1.000	0.000	NaN
β 1-4	1.000		1.000	0.000	NaN
β 1-5	0.992	>	0.868	-0.120	0.0011
β 2-1	1.000		1.000	0.000	NaN
β 2-2	1.000		1.000	0.000	NaN
β 3-1	0.984		0.925	-0.060	0.12
β 3-2	0.992		0.981	-0.011	1
β 3-3	0.984		0.981	-0.003	1
β 4-1	0.929		0.887	-0.042	0.53
β 4-2	0.929		0.887	-0.042	0.53
β 5-1	0.278		0.189	-0.089	0.29
β 5-2	0.278		0.189	-0.089	0.29
β 6-1	0.611	>	0.075	-0.540	1.5E-10
β 6-2	0.611	>	0.075	-0.540	1.5E-10

5. MAIN PUBLICATIONS

Supplementary Table S3. Comparison of the SSE length distributions in the bacterial and eukaryotic CYP structures. Results of the Kolmogorov-Smirnov test comparing the SSE length distributions in Set-NR-Bact vs Set-NR-Euka. The column “Comparison” marks the significant differences at confidence level $\alpha = 0.05$ (>: longer in Bacteria, <: longer in Eukaryota). The column “ p -value” contains the p -values for two-sided alternative hypothesis. Columns “ p_g ” and “ p_l ” contain values for one-sided alternative hypotheses.

SSE label	Mean length (Set-NR-Bact)	Comparison	Mean length (Set-NR-Euka)	Mean difference	Median difference	p -value	p_g	p_l
A'	5.4	<	6.6	1.1	1.0	0.0049	0.77	0.0025
A	11.0	<	12.0	1.4	1.0	4.2E-11	0.41	2.1E-11
B	9.4	<	10.0	0.6	2.0	1.8E-09	0.79	9E-10
B'	6.7		7.8	1.1	0.0	0.05	0.96	0.025
B''	3.2		3.3	0.1	0.0	0.96	0.9	0.61
C	14.0	<	16.0	1.4	0.0	5.9E-05	1	3E-05
D	23.0	>	22.0	-0.3	-3.0	0.022	0.011	0.074
E	17.0		18.0	0.6	0.0	0.17	1	0.083
F	15.0	<	17.0	2.1	3.0	1.7E-07	0.75	8.6E-08
F'	5.4		6.9	1.5	1.0	0.078	0.9	0.039
G'	7.5		6.2	-1.3	-1.5	0.39	0.2	0.38
G	24.0	<	26.0	2.1	2.0	1.3E-06	0.92	6.5E-07
H	8.7	<	10.0	1.6	4.0	7.1E-06	1	3.6E-06
I	32.0		32.0	0.4	0.0	1	1	0.89
J	10.0	<	15.0	5.2	5.0	0	1	1.7E-28
J'	5.2	<	6.9	1.8	1.0	0.0014	1	0.00072
K	14.0	>	14.0	-0.3	-1.0	2.3E-14	1.2E-14	0.18
K'	6.1		6.1	0.0	0.0	1	0.96	0.95
K''	3.8	<	4.8	0.9	0.0	0.0013	1	0.00067
L	19.0	>	18.0	-0.6	-1.0	5.6E-14	2.8E-14	0.95
L'	3.1		3.3	0.2	0.0	1	1	0.86
β 1-0	1.6	>	1.3	-0.3	-1.0	0.017	0.0087	0.93
β 1-1	4.5	<	5.8	1.3	2.0	1.6E-15	1	7.8E-16
β 1-2	4.5	<	5.8	1.4	2.0	1.9E-13	1	9.7E-14
β 1-3	4.3		4.4	0.0	0.0	1	1	0.93
β 1-4	4.8	>	3.6	-1.3	-1.0	5.9E-06	2.9E-06	1
β 1-5	2.1	>	1.7	-0.4	0.0	0.0031	0.0016	1
β 2-1	2.9		2.9	0.0	0.0	0.89	0.72	0.51
β 2-2	2.9		2.9	0.1	0.0	0.89	0.94	0.51
β 3-1	2.5		2.6	0.1	0.0	0.0013	0.00066	0.0027
β 3-2	3.9	<	5.5	1.5	2.0	1.5E-13	0.95	7.4E-14
β 3-3	2.9	<	4.5	1.6	2.0	4.4E-16	0.78	2.5E-16
β 4-1	1.7		2.2	0.5	1.0	0.062	1	0.031
β 4-2	2.0		2.2	0.2	1.0	0.062	0.54	0.031
β 5-1	1.3		1.1	-0.2	0.0	0.87	0.49	1
β 5-2	1.3		1.1	-0.2	0.0	0.87	0.49	1
β 6-1	1.3		2.0	0.7	1.0	0.43	0.98	0.22
β 6-2	1.1	<	3.2	2.2	3.0	0.032	1	0.016

Uncovering of cytochrome P450 anatomy by SecStrAnnotator



Supplementary Figure S5. Phylogenetic tree based on multiple sequence alignment of full sequences from Set-NR. Eukaryotic sequences are highlighted in red, bacterial in light blue, anomalous bacterial group in dark blue, archaeal in green, viral in white.

References

1. Offmann, B., Tyagi, M. & de Brevern, A. G. Local Protein Structures. *Curr. Bioinforma.* **2**, 165–202 (2007).
2. Richardson, J. S., Getzoff, E. D. & Richardson, D. C. The beta bulge: a common small unit of nonrepetitive protein structure. *Proc. Natl. Acad. Sci. U. S. A.* **75**, 2574–2578 (1978).
3. Chan, A. W., Hutchinson, E. G., Harris, D. & Thornton, J. M. Identification, classification, and analysis of beta-bulges in proteins. *Protein Sci. Publ. Protein Soc.* **2**, 1574–1590 (1993).
4. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**, 2577–2637 (1983).

OverProt: secondary structure consensus for protein families

Adam Midlik^{1,2}, Ivana Hutařová Vařeková^{2,3,4}, Jan Hutař², Aliaksei Chareshneu^{1,2}, Karel Berka⁴, Radka Svobodová^{1,2}

¹ CEITEC – Central European Institute of Technology, Masaryk University, Kamenice 5, 625 00 Brno, Czech Republic

² National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Kamenice 5, 625 00 Brno, Czech Republic

³ Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic



⁴ Department of Physical Chemistry, Faculty of Science, Palacký University, 17. listopadu 1192/12, 771 46 Olomouc, Czech Republic

Bioinformatics, (in press), **2022**.

<https://doi.org/10.1093/bioinformatics/btac384>

Structural bioinformatics

OverProt: secondary structure consensus for protein families

Adam Midlik ^{1,2}, Ivana Hutařová Vařeková^{2,3,4}, Jan Hutař², Aliaksei Charehneu^{1,2}, Karel Berka ^{4,*} and Radka Svobodová ^{1,2,*}

¹CEITEC—Central European Institute of Technology, Masaryk University, 625 00 Brno, Czech Republic, ²National Centre for Biomolecular Research, Faculty of Science, Masaryk University, 625 00 Brno, Czech Republic, ³Department of Computer Systems and Communications, Faculty of Informatics, Masaryk University, 602 00 Brno, Czech Republic and ⁴Department of Physical Chemistry, Faculty of Science, Palacký University, 771 46 Olomouc, Czech Republic

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on January 28, 2022; revised on April 24, 2022; editorial decision on May 26, 2022; accepted on June 5, 2022

Abstract

Summary: Every protein family has a set of characteristic secondary structures. However, due to individual variations, a single structure is not enough to represent the whole family. OverProt can create a secondary structure consensus, showing the general fold of the family as well as its variation. Our server provides precomputed results for all CATH superfamilies and user-defined computations, visualized by an interactive viewer, which shows the secondary structure element type, length, frequency of occurrence, spatial variability and β -connectivity.

Availability and implementation: OverProt Server is freely available at <https://overprot.ncbr.muni.cz>.

Contact: karel.berka@upol.cz or radka.svobodova@ceitec.muni.cz

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

The protein structures deposited in the Protein Data Bank (Armstrong *et al.*, 2020) can be classified into protein families based on their similarity. The CATH database currently defines more than 6000 homologous superfamilies (shortly: families), and some of them contain thousands of protein structures. Each family collects homologous protein domains with conserved structure and function (or a set of functions), which is essential for organizing and better understanding the functional information we have for the individual proteins (Sillitoe *et al.*, 2021).

Every protein family has a set of characteristic secondary structure elements (SSEs), namely helices and β -strands. Their arrangement is well defined and relatively consistent throughout the whole family. Hence, they provide a guide for orientation within the structures and can also be used to describe the position of the key regions, e.g. active sites.

Nevertheless, there is usually some variation within each family—some SSEs are missing in some structures, and the exact length and 3D position of each SSE vary from structure to structure. Therefore, describing the secondary structure of a family as a whole (i.e. the secondary structure consensus) is not straightforward. However, having the SSE consensus can give us a helpful insight into the structural family, as it highlights the conserved structural features and variations in the family, just like a sequence logo gives

us an insight into a sequence family (Schneider and Stephens, 1990). The application of SSE consensus was previously shown by the FunFam approach of functional families (Scheibenreif *et al.*, 2019).

In this paper, we present our new software OverProt, which can construct and visualize the secondary structure consensus for a given protein family. This can greatly complement the current family visualization in 3D by structure superimposition [e.g. in CATH or PDBe-KB (PDBe-KB Consortium, 2022)], which is often messy and hinders secondary structure information. OverProt Server (<https://overprot.ncbr.muni.cz>) provides precomputed consensus for each CATH family and allows computation for user-defined families without the need for installation. The desktop version of OverProt, including the source codes and Docker images, is available at <https://gitlab.com/midlik/overprot>.

2 Materials and methods

OverProt consists of three main parts: OverProt Core, OverProt Viewer and OverProt Server.

OverProt Core is an algorithm that constructs the secondary structure consensus for a given set of protein structures (whole chains or their parts, with reasonable structural similarity). We refer to this set as a family and to individual structures as domains. The algorithm (implemented in Python) proceeds in several steps (see [Supplementary Data](#) for details):

5. MAIN PUBLICATIONS

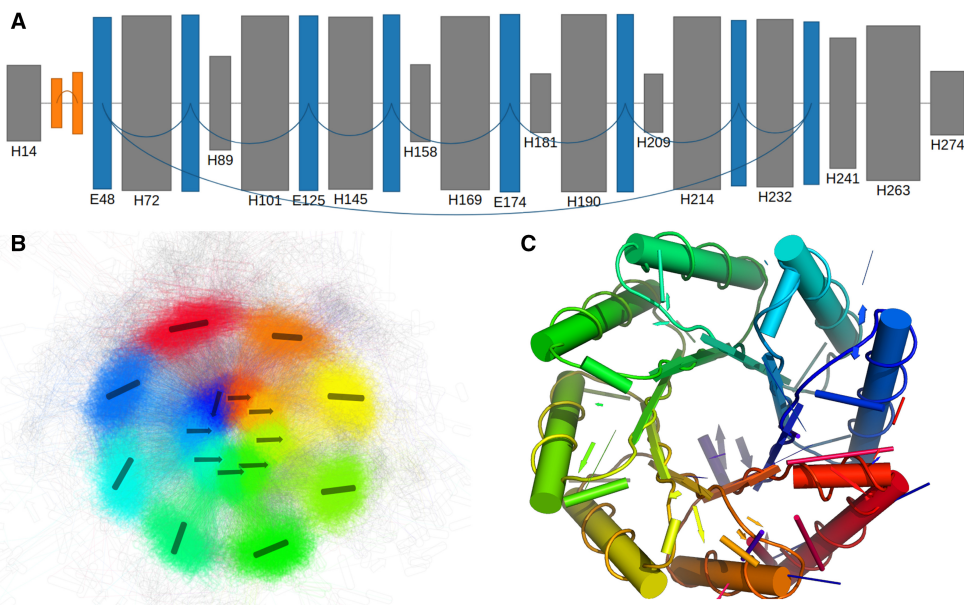


Fig. 1. Secondary structure consensus for family 3.20.20.70 (Aldolase class I). (A) OverProt Viewer reveals a β -barrel (blue dark) with eight strands connected by parallel β -connections (lower arcs), alternating with eight major helices (gray). There are also seven shorter helices and an antiparallel β -sheet with two strands (orange bright) with lower occurrence. The spatial arrangement is shown by (C) 3D view and simplified by (B) 2DProts (A color version of this figure appears in the online version of this article.)

- Preparation: Download input data, select a domain subset (optional). In the desktop version, the user can also provide structure files in mmCIF format (for experimental or modeled structures that are not in the PDB database).
- Structural alignment: First, align a subset of domains (max. 100) by MAPSCI (Ilinkin *et al.*, 2010) to produce a consensus structure. Then, realign each domain to the MAPSCI consensus by the cealign algorithm in PyMOL (Shindyalov and Bourne, 1998).
- Secondary structure assignment: Detect the SSEs in each domain by SecStrAnnotator (Midlik *et al.*, 2019, 2021).
- Guide tree: Cluster the domains by agglomerative clustering to produce a guide tree.
- Merging: Populate the guide tree leaves by the SSE sets of the respective domains. In each internal node, combine the child SSE sets by merging mutually equivalent SSEs. The root then contains the consensus SSE set of the family.
- Visualization: Process the consensus into an SVG diagram, diagram.json file, PyMOL session and PyMOL-generated image. OverProt Viewer is a web component for interactive visualization of the SSE consensus. Its input is the preprocessed diagram.json file. It is implemented in TypeScript with D3.js.
- OverProt Server provides precomputed SSE consensus (database) and runs the OverProt Core algorithm for user-defined sets of domains (jobs). The database is constructed as follows:

- Retrieve the current list of families and their members from CATH and PDBe API. This currently means 6631 families and over 470 000 domains.
- Remove duplicates (multiple domains from the same PDB entry).
- Apply the OverProt Core algorithm to each family.

The database updates are synchronized with CATH updates. OverProt Server is implemented using Python (Flask), Gunicorn, Redis Queue, Nginx and Docker.

3 Results

The database part of the server contains precomputed SSE consensus for all CATH families. The user gets to the family of interest by its CATH ID (e.g. 1.10.630.10) or by a particular PDB entry or domain. The family view contains the interactive OverProt Viewer, 2D diagram [2DProts (Hutařová Vařeková *et al.*, 2021)] and 3D image of the consensus (Fig. 1). More detailed results are available in a ZIP file.

OverProt Viewer shows each consensus SSE as a rectangle or oval, whose height corresponds to its occurrence (percentage of domains in which it is present), width corresponds to its average length (number of residues), and the oval shape shows the variability of its length. Strands from the same β -sheet are shown in the same color; helices are shown in gray. Connections of the strands in a β -sheet are shown by arcs (lower arcs—parallel, upper arcs—antiparallel).

The user can navigate to a particular protein domain—the integrated view relates the consensus SSEs of the family in 1D (OverProt Viewer) to the particular domain in 2D (interactive 2DProts diagram) and 3D [Mol* Viewer (Sehnal *et al.*, 2021)]. When the user hovers over an entity in one view, it gets highlighted in all three views. This page also provides the SSE annotation file for the domain. All data are also available via API.

In user-defined queries, the user can submit a list of up to 500 domains. The job computation typically takes a few minutes. The results can later be accessed by the assigned job URL; no registration is needed.

4 Conclusion

OverProt provides secondary structure consensus for CATH families and user-defined sets of structurally similar structures. The consensus shows the similarities and differences within the family in terms of SSE type, length, frequency of occurrence, spatial variability and β -connectivity. This greatly complements the currently available methods of protein family visualization.

OverProt is already employed in the visualization tool 2DProts (Hutařová Vařeková *et al.*, 2021), and its integration into CATH and PDBe-KB is planned.

Acknowledgement

We thank Jaroslav Koča (in memoriam) for valuable feedback and encouragement.

Funding

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the projects ELIXIR CZ [LM2018131] and e-INFRA CZ [LM2018140].

Conflict of Interest: none declared.

Data availability

The data underlying this article were accessed from the PDB database (<https://www.ebi.ac.uk/pdbe>) and the CATH database (<https://www.cathdb.info>).

The generated data are available at <https://overprot.ncbr.muni.cz>. The software is available at <https://gitlab.com/midlik/overprot>.

References

- Armstrong, D.R. *et al.* (2020) PDBe: improved findability of macromolecular structure data in the PDB. *Nucleic Acids Res.*, **48**, D335–D343.
- Hutařová Vařeková, J. *et al.* (2021) 2DProts: database of family-wide protein secondary structure diagrams. *Bioinformatics*, **37**, 4599–4601.
- Linkin, I. *et al.* (2010) Multiple structure alignment and consensus identification for proteins. *BMC Bioinformatics*, **11**, 71.
- Midlik, A. *et al.* (2019) Automated family-wide annotation of secondary structure elements. *Methods Mol. Biol.*, **1958**, 47–71.
- Midlik, A. *et al.* (2021) Uncovering of cytochrome P450 anatomy by SecStrAnnotator. *Sci. Rep.*, **11**, 12345.
- PDBe-KB Consortium (2022) PDBe-KB: collaboratively defining the biological context of structural data. *Nucleic Acids Res.*, **50**, D534–D542.
- Scheibenreif, L. *et al.* (2019) FunFam protein families improve residue level molecular function prediction. *BMC Bioinformatics*, **20**, 400.
- Schneider, T.D. and Stephens, R.M. (1990) Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res.*, **18**, 6097–6100.
- Sehnal, D. *et al.* (2021) Mol* viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic Acids Res.*, **49**, W431–W437.
- Shindyalov, I.N. and Bourne, P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*, **11**, 739–747.
- Sillitoe, J. *et al.* (2021) CATH: increased structural coverage of functional space. *Nucleic Acids Res.*, **49**, D266–D273.

Supplementary information for “OverProt: secondary structure consensus for protein families”

Adam Midlik^{1,2}, Ivana Hutařová Vařeková^{2,3,4}, Jan Hutař^{1,2}, Aliaksei
Charesheu^{1,2}, Karel Berka^{4,*}, and Radka Svobodová^{1,2,*}

¹CEITEC - Central European Institute of Technology, Masaryk University, 625 00 Brno, Czech Republic

²National Centre for Biomolecular Research, Faculty of Science, Masaryk University, 625 00 Brno, Czech Republic

³Faculty of Informatics, Masaryk University, 602 00 Brno, Czech Republic

⁴Department of Physical Chemistry, Faculty of Science, Palacký University, 771 46 Olomouc, Czech Republic

*To whom correspondence should be addressed.

Table of contents

1 Introduction	2
2 Terminology	2
3 Methods - OverProt Core	3
3.1 Preparation	3
3.2 Structural alignment	3
3.3 Secondary structure assignment	4
3.4 Guide tree	4
3.5 Merging	5
3.6 Annotation	7
3.7 Visualization	7
3.8 Execution	7
4 Interactive visualization by OverProt Viewer	8
5 Data computation for OverProt Server	9
6 Appendix	10
6.1 Distance function for two weighted structures	10
6.2 Merging two weighted structures	12
6.3 Matching two SSE directed acyclic graphs (DAGs)	12
7 References	14

1 Introduction

OverProt is a tool for constructing and visualizing the secondary structure consensus for protein families. The consensus produced by OverProt can be used as a template for annotation of secondary structure elements in protein families, e.g. by SecStrAnnotator.

OverProt consists of three main parts: the main algorithm **OverProt Core** constructs the secondary structure consensus, **OverProt Viewer** visualizes the consensus, and **OverProt Server** presents the results on the web and allows user-defined computations.

The source code is freely available at <https://gitlab.com/midlik/overprot>.

2 Terminology

- **Protein structure** - a set of atoms with assigned 3D coordinates. A structure consists of one or more **chains**. A chain is a sequence of **residues**, each of which consists of the individual **atoms**. OverProt works with structures in **mmCIF format**. Structures deposited in the PDB (Armstrong *et al.*, 2020) are referenced by their PDB ID (e.g. **1tqn**). OverProt follows the *label** numbering scheme when referencing chains and residues within a structure (i.e. items `label_asym_id` and `label_seq_id` in the mmCIF file) - this is in some cases different from the *auth** numbering scheme.
- **Protein domain** - a part of protein structure, either a whole chain or a range (ranges) of residues in a chain. A domain is defined by the structure identifier (PDB ID), chain identifier, and one or more ranges of residues, e.g. **1tqn,A,7:478** or **1n26,A,2:9,94:192**. Residue ranges include the start and end residue (e.g. **5:8** means residues 5, 6, 7, 8).
- **Protein family** - a set of protein domains with a reasonable structural similarity. The set can be provided by the user or it can be defined based on the CATH database (Sillitoe *et al.*, 2021), in which case the family (*CATH superfamily*) is identified by its CATH ID (e.g. **1.10.630.10**) and domains are identified by CATH domain ID (e.g. **1tqnA00**).
- **Secondary structure element (SSE)** - a section of a protein chain with some secondary structure pattern. OverProt focuses on two key types of SSEs - **helices** (H) and **β -strands** (E). Each SSE within a protein structure can be identified by its chain identifier, start (index of its first residue), end (index of its last residue), and type (H/E). For comparing SSEs, it is convenient to simplify an SSE to a line segment (i.e. 3D coordinates of the start and end point).

The term β -connectivity refers to the way in which the strands are connected: a **β -ladder** is a connection of two strands (realized by hydrogen bonds) and can be either parallel or antiparallel; a **β -sheet** is a set of strands which are connected by β -ladders (a connected component).

This model is kept as simple as possible (different helix types (α , 3_{10} , π) are not distinguished; other SSE type (loops, turns) are not taken into account). Secondary structure assignment (detection of SSEs) is performed by **SecStrAnnotator**, more details can be found in its original paper (Midlik *et al.*, 2019).

We will sometimes use the term **base SSEs** to distinguish SSEs from consensus SSEs.

- **Consensus SSE** – a set of equivalent SSEs from different family members.
- **Secondary structure consensus** – a set of consensus SSEs with a given order and β -connectivity.

3 Methods - OverProt Core

OverProt Core is an algorithm that constructs the secondary structure consensus for a given protein family. The algorithm proceeds in several steps. (In the following text, `--xx` refers to a command-line option of `overprot.py`, `[xx]yy` refers to a setting `yy` in section `xx` in the configuration file (`overprot-config.ini`).

3.1 Preparation

- The list of domains for the family is downloaded from PDBE API https://www.ebi.ac.uk/pdbe/api/mappings/{family_id} (if not already given by `--domains`).
- Select sample: If `--sample_size` is smaller than the number of domains, a random subset of the domain list is selected.
The family may contain multiple domains from the same PDB entry. If `[sample_selection]unique_pdb` is `True`, then these are treated as duplicates and only one of them is selected (the first in alphabetical order).
- Download structures: The structures of listed domains are downloaded in mmCIF format; the domains are cut out from the structures and saved in separate files. The sources of structures are given by `--structure_source` and `[download]structure_sources`. The structures are also converted to the PDB format for later steps (namely, for alignment by program MAPSCI). The download step is performed by an auxiliary program `StructureCutter` written in C# (a part of the OverProt project).

3.2 Structural alignment

Multiple structure alignment is performed in 2 steps:

- Program MAPSCI (Ilinkin *et al.*, 2020) is used to calculate a consensus structure (`mapsci/consensus.cif`). For performance reasons, at most 100 domains are selected for this calculation (in a quasi-random way, i.e. for the same family it selects the same subset every time).

To reduce indeterminism and ease later visualization, the consensus structure is centered to the origin (0, 0, 0), rotated so that its PCA (principal component analysis) components are aligned to the XYZ axes (“the structure is laid flat”), and flipped in a consistent way (roughly so that the start and the end of the chain are more in front, and the chain goes from left-top to right-bottom).

In general, MAPSCI produces a reasonable consensus structure, but the alignment of the individual domains is often poor, so the following re-alignment step is necessary.

- In the re-alignment step, all domains are structurally aligned onto the MAPSCI consensus structure via `cealign` algorithm (Shindyalov and Bourne, 1998) provided in PyMOL module (Schrödinger, LLC.) version 2.3.0. In rare cases `cealign` fails (when the domains are too short) - in such cases a simple internal algorithm is used instead (theoretically inefficient and not allowing gaps, but sufficient for these very short domains).

3.3 Secondary structure assignment

The SSEs in each domain are detected by `SecStrAnnotator` (Midlik *et al.*, 2019, 2021) with options `--onlyssa --verbose --batch`.

3.4 Guide tree

The domains are clustered by agglomerative clustering to produce a **guide tree**. The algorithm starts with a set of structures. It finds the two most similar structures and merges them into a new structure. This is then repeated until we end up with a single structure corresponding to the tree root.

This agglomerative algorithm can be expressed by the following pseudocode:

```

Workset = { the structures of all input domains }
while |Workset| > 1:
    A, B = two nearest structures in Workset
    C = merge_structures(A, B)
    Children of C = {A, B}
    Workset = Workset - {A, B} ∪ {C}
    
```

At the end, `Workset` will only contain one structure, which is the tree root. The topology of the tree will be defined by `Children`. An example is shown in Figure 1.

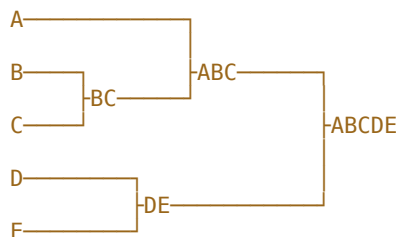


Figure 1: An example of the guide tree construction. 5 structures were initially in `Workset`. B+C were merged into BC, then D+E into DE, then A+BC into ABC, and finally ABC+DE into ABCDE.

The details of the algorithm are described in Appendix 6.1 (distance function, which determines the nearest structures) and 6.2 (operation `merge_structures`).

3.5 Merging

This step is the core of the consensus generation algorithm. As an input, we have a set of k protein domains. Each domain is simplified to a sequence of SSEs (defined by their type, line segment, etc.). The required output is a clustering of all input SSEs (see Figure 2).

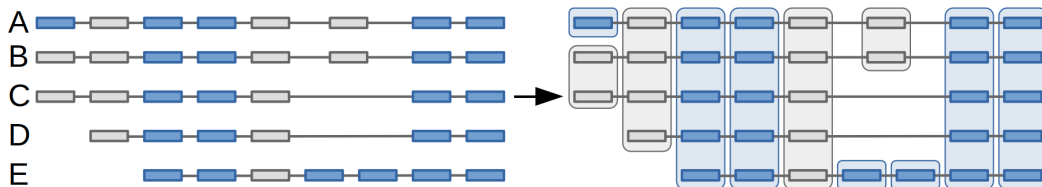


Figure 2: An example of 5 domains simplified to a sequence of base SSEs (gray = helix, blue = strand) and their clustering into 11 clusters.

However, the clustering must fulfil these constraints:

1. Each cluster can contain only elements of the same type (only helices or only strands).
2. A cluster must not contain more than one element from the same protein domain.
3. There must be a partial order of the clusters. This constraint can be formalized as:
 - Base SSE x precedes base SSE y (written $x \rightarrow y$) if they are from the same protein domain and x goes before y in the sequence.
 - Cluster P directly precedes cluster Q ($P \Rightarrow Q$) if there exist SSEs $x \in P, y \in Q$ such that $x \rightarrow y$.
 - Cluster P precedes cluster Q ($P \rightarrow Q$) if there exists a sequence of clusters $P \Rightarrow R_1 \Rightarrow \dots \Rightarrow R_n = Q$ where $n \geq 1$ (in other words, \rightarrow is the transitive closure of \Rightarrow).
 - There must be no cluster P , such that $P \rightarrow P$.

Note: The order of some clusters may be undefined (i.e. neither $P \rightarrow Q$ nor $Q \rightarrow P$) if they contain no SSEs from the same domain. Therefore \rightarrow is a partial order on the clusters (not a total order). We represent the order by a directed acyclic graph (DAG) (see Figure 3).



Figure 3: An example of a DAG representing clusters of SSEs from Figure 2. The height of each rectangle shows the weight of the cluster (the number of base SSEs in the cluster). The color shows the cluster type (gray = helix, blue = strand). The direction of the edges is implicit (left to right). The edges that can be inferred from transitivity are not shown (i.e. we show only the transitive reduction (Hasse diagram)).

The merging step follows the guide tree. First, each guide tree leaf is populated with the DAG of SSEs of the respective domain. In each internal node, the DAGs from the two children nodes are matched together and merged. The root then contains the consensus SSEs of the whole family (see Figure 4).

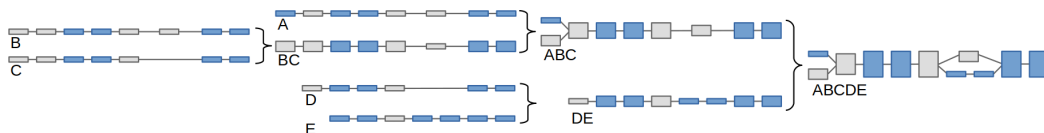


Figure 4: The process of merging 5 DAGs based on the guide tree from Figure 1.

The matching and merging of two SSE DAGs is in principle similar to matching and merging of two weighted structures. The best matching is also found by dynamic programming. However, it is more complicated here because 1) SSEs of different type cannot be matched (this can cause the branching in the resulting DAG), and 2) the dynamic programming algorithm is not as straightforward for matching DAGs as it is for matching sequences. More details are provided in Appendix 6.3.

The β -connectivity is not directly considered in the merging algorithm (though it is included in the distance function for DAG matching). Therefore it is necessary to determine the β -connectivity of the resulting clusters based on the β -connectivity of the base SSEs.

A β -ladder PQo (connecting strand clusters P and Q with orientation o (parallel/antiparallel)) is included in the resulting consensus if

$$\frac{n_{PQo}}{\min\{n_P, n_Q\}} \geq 0.5$$

where n_P is the number of strands in cluster P , n_Q is the number of strands in cluster Q , and n_{PQo} is the number of base ladders connecting a base strand in P to a base strand in Q with orientation o (see Figure 5).

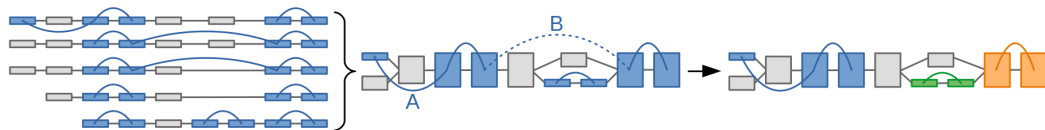


Figure 5: Merging β -ladders from 5 domains. Lower arcs show parallel, upper arcs antiparallel ladders. Ladder A is included because $n_{PQo}/\min\{n_P, n_Q\} = 1/\min\{1, 5\} = 1 \geq 0.5$. Ladder B is not included because $n_{PQo}/\min\{n_P, n_Q\} = 2/\min\{5, 5\} = 0.4 < 0.5$. The rightmost column shows the separation of the consensus strands into sheets (connected components).

After the clustering, a variety of statistics are computed for each consensus SSE and saved in `results/consensus.sses.json`:

- Occurrence - the number of domains that contain this SSE, divided by the total number of domains in the family. In the previous example, the first strand occurs in 1 out of 5 domains; thus its occurrence is 0.2 or 20%.
- Average length - measured as the number of residues.
- Average line segment - the average start and end point in 3D.
- 3D variability - the variance of the start and end point.

Each consensus SSE also gets a unique label (containing its type and sequential number, e.g. E0, H1, H2...) and color.

3.6 Annotation

In this optional step, the generated SSE consensus is used as an annotation template for SecStrAnnotator, and all family members are annotated. Before the annotation, the SSEs with low occurrence (< 5%) are removed, which dramatically reduces the running time of SecStrAnnotator. SecStrAnnotator is run with these options: `--ssa file --align none --metrictype 3 --fallback 30 --unannotated`. Metric type 3 must be used because the default metric requires residue numbers for each SSE, but these are not available for the consensus SSEs. Option `--unannotated` includes also the unannotated SSEs in the resulting annotation files, with labels prefixed by underscore (e.g. `_H0`).

3.7 Visualization

The generated SSE consensus is visualized by several SVG diagrams with different settings and `diagram.json` file is produced, which will be used for interactive visualization by OverProt Viewer. A PyMOL session (`.pse`) is created, with the MAPSCI consensus structure shown as ribbon and the consensus SSEs shown as cylinders and arrows. The width of each cylinder/arrow shows the occurrence of the corresponding helix/strand. A PNG image is also rendered from the session. A session with all domains and their SSEs is generated if `[visualization]create_multi_session` is `True` (very slow, not recommended for larger families).

3.8 Execution

OverProt Core is implemented mostly in Python3 and designed to run in the Linux environment (tested on Ubuntu 20.04). On the other operating systems, it can be run in Docker. Before the first execution, the dependencies must be installed:

```
sh install.sh --clean
```

All steps of the algorithm are combined in `overprot.py`. It is run in a Python virtual environment. Its arguments are the CATH family ID and the output directory:

```
. venv/bin/activate
python overprot.py --help
python overprot.py 1.10.630.10 data/cyp/
```

Multiple families can be processed in parallel using `overprot_multifamily.py`. Its arguments are the family list and the output directory:

```
. venv/bin/activate
python overprot_multifamily.py --help
python overprot_multifamily.py data/families.txt data/multifamily/
```

More details can be found in the `README.md` files in the project repository.

4 Interactive visualization by OverProt Viewer

OverProt Viewer is a web component for interactive visualization of the SSE consensus. Its input is the preprocessed `diagram.json` file. It is implemented in TypeScript with D3.js.

OverProt Viewer shows each consensus SSE as a rectangle or an oval, whose height corresponds to its occurrence and width corresponds to its average length (number of residues). Strands from the same β -sheet are shown in the same color; helices are shown in gray. Connections of the strands in a β -sheet are shown by arcs (lower arcs - parallel, upper arcs - antiparallel). Hovering over an SSE shape shows the SSE details (see Figure 6).

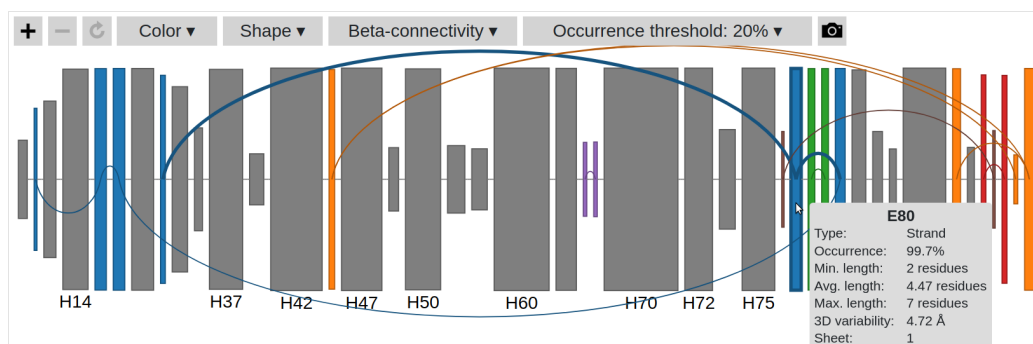


Figure 6: OverProt Viewer showing the secondary structure consensus for CATH family 1.10.630.10 (Cytochrome P450).

Visualization options include:

- Color:
 - Uniform - Show all SSEs in the same color.
 - Type - Show β -strands in blue, helices in gray.
 - Sheet - Assign the same color to all β -strands from the same β -sheet; show helices in gray.
 - Variability - The 3D variability measures the standard deviation of the SSE start and end point coordinates. Low values (dark) indicate conserved SSE position, high values (bright) indicate variable SSE position.
 - Rainbow - Standard rainbow coloring from N-terminus (blue) to C-terminus (red).
- Shape:
 - Rectangle - Show the SSEs as rectangles. The height of the rectangle indicates its occurrence; the width indicates its average length (number of residues).
 - SymCDF - The cumulative distribution function (CDF) describes the statistical distribution of the SSE length. The SymCDF shape consists of four symmetrical copies of the CDF; the bottom right quarter is the classical CDF. The widest part of the shape corresponds to the maximum length, the narrowest to the minimum length, the height corresponds to the occurrence.

- Beta-connectivity:
 - On - The beta-connectivity shows how β -strands are connected to each other in β -sheets. The lower arcs indicate parallel ladders; the upper arcs indicate antiparallel ladders.
 - Off - The beta-connectivity arcs are hidden.
- Occurrence threshold:
 - Hides the SSEs with occurrence lower than the specified threshold. Can be set to any number from 0% to 100%.

OverProt Viewer can be set to dispatch and listen to HTML events. When an SSE is hovered over or clicked, the viewer dispatches an event (`PDB.overprot.hover` or `PDB.overprot.select`). The information about the selected elements is included in `event.detail`. Conversely, the viewer handles the incoming events (`PDB.overprot.do.hover` or `PDB.overprot.do.select`) by highlighting the selected elements. This allows interactivity across several web components, as demonstrated by the integrated view in the OverProt web (https://overprot.ncbr.muni.cz/domain_view?family_id=1.10.630.10&domain_id=1jfbA00) - hovering over an SSE in any of the three components (OverProt Viewer, interactive 2DProts (Hutařová *et al.*, 2021), MolStar Viewer (Sehnal *et al.*, 2021)) highlights it in all three.

5 Data computation for OverProt Server

OverProt Server provides precomputed SSE consensus (database) and runs the OverProt Core algorithm for user-defined sets of domains (jobs). OverProt Server is implemented using Python (Flask), Gunicorn, Redis Queue, Nginx, and Docker. A running instance is available at <https://overprot.ncbr.muni.cz>.

The database is constructed in this way:

- Retrieve the current list of families from CATH (<http://download.cathdb.info/cath/releases/latest-release/cath-classification-data/cath-superfamily-list.txt>). The list currently contains 6631 families, out of which 64 are empty families (January 2022).
- Retrieve the domain lists for each family, including chains and residue ranges, from PDBE API (https://www.ebi.ac.uk/pdbe/api/mappings/{family_id}). This is currently over 470k domains in total (January 2022).
- Remove duplicates (i.e. multiple domains from the same PDB entry). The number of domains without duplicates is currently over 200k (January 2022).
- Apply the OverProt Core algorithm to each family.

The whole process is realized by:

```
. venv/bin/activate
python overprot_multifamily.py --download_family_list_by_size \
```

```
--config working_scripts/overprot-config-overprotserverdb.ini \
--collect - $UPDATE_DIRECTORY
```

6 Appendix

6.1 Distance function for two weighted structures

To be able to compare the real structures of the input domains as well as the artificial structures created by merging, we use the concept of a weighted structure. A weighted structure is a sequence of points (C-alpha coordinates) where each point has its relative weight. In any real structure, all relative weights are equal to 1, but merging can create points with smaller relative weights. The absolute weight of a weighted structure is simply the number of the real structures that have been merged to form this weighted structure.

Formally, a weighted structure A is a tuple $(n^A, \mathbf{R}^A, \mathbf{W}^A, k^A)$ where n^A is the length of the weighted structure (number of points), \mathbf{R}^A is the matrix of their coordinates ($n^A \times 3$), \mathbf{W}^A is the vector of their relative weights $\in (0, 1]$, and k^A is the absolute weight of A . Example of a weighted structure:

$$n^A = 4 \quad \mathbf{R}^A = \begin{bmatrix} -1.1 & -2.9 & 0.1 & 0.4 \\ 0.0 & 1.1 & 0.9 & -2.7 \\ 5.2 & 2.1 & 0.0 & 0.8 \end{bmatrix} \quad \mathbf{W}^A = [1 \quad 0.5 \quad 0.8 \quad 1] \quad k^A = 10$$

\mathbf{r}_i^A and w_i^A will refer to i -th column of \mathbf{R}^A and \mathbf{W}^A .

A protein domain can be converted into a weighted structure as follows: n is the number of residues, \mathbf{r}_i^A are the coordinates of the C-alpha atom of i -th residue, w_i^A is 1, and k^A is 1.

The distance function d is defined for two weighted points:

$$d((\mathbf{r}_i^A, w_i^A), (\mathbf{r}_j^B, w_j^B)) = \left(1 - e^{-\|\mathbf{r}_i^A - \mathbf{r}_j^B\|/R_0}\right) \cdot \min\{w_i^A, w_j^B\} + \frac{1}{2}|w_i^A - w_j^B|$$

The parameter R_0 was set to 10 Å.

In case that one of the weighted points is undefined (\perp), d is still defined:

$$d((\mathbf{r}_i^A, w_i^A), \perp) = \frac{1}{2}w_i^A \quad d(\perp, (\mathbf{r}_j^B, w_j^B)) = \frac{1}{2}w_j^B$$

(Notes: Distance d is not the Euclidean distance of the two points. $d \in [0, 1)$.)

A matching (or alignment) of two weighted structures A, B is a sequence of pairs $[(p_1, q_1), (p_2, q_2), \dots, (p_n, q_n)]$, where p_i and q_i are indices of the points of A and B . Indices must be increasing and must include each index exactly once for both A and B . Value \perp means that a particular point was not matched. Example of a valid matching for $n^A = 4, n^B = 5$:

$$[(1, 1), (2, \perp), (3, 2), (4, 3), (\perp, 4), (\perp, 5)]$$

The distance function D for two weighted structures A and B with a given matching M is defined:

$$D(A, B, M) = \sum_{(p,q) \in M} d((\mathbf{r}_p^A, w_p^A), (\mathbf{r}_q^B, w_q^B))$$

The distance function D^* of two weighted structures A and B is then:

$$D^*(A, B) = D(A, B, M^*)$$

where M^* is the best matching of A and B , i.e. the matching which minimizes $D(A, B, M^*)$.

The best matching can be found by dynamic programming. For this, the distance function d is converted into the score function s :

$$\begin{aligned} s((\mathbf{r}_i^A, w_i^A), (\mathbf{r}_j^B, w_j^B)) &= \frac{1}{2}w_i^A + \frac{1}{2}w_j^B - d((\mathbf{r}_i^A, w_i^A), (\mathbf{r}_j^B, w_j^B)) \\ s((\mathbf{r}_i^A, w_i^A), \perp) &= 0 \quad s(\perp, (\mathbf{r}_j^B, w_j^B)) = 0 \end{aligned}$$

Similarly, D is converted into the total score function S :

$$S(A, B, M) = \sum_{(p,q) \in M} s((\mathbf{r}_p^A, w_p^A), (\mathbf{r}_q^B, w_q^B)) = \frac{1}{2} \sum_{i=1}^{n^A} w_i^A + \frac{1}{2} \sum_{j=1}^{n^B} w_j^B - D(A, B, M)$$

From this equation, it can be seen that maximizing S by dynamic programming also minimizes D . (This dynamic programming algorithm is in principle very similar to the well-known Needleman-Wunsch algorithm for aligning sequences (Needleman and Wunsch, 1970), but it differs in the score function it uses.)

Notes: The distance function D^* is inspired by the edit distance for comparing two strings. It basically measures how much we have to edit A (move/insert/delete points) to transform it into B . Thanks to this design, D^* is a metric (i.e. $D^*(A, A) = 0$, $D^*(A, B) = D^*(B, A)$, and $D^*(A, B) + D^*(B, C) \geq D^*(A, C)$ for any weighted structures A, B, C).

When finding the two nearest items in the workset, it is not necessary to calculate the distance D^* for every pair of items - there are specialized data structures that can significantly decrease the number of distance calculations. We use a non-standard structure NN-tree (nearest neighbor tree). In some larger protein families, this can reduce the number of distance computations to less than 20%. (Standard structures like GH-tree, M-tree, etc. either miss some of the necessary operations (insert, delete) or perform worse than NN-tree for this particular application.) This is only possible because D^* is a metric.

6.2 Merging two weighted structures

Having two weighted structures A, B and their best matching $M^* = [(p_1, q_1), \dots, (p_n, q_n)]$, we can define operation *merge_structures* as follows:

$$\text{merge_structures}(A, B) = C = (n^C, \mathbf{R}^C, \mathbf{W}^C, k^C)$$

$$\begin{aligned} n^C &= n \\ \mathbf{r}_i^C &= \frac{\mathbf{r}_{p_i}^A w_{p_i}^A k^A + \mathbf{r}_{q_i}^B w_{q_i}^B k^B}{w_{p_i}^A k^A + w_{q_i}^B k^B} \\ w_i^C &= w_{p_i}^A k^A + w_{q_i}^B k^B \\ k^C &= k^A + k^B \end{aligned}$$

(If $p_i = \perp$, the values can be calculated by setting $w_{p_i}^A = 0$, thus simplifying to $\mathbf{r}_i^C = \mathbf{r}_{q_i}^B$, $w_i^C = w_{q_i}^B$. Similarly for $q_i = \perp$.)

6.3 Matching two SSE directed acyclic graphs (DAGs)

The distance function d for two SSEs P and Q is defined as the sum of Euclidean distances between their start points and between their end points:

$$d(P, Q) = \|\mathbf{u}_P - \mathbf{u}_Q\| + \|\mathbf{v}_P - \mathbf{v}_Q\|$$

where $\mathbf{u}_P, \mathbf{v}_P$ is the start and end point of SSE P , $\mathbf{u}_Q, \mathbf{v}_Q$ is the start and end point of SSE Q .

The score function s is then defined:

$$s(P, Q) = \begin{cases} SR(d(P, Q)) & \text{if } P, Q \text{ are of the same type (helix/strand)} \\ 0 & \text{otherwise} \end{cases}$$

where SR is the ‘‘smoothed ramp’’ function, which is basically a smooth, strictly decreasing version of the function $y = \max\{0, 1 - x/d_0\}$.

SR is defined by the implicit equation $d_0(1 - \alpha)y^2 + (x + d_0(2\alpha - 1))y - d_0\alpha = 0$. When solving this quadratic equation, the greater root is selected. The parameters were set to $d_0 = 30 \text{ \AA}$ and $\alpha = 0.01$.

The distance function d and the score function s can be easily extended from base SSEs to consensus SSEs. For a consensus SSE P , the point \mathbf{u}_P is simply the arithmetic mean of \mathbf{u} of all base SSEs included in P . Similarly for \mathbf{v}_P .

However, it will be useful to define the weight of a consensus SSE P (w_P) as the number of base SSEs included in P . Similarly, we will define the weights of the consensus β -ladders: w_{PQp} is the number of parallel ladders connecting a base strand in P to a base strand in Q , w_{PQa} is the number of antiparallel ladders connecting a base strand in P to a base strand in Q . (Base SSEs/ladders can be understood as consensus SSEs/ladders with weight 1.)

In order to reflect the β -connectivity in the score function, the “ladder correction” is applied to the strands:

$$s_{\text{corr}}(P_i, Q_j) = \frac{1}{2} \left(s(P_i, Q_j) + \sum_k \sum_l (\alpha_{ijkl} + \beta_{ijkl}) s(P_k, Q_l) \right)$$

where P_i, P_k are strands in the first matched DAG, Q_j, Q_l are strands in the second matched DAG, and the coefficients $\alpha_{ijkl}, \beta_{ijkl}$ maximize the value of $s_{\text{corr}}(P_i, Q_j)$ while fulfilling the following constraints:

$$\begin{aligned} \alpha_{ijkl} &\geq 0 & \beta_{ijkl} &\geq 0 \\ \sum_k \sum_l (\alpha_{ijkl} + \beta_{ijkl}) &\leq 1 \\ \sum_l \alpha_{ijkl} &\leq \frac{w_{P_i P_k a}}{w_{P_i}} & \sum_l \beta_{ijkl} &\leq \frac{w_{P_i P_k p}}{w_{P_i}} \\ \sum_k \alpha_{ijkl} &\leq \frac{w_{Q_j Q_l a}}{w_{Q_j}} & \sum_k \beta_{ijkl} &\leq \frac{w_{Q_j Q_l p}}{w_{Q_j}} \end{aligned}$$

For each pair P_i, Q_j , the values of coefficients $\alpha_{ijkl}, \beta_{ijkl}$ are determined by a greedy algorithm (i.e. first assigning the greatest possible value to the coefficients corresponding to the highest $s(P_k, Q_l)$, then the second highest, etc.).

For helices, no “ladder correction” is necessary, so $s_{\text{corr}}(P_i, Q_j) = s(P_i, Q_j)$.

A matching of two SSE DAGs \mathbf{G}, \mathbf{H} is a set of pairs $M = \{(P_1, Q_1), (P_2, Q_2), \dots, (P_n, Q_n)\}$, where $P_i \in V(\mathbf{G}), Q_j \in V(\mathbf{H})$, fulfilling these conditions:

- Each vertex is matched at most once: $\forall i, j : P_i \neq P_j \Leftrightarrow Q_i \neq Q_j$
- Only vertices of the same type are matched: $\forall i : \text{type}(P_i) = \text{type}(Q_i)$
- No cycle is created: $\nexists i, j : P_i \rightarrow P_j \wedge Q_j \rightarrow Q_i$

The best matching M^* of DAGs \mathbf{G}, \mathbf{H} is the matching which maximizes the total score S :

$$S(\mathbf{G}, \mathbf{H}, M^*) = \sum_{(P, Q) \in M^*} w_P w_Q s_{\text{corr}}(P, Q)$$

The corresponding best score is S^* :

$$S^*(\mathbf{G}, \mathbf{H}) = S(\mathbf{G}, \mathbf{H}, M^*)$$

The problem of finding the best matching and the best score for two DAGs \mathbf{G}, \mathbf{H} can be decomposed to smaller problems:

$$\begin{aligned} S^*(\mathbf{G}, \mathbf{H}) = \max \left(\right. & \{ S^*(\mathbf{G} - P, \mathbf{H}) \mid P \in \text{sinks}(\mathbf{G}) \} \\ & \cup \{ S^*(\mathbf{G}, \mathbf{H} - Q) \mid Q \in \text{sinks}(\mathbf{H}) \} \\ & \left. \cup \{ S^*(\mathbf{G} - P, \mathbf{H} - Q) + w_P w_Q s_{\text{corr}}(P, Q) \mid P \in \text{sinks}(\mathbf{G}), Q \in \text{sinks}(\mathbf{H}) \} \right) \end{aligned}$$

The trivial subproblems can be solved directly without decomposition:

$$S^*(G, K_0) = S^*(K_0, H) = 0 \quad M^*(G, K_0) = M^*(K_0, H) = \{\}$$

where K_0 is a graph with no vertices.

OverProt finds the best matching by a dynamic programming algorithm based on the described decomposition.

After the best matching is found, the matched pairs of vertices are merged. The resulting DAG contains the merged matched vertices plus the nonmatched vertices from the original DAGs G, H . The edges are merged accordingly, and transitive closure is applied. If vertices P, Q are matched and merged into a vertex R , then:

$$w_R = w_P + w_Q \quad \mathbf{u}_R = \frac{\mathbf{u}_P w_P + \mathbf{u}_Q w_Q}{w_P + w_Q} \quad \mathbf{v}_R = \frac{\mathbf{v}_P w_P + \mathbf{v}_Q w_Q}{w_P + w_Q}$$

7 References

- Armstrong, D.R. *et al.* (2020) PDBe: improved findability of macromolecular structure data in the PDB. *Nucleic Acids Res*, **48**, D335–D343. <https://doi.org/10.1093/nar/gkz990>
- Hutařová Vařeková, I. *et al.* (2021) 2DProts: database of family-wide protein secondary structure diagrams. *Bioinformatics*, **37**, 4599–4601. <https://doi.org/10.1093/bioinformatics/btab505>
- Ilinkin, I. *et al.* (2010) Multiple structure alignment and consensus identification for proteins. *BMC Bioinformatics*, **11**, 71. <https://doi.org/10.1186/1471-2105-11-71>
- Midlik, A. *et al.* (2019) Automated family-wide annotation of secondary structure elements. *Methods Mol Biol*, **1958**, 47–71. https://doi.org/10.1007/978-1-4939-9161-7_3
- Midlik, A. *et al.* (2021) Uncovering of cytochrome P450 anatomy by SecStrAnnotator. *Sci Rep*, **11**, 12345. <https://doi.org/10.1038/s41598-021-91494-8>
- Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, **48**, 443–453. [https://doi.org/10.1016/0022-2836\(70\)90057-4](https://doi.org/10.1016/0022-2836(70)90057-4)
- Schrödinger, LLC. The PyMOL Molecular Graphics System, Version 2.3 <https://pymol.org/>
- Sehnal, D. *et al.* (2021) Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic Acids Res*, **49**, W431–W437. <https://doi.org/10.1093/nar/gkab314>
- Shindyalov, I.N. and Bourne, P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, **11**, 739–747. <https://doi.org/10.1093/protein/11.9.739>
- Sillitoe, I. *et al.* (2021) CATH: increased structural coverage of functional space. *Nucleic Acids Res*, **49**, D266–D273. <https://doi.org/10.1093/nar/gkaa1079>

2DProts: database of family-wide protein secondary structure diagrams

Ivana Hutařová Vařeková^{1,2,3}, Jan Hutař^{1,2}, Adam Midlik^{1,2}, Vladimír Horský^{1,2}, Eva Hladká³, Radka Svobodová^{1,2}, Karel Berka⁴

¹ CEITEC – Central European Institute of Technology, Masaryk University, Kamenice 5, 625 00 Brno, Czech Republic

² National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Kamenice 5, 625 00 Brno, Czech Republic

³ Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic

⁴ Department of Physical Chemistry, Regional Centre of Advanced Technologies and Materials, Faculty of Science, Palacký University, 17. listopadu 1192/12, 771 46 Olomouc, Czech Republic

Bioinformatics, 37: 4599–4601, 2021.

<https://doi.org/10.1093/bioinformatics/btab505>

Databases and ontologies

2DProts: database of family-wide protein secondary structure diagrams

Ivana Hutařová Vařeková^{1,2,3}, Jan Hutar^{1,2}, Adam Midlik^{1,2}, Vladimír Horský^{1,2},
Eva Hladká³, Radka Svobodová^{1,2,*} and Karel Berka^{4,*}

¹National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Brno 625 00, Czech Republic, ²Centre for Structural Biology, CEITEC—Central European Institute of Technology, Masaryk University, Brno 625 00, Czech Republic, ³Department of Computer Systems and Communications, Faculty of Informatics, Masaryk University, Brno 602 00, Czech Republic and ⁴Department of Physical Chemistry, Faculty of Science, Palacký University Olomouc, Olomouc 771 46, Czech Republic

*To whom correspondence should be addressed

Associate Editor: Peter Robinson

Received on January 29, 2021; revised on June 28, 2021; editorial decision on July 4, 2021; accepted on July 5, 2021

Abstract

Summary: Secondary structures provide a deep insight into the protein architecture. They can serve for comparison between individual protein family members. The most straightforward way how to deal with protein secondary structure is its visualization using 2D diagrams. Several software tools for the generation of 2D diagrams were developed. Unfortunately, they create 2D diagrams based on only a single protein. Therefore, 2D diagrams of two proteins from one family markedly differ. For this reason, we developed the 2DProts database, which contains secondary structure 2D diagrams for all domains from the CATH and all proteins from PDB databases. These 2D diagrams are generated based on a whole protein family, and they also consider information about the 3D arrangement of secondary structure elements. Moreover, 2DProts database contains multiple 2D diagrams, which provide an overview of a whole protein family's secondary structures. 2DProts is updated weekly and is integrated into CATH.

Availability and Implementation: Freely accessible at <https://2dprots.ncbr.muni.cz>. The web interface was implemented in JavaScript. The database was implemented in Python.

Contact: radka.svobodova@ceitec.muni.cz or karel.berka@upol.cz

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Nowadays, there are more than 170 000 structures in Protein Data Bank (PDB) (Armstrong *et al.*, 2020). These structural data enabled the establishment of rich datasets describing individual protein families. Specifically, close to 7000 protein families are listed in the CATH database (Sillitoe *et al.*, 2021), and, for some of them, hundreds to thousands of their member proteins' structures have been determined. These structures originate from different organisms, bind various ligands and contain diverse mutations. These data provide us with a robust basis for examining individual protein families, discovering their essential parts and understanding their structure-function relationships.

A key insight into the structure of a protein is often provided by the visualization of its secondary structures, i.e., the spatial organization of its secondary structure elements (SSEs) such as α -helices and β -sheets. The most straightforward way to compare protein structures within a protein family might be its visualization using secondary structure 2D diagrams.

Several tools for the 2D visualization of protein secondary structure have been developed (e.g. PROMOTIF; Hutchinson and Thornton, 1996, Pro-origami; Stivala *et al.*, 2011, HERA; Hutchinson and Thornton, 1990). Unfortunately, these tools typically operate with one protein structure. Therefore, two similar proteins from one protein family can have very different 2D diagrams, because they do not consider the global positions of secondary structure elements in space, but only local structure usually within a rectangular grid. These tools are also not able to provide a 2D diagram of multiple secondary structures, e.g., secondary structures of all members of a protein family.

To fill this gap, we have developed the 2DProts database: A comprehensive and up-to-date resource providing secondary structure 2D diagrams for all protein domains from PDB database and multiple 2D diagrams for all protein families from the CATH database.

Main goals of 2DProts are minimization of the error of secondary structure projection from 3D to 2D; highlighting similarities of protein families in each 2D diagram; preservation of differences

5. MAIN PUBLICATIONS

between protein 3D models within a protein family; and visualization of these differences in 2D diagrams of secondary structures of proteins from such family.

2 Algorithm

The algorithm of 2DProts works as follows:

Input: A CATH superfamily (e.g. 2.60.120.400), the list of its domains (e.g. 1gzA00, 1ourA00, ...), and the PDB structures of these domains.

Step 1: For each domain in the given family, find its SSEs (via SecStrAnnotator (Midlik *et al.*, 2019, 2021)) and annotate them in such a way that topologically equivalent SSEs have the same name (via SecStrAnnotator).

Step 2: For each group of SSEs with the same name, compute average length and frequency of SSE occurrence.

Step 3: For each domain in the family:

Step 3.1: Try to select an appropriate starting layout among the previously computed domains.

Step 3.2: Group all β -strands into sheets and compute a 2D model of each individual sheet.

Step 3.3: Divide the helices and sheets into primary (common for most of the domains) and secondary (the remaining ones).

Step 3.4: Place all primary helices and sheets into the 2D diagram.

Step 3.5: Adjust the angles of the primary helices and sheets.

Step 3.6: Add all secondary helices and sheets into the 2D diagram.

Step 3.7: Adjust the angles of the secondary helices and sheets.

Step 4: Draw an individual 2D diagram for each domain and a common multiple 2D diagram for the whole family.

Please note, that steps 3.2 to 3.7 employ an optimization algorithm to minimize the error of projection from 3D to 2D and restrict the deviation from the starting layout. Details of the algorithm are described in the [Supplementary Material](#).

3 Database contents and functionality

For each PDB structure, 2DProts provides 2D diagrams of all its domains (an example of such a diagram for a protein in [Fig. 1a](#) is in [Fig. 1c](#)). On top of that, for each protein family, 2DProts provides multiple 2D diagrams of all domains occurring in the family (see example in [Fig. 1b and d](#)). In total, 2DProts contains a visualization of more than 400 000 protein domains from all PDB protein structures. Individual protein domains are grouped into about 7000 protein families according to the CATH database. Moreover, 2DProts is able to generate 2D and multiple 2D diagrams also for user-defined protein families. 2DProts database is freely accessible at <https://2dprots.ncbr.muni.cz>.

The website of the database includes documentation that explains the methodology, user manual for the database, and four examples of scientifically interesting families (porin, cytochrome reductase, cytochrome P450 and methionine sulfoxide reductase). The last example also demonstrates the possibilities of a more detailed clustering.

Each protein domain is represented in the 2DProts database via its identifier in the CATH format (e.g. 1r9nA01), and its 2D diagram is findable using this identifier. It is also possible to search for all domains of a specific protein using its PDB ID (e.g. 1r9n). Protein families can be found via a CATH identifier (e.g. 2.140.10.20) that can be used in the search field to obtain a multiple 2D diagram of the family.

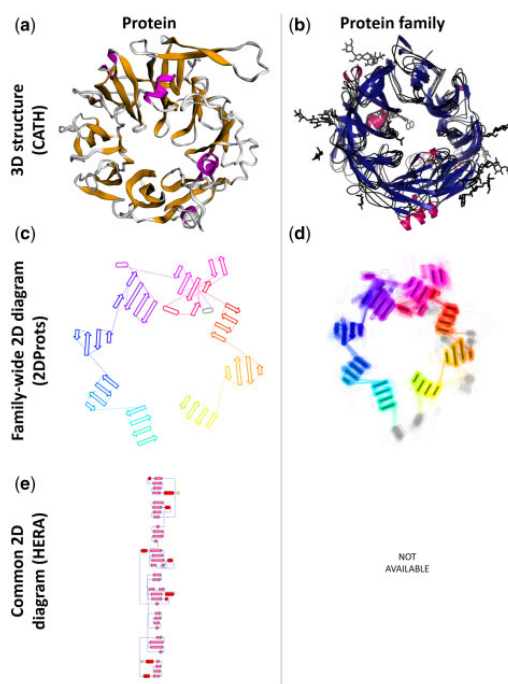


Fig. 1. Example of 3D structures and 2D diagrams of a cytochrome reductase domain (PDB ID 1orw, CATH identifier 1orwB01) from protein family with CATH ID 2.140.10.30. Panels show from top left—(a) representative 3D structure of the domain (visualization obtained from CATH), (b) 3D structures of the all the domains from the family (visualization obtained from CATH), (c) representative 2D diagram of the domain from 2DProts, (d) multiple 2D diagram of all domains from the family from 2DProts, with SSE averages through protein family and transparency, (e) 2D diagram of the domain (visualization obtained from HERA; Hutchinson and Thornton, 1990).

4 Discussion

In comparison with other tools generating 2D diagrams of protein secondary structure, 2DProts has the following three marked advantages: Firstly, 2DProts reflects 3D arrangement of SSEs (see [Fig. 1c](#)), whereas other tools do not (e.g. see [Fig. 1e](#)). Consequently, if some SSEs are close in 3D, they are also close in 2D and vice-versa. Therefore, the 2D visualization is more intuitive. Secondly, 2D diagrams for individual protein family members are intercomparable and can show differences among family members. Thirdly, 2DProts provides multiple 2D diagrams, which can serve as an overview of a domain arrangement within a whole protein family.

Applicability of 2DProts is demonstrated by the fact that 2D diagrams and multiple 2D diagrams from 2DProts were recently integrated into the CATH database (Sillitoe *et al.*, 2021). Its integration into other resources (e.g. PDBE-KB (PDBE-KB consortium, 2020)) is planned. Source code of the website is available at <https://gitlab.com/jhutar/2dprot-web>.

Funding

This work was supported by Ministry of Education, Youth and Sports of the Czech Republic under the ELIXIR CZ research infrastructure project, including access to computing and storage facilities [grant number LM2018131]; and European Regional Development Fund—project ELIXIR-CZ [grant number CZ.02.1.01/0.0/0.0/16_013/0001777].

Conflict of Interest: none declared.

Data availability statement

The input data for the 2DProts database are sourced from the PDB database (<https://www.ebi.ac.uk/pdbe>) and the CATH database (<https://www.cathdb.info>). The SecStrAnnotator used by 2DProts is available at <https://sestra.ncbr.muni.cz>. Database 2DProts with all computed 2D diagrams is freely accessible at <https://2dprots.ncbr.muni.cz>. Source code of the website of 2DProts is available at <https://gitlab.com/jhutar/2dprot-web>.

References

- Armstrong, D.R. *et al.* (2020) PDB: improved findability of macromolecular structure data in the PDB. *Nucleic Acids Res*, **48**, D335–D343.
- Hutchinson, E.G., and Thornton, J.M. (1990) HERA—a program to draw schematic diagrams of protein secondary structures. *Proteins Struct Funct Genet*, **8**, 203–212.
- Hutchinson, E.G., and Thornton, J.M. (1996) PROMOTIF - a program to identify and analyze structural motifs in proteins. *Protein Sci*, **5**, 212–220.
- Midlik, A. *et al.* (2019) Automated family-wide annotation of secondary structure elements. In: Kister, A.E. (ed.) *Protein Supersecondary Structures*. Humana Press, New York, pp. 47–71.
- Midlik, A. *et al.* (2021) Uncovering of cytochrome P450 anatomy by SecStrAnnotator. *Sci Rep* **11**.
- PDBE-KB consortium (2020) PDBE-KB: a community-driven resource for structural and functional annotations. *Nucleic Acids Res*, **48**, D344–D353.
- Sillitoe, I. *et al.* (2021) CATH: increased structural coverage of functional space. *Nucleic Acids Res*, **49**, D266–D273.
- Stivala, A. *et al.* (2011) Automatic generation of protein structure cartoons with Pro-origami. *Bioinformatics*, **27**, 3315–3316.

Supplementary Information for 2DProts: Database of Family-Wide Protein Secondary Structure Diagrams

Ivana Hutařová Vařeková^{1,2,3}, Jan Hutař^{1,2}, Adam Midlik^{1,2}, Vladimír Horský^{1,2}, Eva Hladká³, Radka Svobodová^{1,2,*}, Karel Berka^{4,*}

¹National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Brno 625 00, Czech Republic,

²CEITEC - Central European Institute of Technology, Masaryk University, Brno 625 00, Czech Republic,

³Faculty of Informatics, Masaryk University, Brno 602 00, Czech Republic, and

⁴Department of Physical Chemistry, Faculty of Science, Palacký University, Olomouc 771 46, Czech Republic

Table of Contents

1. Terminology.....	1
2. Methods	2
Step 1: Detection and annotation of SSEs	2
Step 2: Pre-processing of the CATH superfamily	2
Step 3: Processing of individual domains.....	2
Step 3.1: Search for cluster start domain	2
Step 3.2: Joining strands into sheets and generating 2D model for each sheet.....	2
Step 3.3: Division of helices and sheets into primary and secondary	4
Step 3.4: Placing of all primary helices and sheets into the 2D diagram.....	5
Step 3.5: Adjustment of angles of primary helices and sheets in the 2D diagram.....	5
Step 3.6: Adding all secondary helices and sheets into the 2D diagram.....	5
Step 3.7: Adjustment of secondary SSEs angles.....	6
Step 4: Drawing 2D diagrams.....	6
3. Bibliography	8

1. Terminology

The 2D diagram is represented as a plane, which contains an x-axis and y-axis. These axes cross in the origin, at the point (0,0).

The following information defines each secondary structure element (SSE) in a 2D diagram:

- the type of SSE (a helix or a strand),
- the name of the SSE (e.g., H1, E2),
- the 2D position of the centre of the SSE in the above plane,
- the angle of the SSE with respect to the x-axis,
- the length of the SSE,
- the colour of the SSE,
- the start and end residue of the SSE, and
- the beta-connectivity between strands in each sheet.

Note: 2DProts can provide this information in a JSON file.¹

¹ e.g., https://2dprots.ncbr.muni.cz/static/web/layouts/generated-1.10.30.10/layout-4nod_G01.json

In the 2D diagrams, we also introduce one visualization tweak: Fixed ratio of 1:0.2 between the distances between SSEs and the sizes of SSEs. This allows us to place the SSEs comfortably in the 2D diagram without too many overlapping SSEs.

2. Methods

The 2DProts workflow can be described using the following steps:

Step 1: Detection and annotation of SSEs

For a given CATH superfamily, we process the PDB structures of all of its domains. Then, using SecStrAnnotator² (Midlik *et al.*, 2019, 2021), we (i) detect all SSEs present in the protein domains and (ii) annotate all topologically equivalent SSEs with the same name.

Step 2: Pre-processing of the CATH superfamily

For each group of SSEs with the same name (e.g., group of helices denoted H5), we compute statistical data, i.e., its average length (measured in the number of residues) and probability of occurrence.

In this step, we also find a universal start domain of the family. First, we choose seven random domains. Then, we compute their mutual RMSD. Finally, we choose the domain with the lowest sum of RMSD with the other domains as the universal start domain for the whole family. When we update the family (and not generate a new diagram from scratch), the existing start domain from the previous version is used instead.

Step 3: Processing of individual domains

The third step is composed of 7 substeps necessary to compute the positions of the SSEs in 2D diagram for a domain and multiple 2D diagrams of the whole protein family.

Step 3.1: Search for cluster start domain

For each domain, we try to find its cluster start domain, i.e., a domain in the same CATH S35 cluster³ for which the 2D diagram has already been computed. If there is no such domain, we use the universal start domain instead. If the universal start domain has not yet been computed, no start domain is considered.

Step 3.2: Joining strands into sheets and generating 2D model for each sheet

Joining strands into sheets

For each domain, we first sort all of its strands into sheets (i.e., groups of strands that are interconnected by hydrogen bonds) based on connection data that have been computed by the SecStrAnnotator tool. Then, we remove all sheets that only contain strands that are shorter than two residues.

Generating a 2D model for each sheet

For each sheet, we perform the following process as shown in Figure S1:

- **Initial 2D position of strands:** We set the initial 2D position of all the strands to position based on the 2D position of the start domain (if it is present). If it is not present, we set the initial position as (0,0).
- **Rough sheet 2D model:** We use a sequence of optimization algorithms (based on a modified alternating variable method (Korel, 1990)) to compute a “rough” sheet 2D model. In each iteration of the optimization algorithm, we minimize the error of sheet projection to the 3D structure. More precisely, we minimize the error between the minimal strand distance in the 3D structure of the domain and their distance in our 2D diagram. Specifically, we minimize this expression:

² <https://sestra.ncbr.muni.cz>

³ S35 cluster groups domains, guaranteed to share at least 35% sequence identity.

$$\sum_{s_1, s_2 \in S_\beta} |d_{3D}(s_1, s_2) - d_{2D}(s_1, s_2)| (L(s_1) + L(s_2)) \left(1 - \frac{\min(d_{3D}(s_1, s_2), d_{2D}(s_1, s_2))}{40} \right)$$

where S_β is the set of all strands in the sheet; $d_{3D}(s_1, s_2)$ is the minimal distance of strand s_1 and strand s_2 in the 3D structure; $d_{2D}(s_1, s_2)$ is the distance of the positions of s_1 and s_2 in our 2D diagram; and $L(s)$ is the length of strand s (i.e., its number of residues).

The parameter adjusted in each iteration of the optimization algorithm is the upper limit for distances between connected strands. The limit is gradually reduced from up to 30 Å to 4 Å. The starting limit depends on the size of the sheet, on the presence of the start domain 2D diagram, and on the actual results.

Big sheets, whose strands form a barrel conformation (e.g., families 2.40.160.10 and 2.160.20.20) and have no start domain 2D diagram available, need to have a high limit value to reach a similar shape to that of the 3D structure. With a stricter limit, the expected shape is either not reached, or is more time-consuming to reach.

Small sheets with 3 to 6 strands need fewer optimization algorithm steps with lower limit values to reach their expected shape.

- **Improved sheet 2D model:** When this procedure is finished, we have a rough 2D diagram of the whole sheet. In the source 3D structure, distances of neighbouring strands in sheets are typically between 2–3 Å. However, after processing, our 2D diagram model distances are between 0–4 Å. To improve the diagram and to make the sheet 2D diagram more realistic, we iterate the optimization algorithm once more.

In this iteration, we choose one main direction of our sheet 2D model and optimize the coordinates of all strands in a way that sets the distance between each connected neighbour to be around 2.5 Å.

- **Final sheet 2D model:** Afterwards, we set the angle of the sheet. All strands in a sheet are either parallel or anti-parallel. They are all orthogonal to the main direction of the sheet. We therefore have to choose from two possible reflections of the whole strand. If we have a start 2D diagram, we choose the position that bears a stronger similarity to the one in the start domain. Otherwise, we choose the position that better represents the 3D structure of the SSEs.

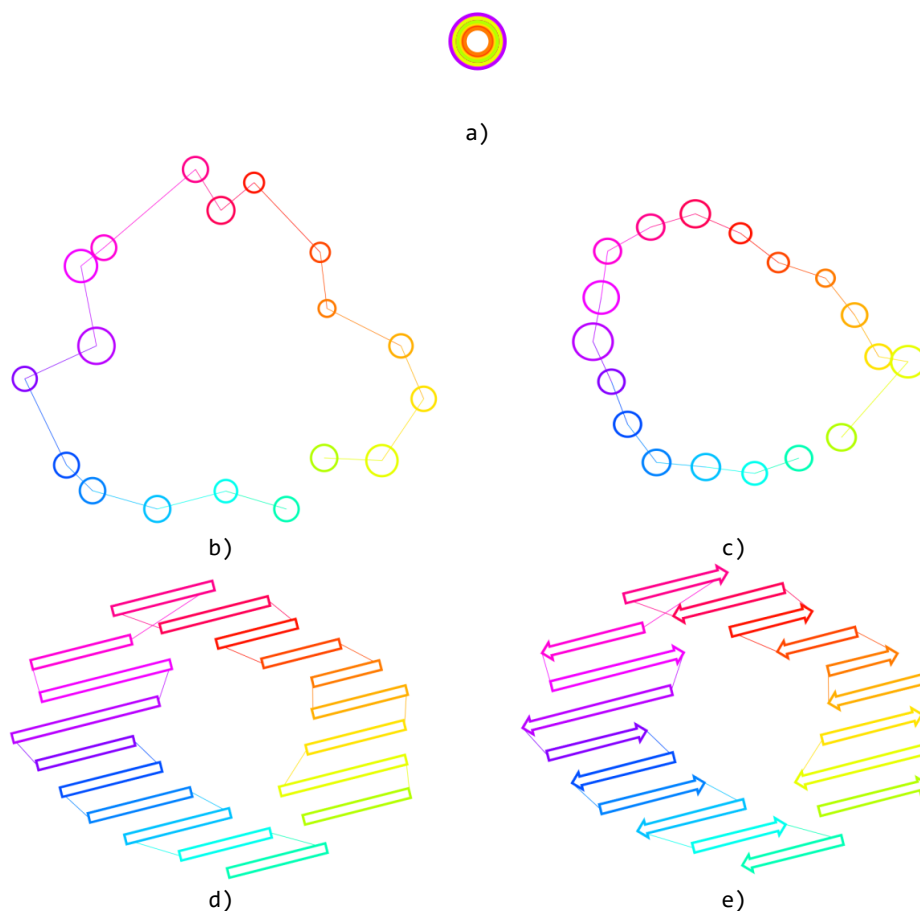


Figure S1: Example of generating sheet 2D model of barrel 3rbhA00 domain (a barrel with 18 strands): a) initial 2D position of strands in origin, b) rough sheet 2D model (first iteration), c) rough sheet 2D model (last iteration), d) improved sheet 2D model, e) final sheet 2D model.

Step 3.3: Division of helices and sheets into primary and secondary

We divide all helices and strands into two sets:

- Primary helices and strands: Common to the majority⁴ of domains in the given family
- Secondary helices and strands: Helices and strands which are not primary

The algorithm then divides sheets based on the above strand division:

- Primary sheets contain at least one primary strand
- Secondary sheets do not contain any primary strands.

⁴ Majority means they are present in at least 80 % of the domains of the family.

Step 3.4: Placing of all primary helices and sheets into the 2D diagram

As the first step, we place all primary helices and sheets into the 2D diagram. This procedure helps us to make the set of diagrams of the whole family more comparable.

- If there is a start domain, we try to find the positions of primary helices and sheets in the start domain. If there is a helix or sheet position, we use it as the starting position for a given helix or sheet. If not, we set the initial position as (0,0).
- Then, we use an optimization algorithm. In this algorithm, we try to move primary helices and move, rotate, and reflect primary sheets.

During early iterations of the optimization algorithm, we use bigger movement steps and bigger rotation angles. In later iterations, we reduce the length of the movement steps and the rotation angle.

The optimization algorithm minimizes the difference between distances separating the SSEs in the input 3D structure and distances in our 2D diagram. Moreover, to prevent ending at an incomparable minimum⁵, we penalize deviation from the start 2D diagram (if there is a start 2D diagram). In this step, we minimize the following function:

$$\sum_{s_1, s_2 \in S_p} |d_{3D}(s_1, s_2) - d_{2D}(s_1, s_2)| (L(s_1) + L(s_2)) + \sum_{s \in S_p \cap S_0} |r(s) - r_0(s)| L(s) \frac{|S_p|}{20}$$

where S_p is the set of all primary SSEs (helices and sheets) of the current domain; S_0 is the set of all SSEs in the layout of the start domain (or empty set if there is no start domain); $d_{3D}(s_1, s_2)$ is the minimal distance of SSEs s_1 and s_2 in the 3D structure; $d_{2D}(s_1, s_2)$ is the distance of the positions of s_1 and s_2 in our 2D diagram; $L(s)$ is the length of SSE s (i.e., for a helix the number of its residues, for a sheet the total number of residues in all its strands); $r(s)$ is the 2D position of SSE s in the layout of the current domain; and $r_0(s)$ is the 2D position of SSE s in the layout of the start domain.

Step 3.5: Adjustment of angles of primary helices and sheets in the 2D diagram

As the next step, we optimize the angle of the primary SSEs. We already set fixed angles for all sheets. However, we need the algorithm to determine the angles of the helices.

- If a helix is in the start domain, we use the angle value from the start domain as the initial value. Otherwise, we set the initial angle to 0°.
- Next, we use an optimization algorithm to minimize the difference between the angles of SSEs in the input 3D structure and our 2D diagram. Specifically, we minimize the following expression:

$$\sum_{s_1, s_2 \in S_p} |\varphi_{3D}(s_1, s_2) - \varphi_{2D}(s_1, s_2)| (L(s_1) + L(s_2))$$

where S_p is the set of all primary SSEs of the current domain; $\varphi_{3D}(s_1, s_2)$ is the angle between SSEs s_1 and s_2 in the 3D structure; $\varphi_{2D}(s_1, s_2)$ is the angle between s_1 and s_2 in the 2D diagram; and $L(s)$ is the length of SSE s (i.e., for a helix its number of residues, for a sheet the total number of residues in all of its strands).

Step 3.6: Adding all secondary helices and sheets into the 2D diagram

At this point, all primary SSEs have their positions. It is then necessary to carry out the same process with the secondary structures with respect to all the primary SSEs, whose positions have already been set.

- First, we find the 2D diagram of secondary helices and sheets in the start domain. Then, if such a 2D diagram exists, we use it as a start 2D diagram. If not, we set position (0,0) to every secondary SSE without a reference 2D diagram in the start domain.

⁵ For one 3D structure, there could be several 2D diagrams with the same deviation from the original 3D structure but with fundamental differences between them.

- Then, we use an optimization algorithm. In this algorithm, we try to move secondary helices and move, rotate, and reflect secondary sheets. First, we use bigger movement steps. Then, during subsequent iterations, we reduce the length of movement steps and the rotation angle. Thus, the optimization algorithm tries to minimize the difference between distances of SSEs in the 3D structure and our 2D diagram with respect to all primary SSEs with firm positions. Moreover, because we want to prevent ending at an incomparable minimum, we penalize the deviation from the start 2D diagram, if it is present. The minimized expression is:

$$\sum_{s_1 \in S_s, s_2 \in S} |d_{3D}(s_1, s_2) - d_{2D}(s_1, s_2)| (L(s_1) + L(s_2)) + \sum_{s \in S_s \cap S_0} |r(s) - r_0(s)| L(s) \frac{|S|}{20}$$

where S is the set of all SSEs of the current domain; S_s is the set of all secondary SSEs of the current domain; $d_{3D}(s_1, s_2)$ is the minimal distance of SSEs s_1 and s_2 in the 3D structure; $d_{2D}(s_1, s_2)$ is the distance of the positions of s_1 and s_2 in our 2D diagram; $L(s)$ is the length of SSE s (i.e., for a helix its number of residues, for a sheet the total number of residues in all of its strands); $r(s)$ is the 2D position of SSE s in the layout of the current domain; and $r_0(s)$ is the 2D position of SSE s in the layout of the start domain.

Step 3.7: Adjustment of secondary SSEs angles

This adjustment is done in the following way:

- If a secondary helix is in the start domain, we use the angle of the secondary helix from the start domain as the initial value. Otherwise, the initial value is set to 0°.
- Then, we use an optimization algorithm to minimize the difference between angles of SSEs in the input 3D structure and our 2D diagram. The minimized expression is:

$$\sum_{s_1 \in S_s, s_2 \in S} |\varphi_{3D}(s_1, s_2) - \varphi_{2D}(s_1, s_2)| (L(s_1) + L(s_2))$$

where S is the set of all SSEs of the current domain; S_s is the set of all secondary SSEs of the current domain; $\varphi_{3D}(s_1, s_2)$ is the angle between s_1 and s_2 in the 3D structure; $\varphi_{2D}(s_1, s_2)$ is the angle between s_1 and s_2 in the 2D diagram; and $L(s)$ is the length of SSE s (i.e., for a helix its number of residues, for a sheet the total number of residues in all its strands).

Step 4: Drawing 2D diagrams

The last part of the process is to draw an output image that depicts a 2D diagram of helices and sheets and their angles.

2DProts can draw a 2D diagram for one domain or a multiple 2D diagram of a set of domains, which is either the whole protein family, a CATH S35 cluster, or some other subset.

An example of a 2D diagram and the corresponding 3D structure can be found in Figure S2. Examples of four multiple 2D diagrams together with the input 3D structures are in Figures S3–S6, demonstrating protein families with a β -barrel, a β -propeller, a helix bundle, and an α/β domain, respectively.

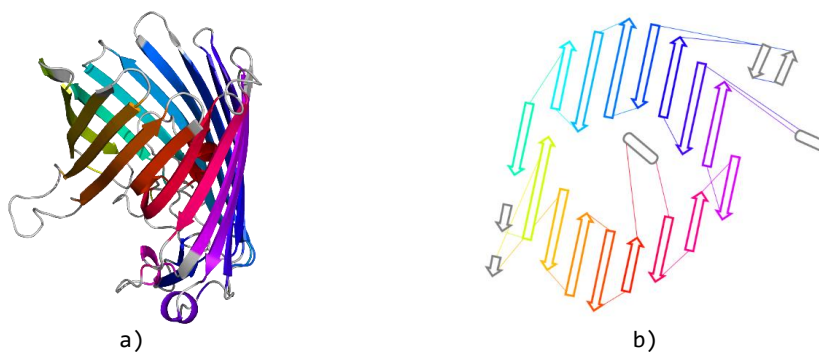


Figure S2: The domain ImpfA00: a) 3D structure, b) 2D diagram from 2DProts.



Figure S3: Domain family 2.40.160.10, containing a β -barrel. a) 3D structure, b) multiple 2D diagram from 2DProts.

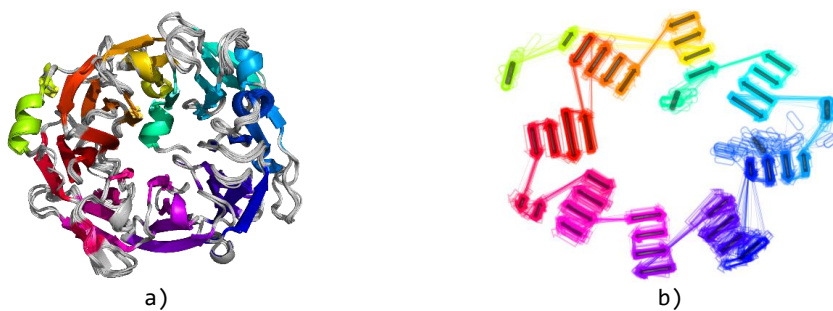


Figure S4: Domain family 2.140.10.20, containing 8 β -sheets arranged in a β -propeller. a) 3D structure model, b) multiple 2D diagram from 2DProts.



Figure S5: Domain family 1.20.1260.10, containing a helix bundle. a) 3D structure, b) multiple 2D diagram from 2DProts.



Figure S6: Domain family 3.40.50.40, containing an alpha/beta domain. a) 3D structure, b) multiple 2D diagram from 2DProts.

3. Bibliography

Korel,B. (1990) Automated Software Test Data Generation. *IEEE Trans. Softw. Eng.*, **16**, 870–879.

Midlik,A. *et al.* (2019) Automated family-wide annotation of secondary structure elements. In, Kister,A.E. (ed), *Protein supersecondary structures*. Humana Press, New York, pp. 47–71.

Midlik,A. *et al.* (2021) Uncovering of cytochrome P450 anatomy by SecStrAnnotator. *Sci. Rep.*, **11**.

Chapter 6

Other Publications

LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data

David Sehnal^{1,2,3}, Mandar Deshpande³, Radka Svobodová Vařeková^{1,2}, Saqib Mir³, Karel Berka⁴, Adam Midlik^{1,2}, Lukáš Pravda^{1,2}, Sameer Velankar³, Jaroslav Koča^{1,2}

¹ CEITEC – Central European Institute of Technology, Masaryk University, Brno, Czech Republic

² National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Brno, Czech Republic

³ Protein Data Bank in Europe (PDBe), European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, UK

⁴ Regional Centre of Advanced Technologies and Materials, Department of Physical Chemistry, Faculty of Science, Palacký University, Olomouc, Czech Republic

Nature Methods, 14: 1121–1122, 2017.

<https://doi.org/10.1038/nmeth.4499>

LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data

To the Editor: We present the LiteMol suite, a tool for visualizing large macromolecular structure data sets that is freely available at <https://www.litemol.org>.

Given rapid advances in electron microscopy and other techniques for determining macromolecular structure, many structures that were previously intractable on account of their size and complexity are now amenable to study at the molecular level. Interactive web-based visualizations that include underlying experimental evidence and rich biological context annotations are critical in exploiting the wealth of these structural data¹. While online resources such as PDB², EMDB³, and others make it possible to access these data, the delivery and visualization of large data sets remain challenging (Supplementary Note 1). To address these challenges, we developed the LiteMol suite.

The LiteMol suite consists of three components (Fig. 1a): data delivery services (CoordinateServer and DensityServer), the BinaryCIF compression format, and a new lightweight 3D molecular viewer (LiteMol Viewer) (Supplementary Methods). Together, these components enable near-instant delivery and

visualization of large macromolecular data sets and speed that is orders of magnitude faster than that of previously available solutions (Supplementary Notes 2 and 3). The LiteMol suite works on all modern web browsers and mobile devices, and this makes macromolecular structure data available to diverse communities of users with and without structural biology expertise.

The data delivery services can dynamically extract subsets of coordinate and experimental data to substantially reduce the network transfer size. CoordinateServer uses a rich molecular query language⁴ to select only those atomic coordinates necessary for the requested visualization (e.g., a ligand-binding site). DensityServer provides experimental maps (e.g., from X-ray or cryo-electron microscopy experiments) as a full-resolution slice (e.g., around a ligand) for a detailed view, or as a downsampled complete map of the entire structure and its general features. Both services use the newly developed BinaryCIF format to further reduce the volume of transferred data.

The BinaryCIF compression format provides a uniform data-storage framework for macromolecular structure data (including experimental maps and annotations), and this removes the need for handling multiple file formats. Standard PDBx/mmCIF dictionary definitions, provided by the wwPDB consortium², are used to store macromolecular models, and this facilitates straightforward adaptation of existing software to use BinaryCIF. These features make BinaryCIF an important improvement over

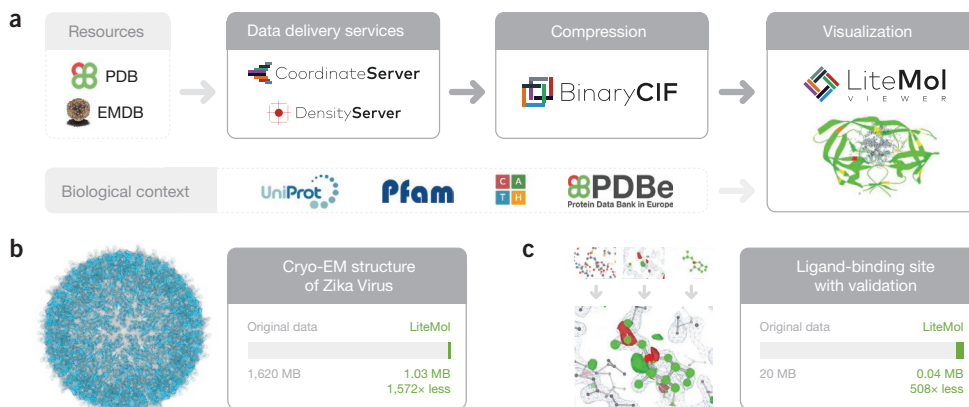


Figure 1 | LiteMol suite architecture and case studies. **(a)** Architecture of the LiteMol suite. CoordinateServer and DensityServer provide an interface to online resources (e.g., PDBe and EMDB) for sending only data relevant for a given visualization. Before transfer, the data is compressed using the BinaryCIF format. LiteMol Viewer provides efficient 3D visualization, including biological context annotations acquired from resources such as UniProt ([uniprot.org](https://www.uniprot.org)), CATH ([cathdb.info](https://www.ebi.ac.uk/cath)), Pfam ([pfam.xfam.org](https://www.ebi.ac.uk/pfam)), PDBe (<https://www.ebi.ac.uk/pdbe/>), and others. **(b)** Illustration of using the LiteMol suite to visualize the cryo-electron microscopy structure of the Zika virus (PDB ID 5IRE; assembly 1) and its underlying experimental map (EMD-8116) using downsampled data. An interactive example is available at <https://viewer.litemol.org/?example=zika-cryo-em>. **(c)** Detailed visualization of N-acetylglucosamine in hyperthermophilic chitinase (residue NAG B 2 in PDB ID 3A4X) via the LiteMol suite. The visualization includes the atomistic model of the ligand's binding site, X-ray experimental map, and ligand structure validation annotation provided by a third-party service (ncbr.muni.cz/ValidatorDB/). An interactive example is available at <https://viewer.litemol.org/?example=3a4x-lig>.

Sanguinarine is reduced by NADH through a covalent adduct

Roman Sándor¹, Jiří Slanina¹, Adam Midlik^{2,3}, Kristýna Šebrlová¹, Lucie Novotná¹, Martina Čarnecká¹, Iva Slaninová⁴, Petr Táborský⁵, Eva Táborská¹, Ondřej Peš¹

¹ Department of Biochemistry, Faculty of Medicine, Masaryk University, Brno, Czech Republic

² National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Brno, Czech Republic

³ Central European Institute of Technology CEITEC MU, Brno, Czech Republic

⁴ Department of Biology, Faculty of Medicine, Masaryk University, Brno, Czech Republic

⁵ Department of Chemistry, Faculty of Science, Masaryk University, Brno, Czech Republic

Phytochemistry, 145: 77–84, 2018.

<https://doi.org/10.1016/j.phytochem.2017.10.010>



Contents lists available at ScienceDirect

Phytochemistry

journal homepage: www.elsevier.com/locate/phytochem

Sanguinarine is reduced by NADH through a covalent adduct



Roman Sandor ^a, Jiri Slanina ^a, Adam Midlik ^{b, c}, Kristyna Sebrlova ^a, Lucie Novotna ^a,
Martina Carnecka ^a, Iva Slaninova ^d, Petr Taborsky ^e, Eva Taborska ^a, Ondrej Pes ^{a, *}

^a Department of Biochemistry, Faculty of Medicine, Masaryk University, Kamenice 5, 62500 Brno, Czech Republic

^b National Centre for Biomolecular Research, Faculty of Science, Masaryk University Brno, Kamenice 5, 62500 Brno, Czech Republic

^c Central European Institute of Technology CEITEC MU, Kamenice 5, 62500 Brno, Czech Republic

^d Department of Biology, Faculty of Medicine, Masaryk University, Kamenice 5, 62500 Brno, Czech Republic

^e Department of Chemistry, Faculty of Science, Masaryk University, Kamenice 5, 62500 Brno, Czech Republic

ARTICLE INFO

Article history:

Received 21 April 2017

Received in revised form

26 October 2017

Accepted 27 October 2017

Available online 5 November 2017

Keywords:

Benzophenanthridine alkaloids

Ene adduct

Hydride transfer

LC-MS

NADH

NADH depletion

Redox cycling

Sanguinarine

ABSTRACT

Sanguinarine is a benzo[c]phenanthridine alkaloid with interesting cytotoxic properties, such as induction of oxidative DNA damage and very rapid apoptosis, which is not mediated by p53-dependent signaling. It has been previously documented that sanguinarine is reduced with NADH even in absence of any enzymes while being converted to its dihydro form. We found that the dark blue fluorescent species, observed during sanguinarine reduction with NADH and misinterpreted by Matkar et al. (Arch. Biochem. Biophys. 2008, 477, 43–52) as an anionic form of the alkaloid, is a covalent adduct formed by the interaction of NADH and sanguinarine. The covalent adduct is then converted slowly to the products, dihydrosanguinarine and NAD⁺, in the second step of reduction. The product of the reduction, dihydrosanguinarine, was continually re-oxidized by the atmospheric oxygen back to sanguinarine, resulting in further reacting with NADH and eventually depleting all NADH molecules. The ability of sanguinarine to diminish the pool of NADH and NADPH is further considered when explaining the sanguinarine-induced apoptosis in living cells.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Sanguinarine (SA) belongs to a family of plant secondary metabolites called quaternary benzo[c]phenanthridine alkaloids (QBAs). These compounds have been extensively studied for their numerous biological activities, such as antitumor, antimicrobial, antifungal, and anti-inflammatory. While the results have been summarized in several works, the greatest attention is given to the anticancer activity of QBAs (Slaninova et al., 2014; Gaziano et al., 2016). It has been reported that SA is *in vitro* cytotoxic preferentially toward cancer cells than normal cells at concentrations that are comparable to those of the current clinically used anticancer agents (Ahmad et al., 2000). Under physiological conditions, a hydroxide anion (OH⁻) is reversibly attached to the iminium bond of SA to give a 6-hydroxy product called an alkanolamine or a pseudo-base (Fig. 1). The alkanolamine form, which is a nonpolar uncharged molecule, can easily enter a cell to establish a new, pH-

dependent equilibrium between the iminium and alkanolamine form inside the cell. The alkaloid toxicity depends on the ability of the planar, charged quaternary form of SA to produce a stable complex with DNA, which subsequently could affect the cell viability (Slaninova et al., 2001; Vacek et al., 2011). Treatment of cells with SA led to a rapid production of reactive oxygen species (ROS) (Burgeiro et al., 2013), fast and severe glutathione depletion (Debiton et al., 2003), oxidative DNA damage and very rapid apoptosis that was not mediated by p53-dependent DNA damage signaling (Matkar et al., 2008a; Hammerova et al., 2011). The first step in the metabolism of SA in rat liver is the reduction of the quaternary form to dihydrosanguinarine (DHSA) (Fig. 1). The conversion might be mediated by several NAD(P)H dependent oxidoreductases (Deroussent et al., 2010; Wu et al., 2013). In cell cultures of *Eschscholzia californica*, SA is reabsorbed and reduced to DHSA by sanguinarine reductase, which was isolated (Weiss et al., 2006) and characterized (Vogel et al., 2010). Additionally, it has been observed that SA underwent the conversion to its inactive reduced form even when incubated with NADH in the absence of any enzyme (Kovar et al., 1986; Matkar et al., 2008b), however; physiologically important reducing agents, such as glutathione and L-ascorbic acid,

* Corresponding author.

E-mail address: ondramayl@gmail.com (O. Pes).

Visualization and analysis of protein structures with LiteMol suite

David Sehnal^{1,2}, Radka Svobodová^{1,2}, Karel Berka³, Lukáš Pravda^{1,2},
Adam Midlik^{1,2}, Jaroslav Koča^{1,2}

¹ CEITEC – Central European Institute of Technology, Masaryk University, Brno, Czech Republic

² National Centre for Biomolecular Research, Faculty of Science, Masaryk University, Brno, Czech Republic

³ Regional Centre of Advanced Technologies and Materials, Department of Physical Chemistry, Faculty of Science, Palacký University, Olomouc, Czech Republic

In Gáspári,Z. (ed.), *Structural Bioinformatics*.
Humana, New York, NY. Vol. 2112, pp. 1–13, **2020**.
https://doi.org/10.1007/978-1-0716-0270-6_1



Chapter 1

Visualization and Analysis of Protein Structures with LiteMol Suite

David Sehnal, Radka Svobodová, Karel Berka, Lukáš Pravda, Adam Midlik, and Jaroslav Koča

Abstract

LiteMol suite is an innovative solution that enables near-instant delivery of model and experimental biomacromolecular structural data, providing users with an interactive and responsive experience in all modern web browsers and mobile devices. LiteMol suite is a combination of data delivery services (CoordinateServer and DensityServer), compression format (BinaryCIF), and a molecular viewer (LiteMol Viewer). The LiteMol suite is integrated into Protein Data Bank in Europe (PDBe) and other life science web applications (e.g., UniProt, Ensemble, SIB, and CNRS services), it is freely available at <https://litemol.org>, and its source code is available via GitHub. LiteMol suite provides advanced functionality (annotations and their visualization, powerful selection features), and this chapter will describe their use for visual inspection of protein structures.

Key words Protein visualization, Atom selection, Validation report, Ligand representation, Electron density

1 Introduction

Visualization is a critical step in understanding and making effective use of macromolecular structure data. The review by O'Donoghue et al. [1] describes a range of use cases requiring interactive visualization to help answer biological questions, from the basic display of secondary structure to the determination of complex structure-sequence relationships or analysis of ligand binding sites. Moreover, visual inspection of the data obtained from X-ray diffraction experiments (i.e., electron densities) or electron microscopy imaging (i.e., electric potential maps) allows users to assess the quality of the models derived from data.

For these reasons, we have developed LiteMol suite [2], an innovative open-source solution consisting of a 3D molecular visualizer (LiteMol Viewer), data delivery services (CoordinateServer and DensityServer), and a data compression format (BinaryCIF).

Chapter 7

Curriculum Vitae

Adam Midlik

Októbrová 31/A, 080 01 Prešov, Slovakia
+420 776650976
midlik@mail.muni.cz, midlik@gmail.com
<https://publons.com/researcher/CAJ-3491-2022>



Education

- **2016–now: Biomolecular Chemistry, doctoral degree programme**
Masaryk University, Faculty of Science, Brno, Czechia
Thesis: *Annotation and visualization of protein secondary structure*
- **2014–2017: Applied Informatics, Master’s degree programme**
Masaryk University, Faculty of Informatics, Brno, Czechia
Thesis: *Annotation of secondary structure elements in proteins*
- **2013–2016: Analytical Chemistry, Master’s degree programme**
Masaryk University, Faculty of Science, Brno, Czechia
Thesis: *Study on metabolism of benzophenathridine alkaloids*
- **2012–2014: Applied Informatics, Bachelor’s degree programme**
Masaryk University, Faculty of Informatics, Brno, Czechia
Thesis: *Selection of protein fragments using minimal bond breaking*
- **2010–2013: Chemistry, Bachelor’s degree programme**
Masaryk University, Faculty of Science, Brno, Czechia
Thesis: *Study on bioluminescence of Eisenia lucens*

Work experience

- **2017–now: Central European Institute of Technology (CEITEC), Brno**
Research specialist (structural bioinformatics)
- **2013–2015: Department of Biochemistry, Faculty of Medicine, MU, Brno**
Research specialist (metabolic assays, HPLC-MS)

Awards

- **2017:** **Dean's Award** for Master's thesis at FI MU
- **2016:** **Brno Ph.D. Talent** competition for talented doctoral students
- **2014:** **Dean's Award** for excellent academic performance and for Bachelor's thesis at FI MU

International stays

- **2019:** **Vienna BioCenter Core Facilities, Wien, Austria** (3 weeks)
- **2017:** **PDBe, EMBL-EBI, Hinxton, UK** (1 week)
- **2015:** **University of Barcelona, Faculty of Chemistry, Barcelona, Spain**
Erasmus+ study exchange focused on research work (5 months)
Stability study of triplex DNA structures – data analysis of circular dichroism melting experiments using hard and hybrid modelling
- **2011–2012:** **University of Crete, Faculty of Chemistry, Heraklion, Greece**
LLP Erasmus study exchange (6 months)

Publications

Co-authored 7 scientific papers in peer-reviewed journals and 2 book chapters.

Scientific papers

- Midlik,A., Hutařová Vařeková,I., Hutař,J., Charehneu,A., Berka,K., Svobodová,R. (2022) OverProt: secondary structure consensus for protein families. *Bioinformatics*, (in press). <https://doi.org/10.1093/bioinformatics/btac384>.
- Midlik,A., Navrátilová,V., Moturu,T.R., Koča,J., Svobodová,R., Berka,K. (2021) Uncovering of cytochrome P450 anatomy by SecStrAnnotator. *Sci Rep*, **11**, 12345. <https://doi.org/10.1038/s41598-021-91494-8>.
- Hutařová Vařeková,I., Hutař,J., Midlik,A., Horský,V., Hladká,E., Svobodová,R., Berka,K. (2021) 2DProts: database of family-wide protein secondary structure diagrams. *Bioinformatics*, **37**, 4599–4601. <https://doi.org/10.1093/bioinformatics/btab505>.
- Sándor,R., Slanina,J., Midlik,A., Šebrlová,K., Novotná,L., Čarnecká,M., Slaninová,I., Táborský,P., Táborská,E., Peš,O. (2018) Sanguinarine is reduced by

NADH through a covalent adduct. *Phytochemistry*, **145**, 77–84. <https://doi.org/10.1016/j.phytochem.2017.10.010>.

- Sehnal,D., Deshpande,M., Svobodová Vařeková,R., Mir,S., Berka,K., Midlik,A., Pravda,L., Velankar,S., Koča,J. (2017) LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data. *Nature Methods*, **14**, 1121–1122. <https://doi.org/10.1038/nmeth.4499>.
- Sándor,R., Midlik,A., Šebrlová,K., Dovrtělová,G., Nosková,K., Juřica,J., Slani nová,I., Táborská,E., Peš,O. (2016) Identification of metabolites of selected benzophenanthridine alkaloids and their toxicity evaluation. *J Pharm Biomed Anal*, **121**, 174–180. <https://doi.org/10.1016/j.jpba.2016.01.024>.
- Peš,O., Midlik,A., Schlaghamerský,J., Zitnan,M., Táborský,P. (2016) A study on bioluminescence and photoluminescence in the earthworm *Eisenia lucens*. *Photochem Photobiol Sci*, **15**, 175–180. <https://doi.org/10.1039/c5pp00412h>.

Book chapters

- Sehnal,D., Svobodová,R., Berka,K., Pravda,L., Midlik,A., Koča,J. (2020) Visualization and analysis of protein structures with LiteMol suite. In Gáspári,Z. (ed.), *Structural Bioinformatics*. Humana, New York, NY. Vol. 2112, pp. 1–13. https://doi.org/10.1007/978-1-0716-0270-6_1.
- Midlik,A., Hutařová Vařeková,I., Hutař,J., Moturu,T.R., Navrátilová,V., Koča,J., Berka,K., Svobodová Vařeková,R. (2019) Automated family-wide annotation of secondary structure elements. In Kister,A.E. (ed.), *Protein Supersecondary Structures*. Humana Press, New York, NY. Vol. 1958, pp. 47–71. https://doi.org/10.1007/978-1-4939-9161-7_3.

Conference presentations

2 talks and 11 posters at national and international conferences.

Talks

- **ENBIK 2022** – National Bioinformatic Conference, Němčice, Czechia
Discovering the general architecture of protein families with OverProt
- **XX. Meeting of Biochemists and Molecular Biologists 2019**, Brno, Czechia
Creation and visualization of secondary structure consensus for protein families

Posters

- **ELIXIR CZ 2021** – ELIXIR CZ Annual Conference, Prague, Czechia
- **Bringing molecular structure to life: 50 years of the PDB, 2021**, EMBL virtual conference
- **ELIXIR CZ 2020** – ELIXIR CZ Annual Conference, virtual
- **ELIXIR CZ 2019** – ELIXIR CZ Annual Conference, Kurdějov, Czechia
- **ECCB 2018** – European Conference on Computational Biology, Athens, Greece
- **ENBIK 2018** – National Bioinformatic Conference, Bystřice nad Pernštejnem, Czechia
- **ISMB/ECCB 2017** – Intelligent Systems for Molecular Biology & European Conference on Computational Biology, Prague, Czechia
- **ENBIK 2016** – National Bioinformatic Conference, Loučeň, Czechia
- **CECE 2014** – International Interdisciplinary Meeting on Bioanalysis, Brno, Czechia
- **ESAS 2014** – European Symposium on Atomic Spectrometry & Czech - Slovak Spectroscopic Conference, Prague, Czechia
- **CECE 2013** – International Interdisciplinary Meeting on Bioanalysis, Brno, Czechia

Teaching experience

- **2019–2021: Introduction to programming in Python** (lecturer)
- **2017: Introduction to Mathematics – seminar** (seminar tutor)
- **2019: Students' Professional Activities (SOČ) supervision**
(supervision of 1 high school student, *Protein Tunnels in Cytochromes P450*)

Skills

- **Structural bioinformatics:** focus on protein secondary structure, channels, visualization
- **IT:** Python, C#, HTML/CSS/JavaScript, Linux/bash, R, Matlab, LaTeX, machine learning basics
- **Languages:** Slovak – native; English, Czech – fluent; Spanish, Polish, Greek – intermediate

List of Abbreviations

API	Application programming interface
CATH	Class–Architecture–Topology–Homologous superfamily (a database)
CO	Carbonyl group (in protein backbone)
CYP	Cytochrome P450
DAG	Directed acyclic graph
DP	Dynamic programming
DSSP	Define Secondary Structure of Proteins (an SSA algorithm)
ECOD	Evolutionary Classification of protein Domains (a database)
GPCR	G-protein coupled receptor
IDP	Intrinsically disordered protein
IDR	Intrinsically disordered region
IUPR	Intrinsically unstructured protein region (in SCOP2 database)
mmCIF	Macromolecular Crystallographic Information File (a file format)
MOM	Mixed ordered matching (an algorithm)
NADH	Nicotinamide adenine dinucleotide, reduced form
NH	Amide group (in protein backbone)
NMR	Nuclear magnetic resonance
PDB	Protein Data Bank (a database and a file format)
PDBe	Protein Data Bank in Europe
PDBe-KB	Protein Data Bank in Europe – Knowledge Base
PDB ID	Protein Data Bank identifier
PDBsum	Protein Data Bank summaries (a database)
RMSD	Root-mean-square deviation

LIST OF ABBREVIATIONS

SCOP	Structural Classification of Proteins (a database)
SCOPE	SCOP–extended (a database)
SIFTS	Structure Integration with Function, Taxonomy and Sequence (a data resource)
SSA	Secondary structure assignment
SSE	Secondary structure element
TM-score	Template modelling score
1D	1-dimensional
2D	2-dimensional
3D	3-dimensional

References

- [1] Kendrew, J.C., Bodo, G., Dintzis, H.M., Parrish, R.G., Wyckoff, H., and Phillips, D.C. (1958) A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. *Nature*, **181**, 662–666. <https://doi.org/10.1038/181662a0>.
- [2] Velankar, S., Burley, S.K., Kurisu, G., Hoch, J.C., and Markley, J.L. (2021) The Protein Data Bank archive. In Owens, R.J. (ed.), *Structural Proteomics*. Humana, New York, NY. Vol. 2305, pp. 3–21. https://doi.org/10.1007/978-1-0716-1406-8_1.
- [3] Chothia, C. (1992) One thousand families for the molecular biologist. *Nature*, **357**, 543–544. <https://doi.org/10.1038/357543a0>.
- [4] Thornton, J.M., Todd, A.E., Milburn, D., Borkakoti, N., and Orengo, C.A. (2000) From structure to function: approaches and limitations. *Nature Structural Biology*, **7**, 991–994. <https://doi.org/10.1038/80784>.
- [5] Murzin, A.G., Brenner, S.E., Hubbard, T., and Chothia, C. (1995) SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, **247**, 536–540. [https://doi.org/10.1016/S0022-2836\(05\)80134-2](https://doi.org/10.1016/S0022-2836(05)80134-2).
- [6] Sillitoe, I. et al. (2021) CATH: increased structural coverage of functional space. *Nucleic Acids Research*, **49**, D266–D273. <https://doi.org/10.1093/nar/gkaa1079>.
- [7] Andreeva, A., Kulesha, E., Gough, J., and Murzin, A.G. (2020) The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic Acids Research*, **48**, D376–D382. <https://doi.org/10.1093/nar/gkz1064>.
- [8] Jumper, J. et al. (2021) Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**, 583–589. <https://doi.org/10.1038/s41586-021-03819-2>.

- [9] Varadi,M. et al. (2022) AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, **50**, D439–D444. <https://doi.org/10.1093/nar/gkab1061>.
- [10] Midlik,A., Hutařová Vařeková,I., Hutař,J., Moturu,T.R., Navrátilová,V., Koča, J., Berka,K., and Svobodová Vařeková,R. (2019) Automated family-wide annotation of secondary structure elements. In Kister,A.E. (ed.), *Protein Supersecondary Structures*. Humana Press, New York, NY. Vol. 1958, pp. 47–71. https://doi.org/10.1007/978-1-4939-9161-7_3.
- [11] Isberg,V. et al. (2015) Generic GPCR residue numbers - aligning topology maps while minding the gaps. *Trends in Pharmacological Sciences*, **36**, 22–31. <https://doi.org/10.1016/j.tips.2014.11.001>.
- [12] Ehrenmann,F., Kaas,Q., and Lefranc,M.-P. (2010) IMGT/3Dstructure-DB and IMGT/DomainGapAlign: a database and a tool for immunoglobulins or antibodies, T cell receptors, MHC, IgSF and MhcSF. *Nucleic Acids Research*, **38**, D301–D307. <https://doi.org/10.1093/nar/gkp946>.
- [13] Pravda,L., Sehnal,D., Svobodová Vařeková,R., Navrátilová,V., Toušek,D., Berka,K., Otyepka,M., and Koča,J. (2018) ChannelsDB: database of biomacromolecular tunnels and pores. *Nucleic Acids Research*, **46**, D399–D405. <https://doi.org/10.1093/nar/gkx868>.
- [14] Ravichandran,K.G., Boddupalli,S.S., Hasermann,C.A., Peterson,J.A., and Deisenhofer,J. (1993) Crystal structure of hemoprotein domain of P450BM-3, a prototype for microsomal P450's. *Science*, **261**, 731–736. <https://doi.org/10.1126/science.8342039>.
- [15] Ollis,D.L. et al. (1992) The α/β hydrolase fold. *Protein Engineering, Design and Selection*, **5**, 197–211. <https://doi.org/10.1093/protein/5.3.197>.
- [16] Otyepka,M., Skopalík,J., Anzenbacherová,E., and Anzenbacher,P. (2007) What common structural features and variations of mammalian P450s are known to date? *Biochimica et Biophysica Acta (BBA) - General Subjects*, **1770**, 376–389. <https://doi.org/10.1016/j.bbagen.2006.09.013>.
- [17] Carr,P.D., and Ollis,D.L. (2009) Alpha/beta hydrolase fold: an update. *Protein and Peptide Letters*, **16**, 1137–1148. <https://doi.org/10.2174/092986609789071298>.

-
- [18] Lenfant,N., Hotelier,T., Velluet,E., Bourne,Y., Marchot,P., and Chatonnet,A. (2013) ESTHER, the database of the α/β -hydrolase fold superfamily of proteins: tools to explore diversity of functions. *Nucleic Acids Research*, **41**, D423–429. <https://doi.org/10.1093/nar/gks1154>.
- [19] Gotoh,O. (1992) Substrate recognition sites in cytochrome P450 family 2 (CYP2) proteins inferred from comparative analyses of amino acid and coding nucleotide sequences. *The Journal of Biological Chemistry*, **267**, 83–90. ISSN: 0021-9258.
- [20] Zawaira,A., Ching,L.Y., Coulson,L., Blackburn,J., and Wei,Y.C. (2011) An expanded, unified substrate recognition site map for mammalian cytochrome P450s: analysis of molecular interactions between 15 mammalian CYP450 isoforms and 868 substrates. *Current Drug Metabolism*, **12**, 684–700. <https://doi.org/10.2174/138920011796504554>.
- [21] Cojocar,V., Winn,P.J., and Wade,R.C. (2007) The ins and outs of cytochrome P450s. *Biochimica Et Biophysica Acta*, **1770**, 390–401. <https://doi.org/10.1016/j.bbagen.2006.07.005>.
- [22] Strushkevich,N., MacKenzie,F., Cherkesova,T., Grabovec,I., Usanov,S., and Park,H.-W. (2011) Structural basis for pregnenolone biosynthesis by the mitochondrial monooxygenase system. *Proceedings of the National Academy of Sciences*, **108**, 10139–10143. <https://doi.org/10.1073/pnas.1019441108>.
- [23] Rowland,P. et al. (2006) Crystal structure of human cytochrome P450 2D6. *The Journal of Biological Chemistry*, **281**, 7614–7622. <https://doi.org/10.1074/jbc.M511232200>.
- [24] Yu,X., Cojocar,V., and Wade,R.C. (2013) Conformational diversity and ligand tunnels of mammalian cytochrome P450s. *Biotechnology and Applied Biochemistry*, **60**, 134–145. <https://doi.org/10.1002/bab.1074>.
- [25] Urban,P., Lautier,T., Pompon,D., and Truan,G. (2018) Ligand access channels in cytochrome P450 enzymes: A review. *International Journal of Molecular Sciences*, **19**, 1617. <https://doi.org/10.3390/ijms19061617>.
- [26] Midlik,A., Navrátilová,V., Moturu,T.R., Koča,J., Svobodová,R., and Berka,K. (2021) Uncovering of cytochrome P450 anatomy by SecStrAnnotator. *Scientific Reports*, **11**, 12345. <https://doi.org/10.1038/s41598-021-91494-8>.

- [27] Midlik,A., Hutařová Vařeková,I., Hutař,J., Charesheanu,A., Berka,K., and Svobodová,R. (2022) OverProt: secondary structure consensus for protein families. *Bioinformatics*, (in press). <https://doi.org/10.1093/bioinformatics/btac384>.
- [28] Hutařová Vařeková,I., Hutař,J., Midlik,A., Horský,V., Hladká,E., Svobodová,R., and Berka,K. (2021) 2DProts: database of family-wide protein secondary structure diagrams. *Bioinformatics*, **37**, 4599–4601. <https://doi.org/10.1093/bioinformatics/btab505>.
- [29] Berg,J.M., Tymoczko,J.L., and Stryer,L. (2002) *Biochemistry, 5th ed.* W.H. Freeman, New York. ISBN: 9780716730514.
- [30] Lesk,A.M. (2001) *Introduction to protein architecture: the structural biology of proteins.* Oxford University Press, New York. ISBN: 9780198504740.
- [31] Rossmann,M.G. (2013) Super-secondary structure: A historical perspective. In Kister,A.E. (ed.), *Protein Supersecondary Structures.* Humana Press, Totowa, NJ. Vol. 932, pp. 1–4. https://doi.org/10.1007/978-1-62703-065-6_1.
- [32] Pauling,L., Corey,R.B., and Branson,H.R. (1951) The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, **37**, 205–211. <https://doi.org/10.1073/pnas.37.4.205>.
- [33] Pauling,L., and Corey,R.B. (1951) Configurations of polypeptide chains with favored orientations around single bonds: Two new pleated sheets. *Proceedings of the National Academy of Sciences*, **37**, 729–740. <https://doi.org/10.1073/pnas.37.11.729>.
- [34] Ramachandran,G.N., Ramakrishnan,C., and Sasisekharan,V. (1963) Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, **7**, 95–99. [https://doi.org/10.1016/s0022-2836\(63\)80023-6](https://doi.org/10.1016/s0022-2836(63)80023-6).
- [35] Offmann,B., Tyagi,M., and de Brevern,A.G. (2007) Local protein structures. *Current Bioinformatics*, **2**, 165–202. <https://doi.org/10.2174/157489307781662105>.
- [36] Novotny,M., and Kleywegt,G.J. (2005) A survey of left-handed helices in protein structures. *Journal of Molecular Biology*, **347**, 231–241. <https://doi.org/10.1016/j.jmb.2005.01.037>.
- [37] Cao,C., Xu,S., and Wang,L. (2015) An algorithm for protein helix assignment using helix geometry. *PLOS ONE*, **10**, e0129674. <https://doi.org/10.1371/journal.pone.0129674>.

-
- [38] Fodje,M., and Al-Karadaghi,S. (2002) Occurrence, conformational features and amino acid propensities for the π -helix. *Protein Engineering, Design and Selection*, **15**, 353–358. <https://doi.org/10.1093/protein/15.5.353>.
- [39] Kabsch,W., and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637. <https://doi.org/10.1002/bip.360221211>.
- [40] Kumar,P., and Bansal,M. (2015) Dissecting π -helices: sequence, structure and function. *FEBS Journal*, **282**, 4415–4432. <https://doi.org/10.1111/febs.13507>.
- [41] Ehm,T., Shinar,H., Meir,S., Sekhon,A., Sethi,V., Morgan,I.L., Rahamim,G., Saleh,O.A., and Beck,R. (2021) Intrinsically disordered proteins at the nanoscale. *Nano Futures*, **5**, 022501. <https://doi.org/10.1088/2399-1984/abfb7c>.
- [42] Richardson,J.S., Getzoff,E.D., and Richardson,D.C. (1978) The beta bulge: a common small unit of nonrepetitive protein structure. *Proceedings of the National Academy of Sciences*, **75**, 2574–2578. <https://doi.org/10.1073/pnas.75.6.2574>.
- [43] Chan,A.W.E., Hutchinson,E.G., Harris,D., and Thornton,J.M. (1993) Identification, classification, and analysis of beta-bulges in proteins. *Protein Science*, **2**, 1574–1590. <https://doi.org/10.1002/pro.5560021004>.
- [44] Martin,J., Letellier,G., Marin,A., Taly,J.-F., de Brevern,A.G., and Gibrat,J.-F. (2005) Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. *BMC Structural Biology*, **5**, 17. <https://doi.org/10.1186/1472-6807-5-17>.
- [45] Richards,F.M., and Kundrot,C.E. (1988) Identification of structural motifs from protein coordinate data: Secondary structure and first-level supersecondary structure. *Proteins: Structure, Function, and Genetics*, **3**, 71–84. <https://doi.org/10.1002/prot.340030202>.
- [46] Sklenar,H., Etchebest,C., and Lavery,R. (1989) Describing protein structure: A general algorithm yielding complete helicoidal parameters and a unique overall axis. *Proteins: Structure, Function, and Genetics*, **6**, 46–60. <https://doi.org/10.1002/prot.340060105>.
- [47] Labesse,G., Colloc'h,N., Pothier,J., and Mornon,J.-P. (1997) P-SEA: a new efficient assignment of secondary structure from C α trace of proteins. *Bioinformatics*, **13**, 291–295. <https://doi.org/10.1093/bioinformatics/13.3.291>.

- [48] Majumdar,I., Krishna,S.S., and Grishin,N.V. (2005) PALSSE: A program to delineate linear secondary structural elements from protein structures. *BMC Bioinformatics*, **6**, 202. <https://doi.org/10.1186/1471-2105-6-202>.
- [49] Taylor,W.R. (2001) Defining linear segments in protein structure. *Journal of Molecular Biology*, **310**, 1135–1150. <https://doi.org/10.1006/jmbi.2001.4817>.
- [50] King,S.M., and Johnson,W.C. (1999) Assigning secondary structure from protein coordinate data. *Proteins: Structure, Function, and Genetics*, **35**, 313–320. [https://doi.org/10.1002/\(SICI\)1097-0134\(19990515\)35:3<313::AID-PROT5>3.0.CO;2-1](https://doi.org/10.1002/(SICI)1097-0134(19990515)35:3<313::AID-PROT5>3.0.CO;2-1).
- [51] Konagurthu,A.S., Lesk,A.M., and Allison,L. (2012) Minimum message length inference of secondary structure from protein coordinate data. *Bioinformatics*, **28**, i97–i105. <https://doi.org/10.1093/bioinformatics/bts223>.
- [52] Nagy,G., and Oostenbrink,C. (2014) Dihedral-based segment identification and classification of biopolymers I: Proteins. *Journal of Chemical Information and Modeling*, **54**, 266–277. <https://doi.org/10.1021/ci400541d>.
- [53] Kneller,G.R., and Calligari,P. (2006) Efficient characterization of protein secondary structure in terms of screw motions. *Acta Crystallographica Section D Biological Crystallography*, **62**, 302–311. <https://doi.org/10.1107/S0907444905042654>.
- [54] Mitchell,E.M., Artymiuk,P.J., Rice,D.W., and Willett,P. (1990) Use of techniques derived from graph theory to compare secondary structure motifs in proteins. *Journal of Molecular Biology*, **212**, 151–166. [https://doi.org/10.1016/0022-2836\(90\)90312-A](https://doi.org/10.1016/0022-2836(90)90312-A).
- [55] Frishman,D., and Argos,P. (1995) Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Genetics*, **23**, 566–579. <https://doi.org/10.1002/prot.340230412>.
- [56] Dupuis,F., Sadoc,J.-F., and Mornon,J.-P. (2004) Protein secondary structure assignment through Voronoï tessellation. *Proteins: Structure, Function, and Bioinformatics*, **55**, 519–528. <https://doi.org/10.1002/prot.10566>.
- [57] PDBe-KB consortium (2022) PDBe-KB: collaboratively defining the biological context of structural data. *Nucleic Acids Research*, **50**, D534–D542. <https://doi.org/10.1093/nar/gkab988>.

-
- [58] Dana,J.M., Gutmanas,A., Tyagi,N., Qi,G., O'Donovan,C., Martin,M., and Velankar,S. (2019) SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Research*, **47**, D482–D489. <https://doi.org/10.1093/nar/gky1114>.
- [59] Gabanyi,M.J., and Berman,H.M. (2015) Protein structure annotation resources. In Owens,R.J. (ed.), *Structural Proteomics*. Humana Press, New York, NY. Vol. 1261, pp. 3–20. https://doi.org/10.1007/978-1-4939-2230-7_1.
- [60] Krissinel,E., and Henrick,K. (2004) Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D Biological Crystallography*, **60**, 2256–2268. <https://doi.org/10.1107/S0907444904026460>.
- [61] Kocincová,L., Jarešová,M., Byška,J., Parulek,J., Hauser,H., and Kozlíková,B. (2017) Comparative visualization of protein secondary structures. *BMC Bioinformatics*, **18**, 23. <https://doi.org/10.1186/s12859-016-1449-z>.
- [62] Richardson,J.S. (1981) The anatomy and taxonomy of protein structure. In *Advances in Protein Chemistry*. Elsevier. Vol. 34, pp. 167–339. [https://doi.org/10.1016/S0065-3233\(08\)60520-3](https://doi.org/10.1016/S0065-3233(08)60520-3).
- [63] Heinrich,J., Burch,M., and O'Donoghue,S.I. (2014) On the use of 1D, 2D, and 3D visualisation for molecular graphics. In *2014 IEEE VIS International Workshop on 3DVis (3DVis)*, pp. 55–60. <https://doi.org/10.1109/3DVis.2014.7160101>.
- [64] Velankar,S. et al. (2016) PDBe: improved accessibility of macromolecular structure data from PDB and EMDB. *Nucleic Acids Research*, **44**, D385–D395. <https://doi.org/10.1093/nar/gkv1047>.
- [65] Stivala,A., Wybrow,M., Wirth,A., Whisstock,J.C., and Stuckey,P.J. (2011) Automatic generation of protein structure cartoons with Pro-origami. *Bioinformatics*, **27**, 3315–3316. <https://doi.org/10.1093/bioinformatics/btr575>.
- [66] Michalopoulos,I. (2004) TOPS: an enhanced database of protein structural topology. *Nucleic Acids Research*, **32**, 251D–254. <https://doi.org/10.1093/nar/gkh060>.
- [67] Hutchinson,E.G., and Thornton,J.M. (1990) HERA—A program to draw schematic diagrams of protein secondary structures. *Proteins: Structure, Function, and Genetics*, **8**, 203–212. <https://doi.org/10.1002/prot.340080303>.

- [68] Hutchinson,E.G., and Thornton,J.M. (1996) PROMOTIF-A program to identify and analyze structural motifs in proteins. *Protein Science*, **5**, 212–220. <https://doi.org/10.1002/pro.5560050204>.
- [69] Watkins,X., Garcia,L.J., Pundir,S., Martin,M.J., and UniProt Consortium (2017) ProtVista: visualization of protein sequence annotations. *Bioinformatics*, **33**, 2040–2041. <https://doi.org/10.1093/bioinformatics/btx120>.
- [70] Laskowski,R.A., Jabłońska,J., Pravda,L., Vařeková,R.S., and Thornton,J.M. (2018) PDBsum: Structural summaries of PDB entries. *Protein Science*, **27**, 129–134. <https://doi.org/10.1002/pro.3289>.
- [71] Klose,D.P., Wallace,B.A., and Janes,R.W. (2010) 2Struc: the secondary structure server. *Bioinformatics*, **26**, 2624–2625. <https://doi.org/10.1093/bioinformatics/btq480>.
- [72] Schäfer,T., Scheck,A., Bruneß,D., May,P., and Koch,I. (2016) The new protein topology graph library web server. *Bioinformatics*, **32**, 474–476. <https://doi.org/10.1093/bioinformatics/btv574>.
- [73] Kayikci,M., Venkatakrishnan,A.J., Scott-Brown,J., Ravarani,C.N.J., Flock,T., and Babu,M.M. (2018) Visualization and analysis of non-covalent contacts using the Protein Contacts Atlas. *Nature Structural & Molecular Biology*, **25**, 185–194. <https://doi.org/10.1038/s41594-017-0019-z>.
- [74] Orengo,C.A., and Thornton,J.M. (2005) Protein families and their evolution—a structural perspective. *Annual Review of Biochemistry*, **74**, 867–900. <https://doi.org/10.1146/annurev.biochem.74.082803.133029>.
- [75] Orengo,C., Michie,A., Jones,S., Jones,D., Swindells,M., and Thornton,J. (1997) CATH – a hierarchic classification of protein domain structures. *Structure*, **5**, 1093–1109. [https://doi.org/10.1016/S0969-2126\(97\)00260-8](https://doi.org/10.1016/S0969-2126(97)00260-8).
- [76] Coutsias,E.A., Seok,C., and Dill,K.A. (2004) Using quaternions to calculate RMSD. *Journal of Computational Chemistry*, **25**, 1849–1857. <https://doi.org/10.1002/jcc.20110>.
- [77] Zhang,Y., and Skolnick,J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, **57**, 702–710. <https://doi.org/10.1002/prot.20264>.
- [78] Levitt,M., and Chothia,C. (1976) Structural patterns in globular proteins. *Nature*, **261**, 552–558. <https://doi.org/10.1038/261552a0>.

-
- [79] Schaeffer,R.D., and Daggett,V. (2011) Protein folds and protein folding. *Protein Engineering Design and Selection*, **24**, 11–19. <https://doi.org/10.1093/protein/gzq096>.
- [80] Apic,G., Gough,J., and Teichmann,S.A. (2001) Domain combinations in archaeal, eubacterial and eukaryotic proteomes. *Journal of Molecular Biology*, **310**, 311–325. <https://doi.org/10.1006/jmbi.2001.4776>.
- [81] Fox,N.K., Brenner,S.E., and Chandonia,J.-M. (2014) SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Research*, **42**, D304–309. <https://doi.org/10.1093/nar/gkt1240>.
- [82] Chandonia,J.-M., Guan,L., Lin,S., Yu,C., Fox,N.K., and Brenner,S.E. (2022) SCOPe: improvements to the structural classification of proteins – extended database to facilitate variant interpretation and machine learning. *Nucleic Acids Research*, **50**, D553–D559. <https://doi.org/10.1093/nar/gkab1054>.
- [83] Andreeva,A., Howorth,D., Chothia,C., Kulesha,E., and Murzin,A.G. (2014) SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Research*, **42**, D310–D314. <https://doi.org/10.1093/nar/gkt1242>.
- [84] Chandonia,J.-M., Hon,G., Walker,N.S., Lo Conte,L., Koehl,P., Levitt,M., and Brenner,S.E. (2004) The ASTRAL Compendium in 2004. *Nucleic Acids Research*, **32**, D189–D192. <https://doi.org/10.1093/nar/gkh034>.
- [85] Mistry,J., Finn,R.D., Eddy,S.R., Bateman,A., and Punta,M. (2013) Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions. *Nucleic Acids Research*, **41**, e121. <https://doi.org/10.1093/nar/gkt263>.
- [86] Das,S., Lee,D., Sillitoe,I., Dawson,N.L., Lees,J.G., and Orengo,C.A. (2015) Functional classification of CATH superfamilies: a domain-based approach for protein function annotation. *Bioinformatics*, **31**, 3460–3467. <https://doi.org/10.1093/bioinformatics/btv398>.
- [87] Cheng,H., Schaeffer,R.D., Liao,Y., Kinch,L.N., Pei,J., Shi,S., Kim,B.-H., and Grishin,N.V. (2014) ECOD: An evolutionary classification of protein domains. *PLoS Computational Biology*, **10**, e1003926. <https://doi.org/10.1371/journal.pcbi.1003926>.
- [88] Eddy,S.R. (2004) What is dynamic programming? *Nature Biotechnology*, **22**, 909–910. <https://doi.org/10.1038/nbt0704-909>.

- [89] Thompson,J.D., Linard,B., Lecompte,O., and Poch,O. (2011) A comprehensive benchmark study of multiple sequence alignment methods: Current challenges and future perspectives. *PLoS ONE*, **6**, e18093. <https://doi.org/10.1371/journal.pone.0018093>.
- [90] Schneider,T.D., and Stephens,R.M. (1990) Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, **18**, 6097–6100. <https://doi.org/10.1093/nar/18.20.6097>.