RESEARCH

Comparative Visualization of Protein Secondary Structures

Lucia Kocincová[1*], Miroslava Jarešová[1], Jan Byška[1,2], Július Parulek[2], Helwig Hauser[2] and Barbora Kozlíková[1]

Abstract

Background: Protein function is determined by many factors, namely by its constitution, spatial arrangement, and dynamic behavior. Studying these factors helps the biochemists and biologists to better understand the protein behavior and to design proteins with modified properties. One of the most common approaches to these studies is to compare the protein structure with other molecules and to reveal similarities and differences in their polypeptide chains.

Results: We support the comparison process by proposing a new visualization technique that bridges the gap between traditionally used 1D and 3D representations. By introducing the information about mutual positions of protein chains into the 1D sequential representation the users are able to observe the spatial differences between the proteins without any occlusion commonly present in 3D view. Our representation is designed to serve namely for comparison of multiple proteins or a set of time steps of molecular dynamics simulation.

Conclusions: The novel representation is demonstrated on two case studies. The first study aims to compare a set of proteins from the family of cytochromes P450 where the position of the secondary structures has a significant impact on the substrate channeling. The second study focuses on the protein flexibility when by comparing a set of time steps our representation helps to reveal the most dynamically changing parts of the protein chain.

Keywords: Molecular Sequence Analysis; Molecular Structural Biology; Computational Proteomics

Background

Studying the structure of proteins has been in the scope of researchers for many decades, namely because of their importance in all living cells. Better understanding of their constitution and behavior helps to understand and control their function and properties.

Protein structure consists of a polypeptide chain of amino acids which is unique for each type of protein. The chain is folded into a spatial conformation which can possess specific patterns, called secondary structures. Among these structures belong so called alpha-helices and beta-sheets. The amino acids forming these secondary structures are hold their shape thanks to weak hydrogen bonds between them. Visual representation of the protein consisting of secondary structures is denoted as cartoon or ribbon (see Figure 1 left). This highly abstracted visualization omits individual atoms of the protein and highlights only the protein backbone represented by the secondary structures. Such a representation is very popular by researchers because of its balanced tradeoff between the level of abstractness and conveying the spatial arrangement of the chain.

When comparing several protein structures, e.g., when searching for similar structures in order to get the information about an unknown protein, there are several existing algorithms for aligning such structures. These algorithms are aligning the whole structures (structure alignment) or are parsing the sequence of amino acids and searching for corresponding patterns (sequence alignment). The results of these alignments are traditionally presented in a form of color-coded one-dimensional sequential information (see Figure 2). Each row represents one protein structure and the user can observe the similarities and differences between protein chains by exploring the columns. Some methods equip the sequence with the information about secondary structures (see Figure 1 right). However, all of them lack the mutual spatial orientation of the secondary structures of the aligned structures. This information is crucial in many cases, namely when exploring the protein inner void space playing a significant role in protein reactivity with other molecules.

*Correspondence: lucia.koc@gmail.com

[1]Masaryk University, Brno, Czech Republic
Full list of author information is available at the end of the article

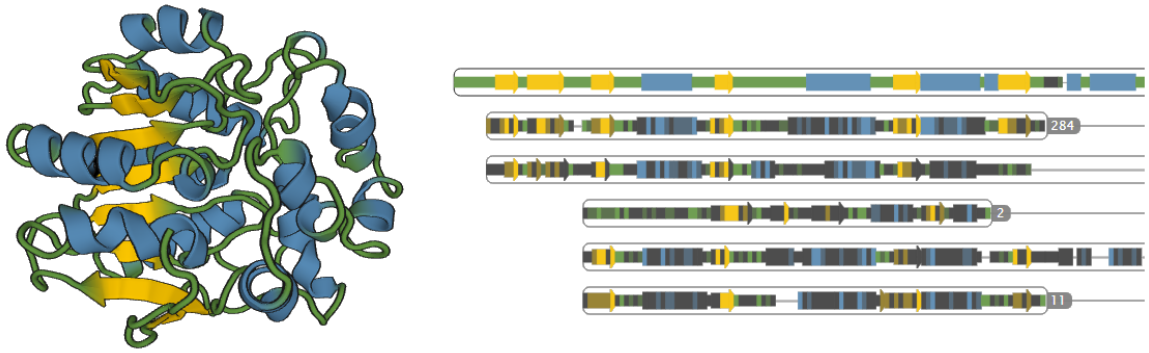

Figure 1 Left – cartoon representation of the DhaA haloalkane dehalogenase (PDB ID 1CQW). Right – part of the sequential representation of DhaA along with the information about secondary structures and several other structures sequentially aligned to DhaA. Images generated using the Aquaria tool by O'Donoghue et al. [1].

This void space is determined by the surrounding amino acids, i.e., secondary structures. Therefore, the changes in the spatial position of the secondary structures directly influence the volume and shape of the void space.

Figure 2 Example of the sequence alignment visualization.

The mutual spatial arrangement of the secondary structures can be easily observed in a 3D view. However, for comparison of multiple proteins, such a representation is very limited with respect to its scalability. In other words, due to the occlusion problems, the spatial representation is suitable for comparison of only few structures. Figure 3 demonstrates the case when four similar proteins are aligned. Even with such a small number of molecules it is hard to perceive the differences in the secondary structure positions.

To overcome the problems of the lack of mutual arrangement of the compared protein in the sequential representation and problems with occlusion in the spatial view, we propose a new method designed to serve as a tool for comparison of multiple structures and intuitive exploration of their spatial differences. It takes the advantages of the sequential information which consists of individual secondary structures and when comparing this sequence with other proteins, it encodes the mutual spatial arrangement of the secondary structures of the aligned proteins. In consequence, the user can observe this arrangement without the occlusion problems of the 3D view. Our solution also utilizes the fact that the domain experts are

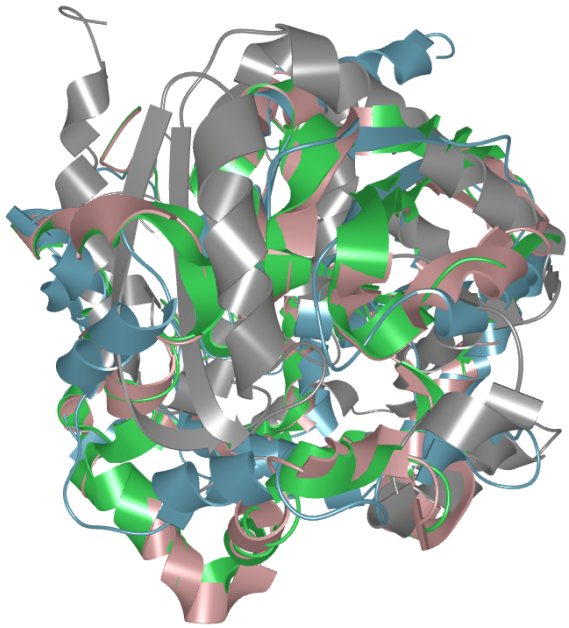

Figure 3 Structure alignment of four proteins with similar structure visualized in 3D.

well accustomed with the sequential representation as well as with secondary structures and their cartoon representation. Therefore, our proposed visualization is interactively linked with the 3D view. Selection of interesting secondary structures in the novel representation is directly projected to this spatial view.

Related Work

[2], maybe also [3]

Methodology

TODO

Design

Strong points of our representation:

- the domain experts are accustomed with the sequential representation as well as with secondary structures
- interactive manipulation
- predefined ranking according to RMSD
- changing superimposition and juxtaposition
- linking with 3D
- automatic highlighting of the most similar and the most different secondary structures
- filtering (removing the most different proteins)
- coloring of SS in the abstracted representation with respect to the physico-chemical properties of the amino acids

Implementation

With the use of the Combinatorial Extension (CE) algorithm for structure alignment [4] and force layout algorithm, we let the structures to align in 3D, so there is no deformation caused by choosing a particular projection or distortion. At the end, we visualize the flattened molecules as if they were stretched out from 3D by pulling the chosen reference molecule at its ends into a straight line, so the actual length of secondary structures is preserved as well as the position of near structures of other molecules which are "locked" to the reference molecule. Algorithm - GAPS - Detail exploration of differences of two molecules

0.0.1 Introduction

We may imagine molecule that contains structures as a nit with colored bubbles on it. Each of these two nits may or may not contain complete set bubbles and if bubbles may not be in same order. Bubbles may be moved along the nit but their order can not change. From protein structure view we are interested where these two nits have not the same order of theirs bubbles or some bubbles are missing. We are searching for gaps in any of these two nits/molecules. Imagine that you put both of nits along each other and you are moving bubbles along the nit such a way to achieve same color of bubbles to be along each other. Best solution would be to minimize amount of gaps, respectively separated bubbles with no pair. Thus the pair making basic algorithm. This require a non trivial recursive algorithm that may be very costly for memory and computational time and its description will be only pseudo-coded here. We will reference this algorithm as Global Best Effort and it's implementation is in stage of optimization of purposes of molecule structure matchmaking. Our hunger approach takes simpler approach but already has promising results

0.0.2 The Hunger Algorithm

Main idea of algorithm is to find possible gaps. Input for the algorithm are two molecules. Output of algorithm are two molecules with added gaps. Algorithm uses distance search and structure comparator. Idea for this algorithm comes from double stack algorithm. Algorithm runs until both molecules are fully traversed and uses step approach. In each step a structure from at least one molecule has to be added to output molecules. If any gap happens progress on input molecules will not be same, but as algorithm progress this may change and so.

Match Making function Each matchmaking from molecule A to molecule B is done only from the current index of progress on each molecule. The molecules are traversed from current index to end of molecule. First match is returned. At the moment only distance in 3D and structure type are considered in matchmaking. In future, penalization for number of possible gaps will be added. Even though this penalization is already implemented in terms of first match. For Global Best Effort algorithm Match making function will have to take responsibility for best match finding.

In each step algorithm decide what to include in output molecules and how to progress on input molecules. Algorithm holds progress in both input molecules.

1 If any molecule is already depleted. Fill output of that molecule with gaps and fill the other molecule with structures that left. This may be imagined as filling the tail.
2 Find best matches from one molecule to another and vice versa, according to the match making function earlier. Take math that has minimum gaps that is needed. Now molecule A will represent the structure which pair has been found in molecule B. Fill the output of molecule A with gaps in ration 1:1 to structures that will be added to output of structure B. Add structure from input A to output of molecule A and same in molecule B. Progress input counter in molecule A by one and in molecule B by 1 + structures added to outout of moleculeB.
3 If no match is returned from both search add gap to molecule A and structure from molecule A to output of molecule A. Add structure from molecule B and gap to molecule B. Progress the counter on both molecules A and B.
4 If both molecules are at end the solution has been found.

0.0.3 Global Best Effort

Motivation: Imagine the previous nit where both of nits have the colors in this order:

A: RED , BLUE , YELLOW

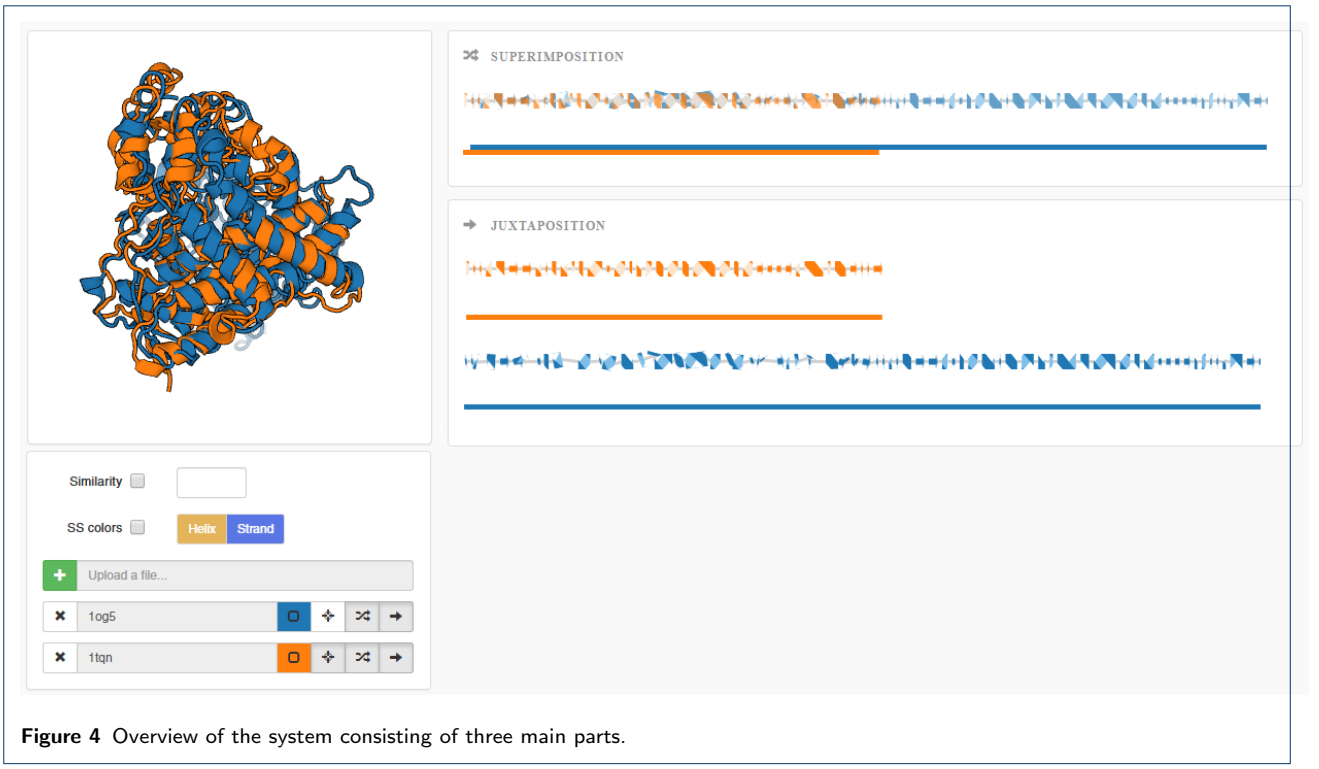

Figure 4 Overview of the system consisting of three main parts.

B: GREEN, BLUE , YELLOW, RED
There are two possibilities for output:
A: RED, GAP , BLUE, YELLOW, GAP
B: GAP, GREEN, BLUE, YELLOW, RED
or
A: GAP , GAP , GAP , RED, BLUE, YELLOW
B: GREEN, BLUE, YELLOW, RED, GAP , GAP

From the point of nit it is simple. The first option of nit it is easy. The first option will by found. In 3D match making situation may not be the same. Imagine the RED structures are very close to each other. The RED in molecule A is very close to RED in molecule B. The matchmaking function will definitely found out that the RED structures are the best match and found out that GREEN has no match in molecule A. From this point the second solution will be taken even if it is not the best estimation in global scale.

Solution: Algorithm already uses recursive calling. It is derived from your Hunger algorithm. Imagine it as a tree that is build depth first with early cut, so dead end computation with already big penalization are cut before additional computation. Algorithm stores all solutions in solutions pool.

1 If penalisation of solution is already too big. Stop this branch and do not continue.
2 If tail of any molecule happens do as hunger algorithm
3 Instead of computing only first match from each molecule, compute all matches. This may sound very demanding but it is required. Do not forget that Match Making function takes distance as a main parameter for match search, thus the possible pairs may be greatly minimised.
4 For each of possible match do operation as would hunger algorithm and continue. This is done via recursion and penalization for amount of gap is passed with each call of function.
5 If no match is found continue as a hunger algorithm.
6 If both molecules are at end continue as a hunger algorithm.

As the algorithm computes depth first solutions are found. Steps 2 and 6 adding solutions to solution pool. The solution pool has to be filtered from time to time. At first solutions with higher penalisation will be added to pool, because the maximum penalisation will not be set. As solutions will be added to pool maximum penalisation will be set to lowest from the pool. In each solution push to pool all solutions in pool with solution higher penalisation than minimal penalistion will be removed from pool. By this only solutions with lowes penalisation will stay in pool.

0.0.4 Dynamics

Typically in dynamics we are comparing more than two molecules at same time. As seen earlier, Hunger and Best Effort algorithm modifies the output molecules. This is not acceptable in global scale as we may have to

compare each molecule with each molecule and there is not a molecule that can not change. Another approach is required. As we only comparing same molecule in different time we are sure that structures will not change positions, even if they may be missing respectively not recognised in that time moment. Algorithm goes through all molecule time snapshots at each time. Algorithm takes all snapshots and returns one super snapshot without gaps but with all structures in order from beginning found in all snapshots. Algorithm holds position for each molecule snapshot.

If molecules are fully traversed end the solution. For each snapshot (snapshot A): Take current structure (structure A) and found that structure on each other snapshot (snapshot X). Traverse only nodes that are has not been already passed on traversed each structure (snapshot X). If GAP would be required. Do not search further and take another snapshot to take structure to find. Note that that in reality match will be found usually within 5 steps on chain so it is very rare to traverse whole molecule. (Time reduction) If structures is found that do not require GAP (structure is missing in other snapshots or it is in first position to take) put it into an output super snapshot and progress on each structure where the structure has been found. If for all snapshots GAP would be required to put (each snapshot would have to produce different structure on same position ... very unexpected ... maybe even impossible) take all snapshots and put all stuctures into super snapshot and update counter on each snapshot.

When this part of algorithm is ended. Run Hunger Algorithm with updated Match Making function. Match Making function will only take in count the structure type and not distance. From the experience we presume that the distance of the structures will not change that much to count with it. There is no need for Best Effort algorithm because the snapshots will be compared with super snapshot that has to encapsulate all snapshots. The super snapshot will no change so input of super snapshot will be the same as output for super snapshot from Hunger algorithm.

On the final visuallisation gaps represent where are the dynamics points of interest. If no gaps are found the threshold for structure recognision may be too big and if many gaps are found the molecule may be instable or the threshold is set very strictly.

Interaction

The 3D view and 2D view are both interactive – basic information about the secondary structure is shown when mouse is moving over the visualizations and a structure is highlighted in green when a structure is selected by a mouse click. Moreover, the 3D and 2D views are interconnected, thus creating a unique way to explore the molecules – when any of the structure is clicked on, the highlight is visible immediately in both views. This feature gives the user very important context of the actual spatial positions of the selected structures and enables him or her to interact with both views independtly, yet still in context. Therefore, the user does not have to tediously decode the flattened visualization trying to figure out what the spatial context of a given structure is, he or she just has to glipms into the 3D view and then continue to the next interesting area for exploration.

The visibility of molecules and their attributes in both views is possible via buttons in configuration panel below the 3D viewer. Molecules in the juxtaposition view are sortable, so the user can see the differences between the pairs of molecules of his or her choice.

Results and Discussion

TODO

Case Study

Conclusions

TODO

Future work - algorithm for generating gaps – design and implement non-trivial approach to visualizing gaps in superimposition of more than two structures - introduce a similarity index which would be created with the cooperation of domain experts - automatic suggestions of interesting parts of the aligned chains – and highlight them in both 2D and 3D - contour based visualization of many superimposed structures - encode additional information into the visualization, e.g. tunnels, ligands (see what is already done in STAR)

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Masaryk University, Brno, Czech Republic. [2]University of Bergen, Norway.

References

1. O'Donoghue, S.I., Sabir, K.S., Kalemanov, M., Stolte, C., Wellmann, B., Ho, V., Roos, M., Perdigao, N., Buske, F.A., Heinrich, J., Rost, B., Schafferhans, A.: Aquaria: simplifying discovery and insight from protein structures. Nat Meth **12**(2), 98–99 (2015)
2. Stolte, C., Sabir, K.S., Heinrich, J., Hammang, C.J., Schafferhans, A., O'Donoghue, S.I.: Integrated visual analysis of protein structures, sequences, and feature data. BMC Bioinformatics **16 Suppl 11**, 7 (2015)
3. Nguyen, K.T., Ropinski, T.: Large-scale multiple sequence alignment visualization through gradient vector flow analysis. In: Biological Data Visualization (BioVis), 2013 IEEE Symposium On, pp. 9–16 (2013)
4. Shindyalov, I.N., Bourne, P.E.: Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. Protein Engineering **11**(9), 739–747 (1998)