



Biological Assembly Comparison with VAST+

Thomas Madej, Aron Marchler-Bauer, Christopher Lanczycki,
Dachuan Zhang, and Stephen H. Bryant

Abstract

The VAST+ algorithm is an efficient, simple, and elegant solution to the problem of comparing the atomic structures of biological assemblies. Given two protein assemblies, it takes as input all the pairwise structural alignments of the component proteins. It then clusters the rotation matrices from the pairwise superpositions, with the clusters corresponding to subsets of the two assemblies that may be aligned and well superposed. It uses the Vector Alignment Search Tool (VAST) protein–protein comparison method for the input structural alignments, but other methods could be used, as well. From a chosen cluster, an “original” alignment for the assembly may be defined by simply combining the relevant input alignments. However, it is often useful to reduce/trim the original alignment, using a Monte Carlo refinement algorithm, which allows biologically relevant conformational differences to be more readily detected and observed. The method is easily extended to include RNA or DNA molecules. VAST+ results may be accessed via the URL <https://www.ncbi.nlm.nih.gov/Structure>, then entering a PDB accession or terms in the search box, and using the link [VAST+] in the upper right corner of the Structure Summary page.

Key words Protein complex, Molecular assembly, Structure comparison, Structure alignment, Secondary structure element

1 Introduction

Quite some years ago, in the earlier days of structural bioinformatics, among the hot topics were the prediction of protein three-dimensional (3D) structure, exploration of the protein fold universe, and understanding the molecular evolution of proteins. For all these studies, the recognition of distant homologues or analogous folds via protein-to-protein structural similarity is necessary. Efficient and useful protein-to-protein structure comparison methods were developed, such as DALI, SSAP, Vector Alignment Search Tool (VAST), CE, MATRAS, TM-align, and others [1–6]. Of course, it was also clear that the comparison of biological assemblies of proteins (and including other molecule types) to one another is important, although the range of complexity of available biological assemblies was limited in the early days of the Protein Data Bank

(PDB). Comparison of assemblies is necessary for the study of molecular interactions and interfaces in atomic-level detail. Good computing performance is highly desirable, as the public structure database PDB right now contains nearly 150,000 structures and continues to grow.

Although it may seem that the comparison of biological molecular assemblies is not much more complicated than the comparison of two molecules, there is an extra degree of complexity introduced because assemblies can be very large, including several thousand or more residues/nucleotides, much larger than size of typical protein–protein comparisons. The fact that the molecules cannot be ordered in any canonical way is another complication. The MM-align algorithm (MultiMer-align [7]) solves these problems by using dynamic programming to achieve good performance and handles the comparison of assemblies by essentially comparing many pairs of large individual “artificial” molecules. The ordering problem is taken care of by considering every possible order of the protein chains, which results in many large individual and “artificial” proteins. The SCPC (Structural Comparison of Protein Complexes [8]) method detects similarities between substructures using secondary structure elements (SSEs). The individual protein chains are compared, using the SSE decompositions. This gives a collection of similar pairs, which are then further agglomerated into larger similar substructures, using a scoring function that cross-checks SSE positions across the constituent pairs. There is also 3D Complex, which represents assemblies by graphs and uses a graph matching algorithm to assess similarity between assemblies [9], producing a hierarchical classification of complexes.

In this chapter, we describe VAST+, a biological assembly comparison algorithm [10] that uses our original VAST [3] protein–protein comparison method. As will be apparent, there is no strict dependence on the VAST algorithm per se; any other pairwise-molecule structure comparison method could be used to provide the needed input to the VAST+ program. Besides the identification of similarities between biological assemblies, we have also made an effort to detect and emphasize meaningful *dissimilarities*, as described in the methods section about “refined alignment.” By considering dissimilarities between assemblies, one can more easily visualize state transitions and important conformational changes. In general, there will be many dissimilarities between assemblies that differ in many details, and automatically detecting and annotating those that may be biologically significant is a major challenge.

2 Methods

2.1 Clustering by Rotation Matrix Distance

Here is the intuition behind the method. In Fig. 1, there are two “assemblies,” ABC (blue) and a similar copy ABC (red). If we do a superposition of the A’s only, then we translate the center of mass (centroid) of the red-A to the centroid of the blue-A and rotate, R1. The amount of rotation is given by the angle between two lines: one line is determined by the centroid of the blue-A and the centroid of the blue assembly, c_1 ; the other line goes through the centroid of the red-A and the centroid of the red assembly, c_2 .

To superpose the two assemblies, i.e., red-ABC and blue-ABC simultaneously, we translate c_2 to c_1 and then rotate (R2); again, the degree of rotation is determined by the angle between the translated lines. A translation does not change the angle between lines, and therefore we see that R1 must be equal to R2.

Of course, this is harder to visualize in 3D, but the same argument applies. The three superpositions of red-A and blue-A, red-B and blue-B, and red-C and blue-C represent the comparisons between single protein chains. The superposition of red-ABC to blue-ABC corresponds to the comparison of two molecular assemblies.

There is only one problem that arises, which is shown in Fig. 2. In this situation, we see that the rotation matrix for the A’s, after translation, is the identity matrix and likewise for the B’s. But the “interface” between A and B in the two assemblies is different! To handle this inconvenience, we simply introduce another numerical restriction, the “orientation check.” Namely, in order to cluster $\text{rot}(A, A')$ and $\text{rot}(B, B')$ together, not only do we require that the Euclidean distance between $\text{rot}(A, A')$ and $\text{rot}(B, B')$ be small enough, but also the vector from the centroid of A to the centroid of B needs to be in roughly the same direction as the corresponding

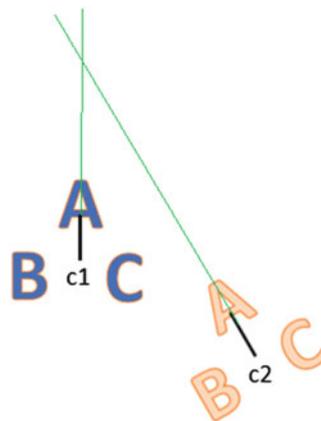


Fig. 1 The angle between the lines gives the rotation required to superpose both the red-A and blue-A and also the entire “assemblies” red-ABC and blue-ABC

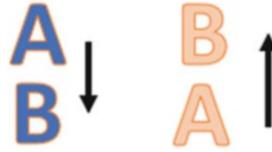


Fig. 2 The rotations to superpose red-A with blue-A and red-B with blue-B are both the identity rotation, but red-AB and blue-AB cannot be superposed together because of a difference in orientation

vector between A' and B' , after the translation and rotation. In the example in Fig. 2, these vectors point in opposite directions.

To determine a reasonable clustering threshold, we took random pairs of arbitrarily chosen rotation matrices obtained from structure comparisons and calculated the Euclidean distance between them. The mean distance was about 2.4, and only about 1.7% of the pairs had a distance less than 1.0. We chose 1.0 as a threshold; this is the simple Euclidean distance in nine dimensions and there are no units.

In outline, the algorithm is thus as follows. Given two protein assemblies to compare, first compute all pairwise structural alignments between the component protein chains. For each pair of similar proteins, there is a rotation matrix for the superposition. The rotation matrices for the pairs are also going to include, with minor deviations, the rotations that superpose entire similar sub-assemblies, by the preceding argument. Cluster the rotation matrices using complete-linkage clustering, with Euclidean distance between the matrices (i.e., distance < 1.0), and using the “orientation check.” The resulting clusters correspond to mappings between protein chains in one assembly and protein chains in the other assembly. An overall alignment for the assemblies can be obtained by simply combining the pairwise structural alignments that we started with for the component chains. In the next section, we will refer to this as an “original alignment” of the assemblies. We can get a superposition of the assemblies via the original alignment. Each of the clusters gives a different alignment; from among these various possibilities, choose one according to whatever desirable criteria. It makes the most sense to consider the clusters with the most protein chains aligned (i.e., the largest clusters), and then further select among these, e.g., by largest total number of residues aligned or smallest root mean square deviation (RMSD; all RMSDs referred to are superposition RMSDs, i.e., obtained after optimal 3D superposition).

Consider the case of comparing two hemoglobin tetramers. Each has four chains, ABCD and $A'B'C'D'$. The chains are all pairwise comparable, so there are rotation matrices $\text{rot}(A, A')$, $\text{rot}(A, B')$, $\text{rot}(A, C')$, etc., which are 16 in all. After the clustering, we get four clusters of size 4, and these correspond exactly to all the

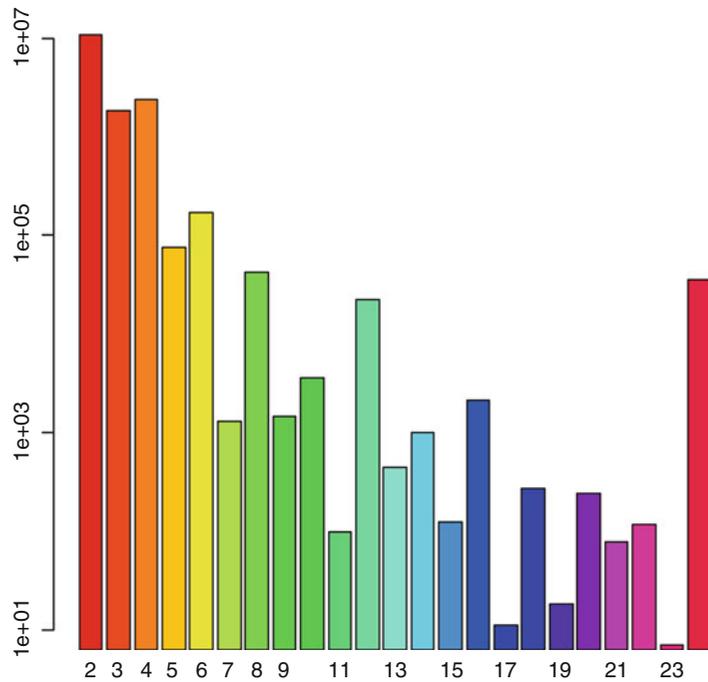


Fig. 3 Counts of similar assemblies in the VAST+ database. The *x*-axis contains the number of molecules in the complexes from dimers to 24-mers. The *y*-axis is on a logarithmic scale, because the dimer-dimer, trimer-trimer, and tetramer-tetramer counts are so dominant

possible alignments of the tetramers. Some of these will map the alpha chains to alpha and beta to beta, and some will map alpha to beta and beta to alpha. The ones mapping alpha to alpha and beta to beta will have slightly larger alignments with a better RMSD, and one will be chosen as the representative assembly alignment, correctly.

If two biological assemblies include nucleotide chains, with structural alignments computed between them, then there is no problem in applying the rotation matrix clustering to include the alignments between the nucleotide chains. Thus, the method is easily extended to more complicated molecular assemblies such as ribosomes.

Figure 3 displays the number of similar assemblies in the VAST+ database, in terms of dimer-dimer, trimer-trimer, etc., on up through 24-mer to 24-mer similarities.

2.2 Refined Alignment

As you can guess, the assembly superposition obtained by using the pairwise alignments/superpositions of the component molecules, i.e., the “original” alignment, amounts to an average superposition. In many, or even most, cases, this is satisfactory, e.g., for closely related structures, the RMSD of the assembly superposition may be

1.0 Å or less, which is better than the resolution of almost all structures in the database. However, in some cases, the average superposition obscures our understanding of the assembly similarity. This will be apparent in the examples given below.

The purpose of the “refined alignment” algorithm is to process the original alignment by trimming it, in order to more easily view any biologically relevant differences between the assemblies. The trimmed pieces, which deviate from a “common core” of the two assemblies, are the positions where the conformations differ to a greater degree. Here is a brief description of the refined alignment algorithm. We formulate it as an optimization problem, where the scoring function is

$$f(S) = N - E$$

Here S is the set of atoms (C-alphas, really paired/aligned atoms) that are included in the current alignment. The current alignment is a subset of the original alignment. Then N is simply the size of S , and E is the error term. We define the error term E as a sum of indexed error terms $E(i, j)$:

$$E = \sum(S) E(i, j)$$

The “ $\sum(S)$ ” means “sum over all positions i, j in S ,” and $E(i, j) = \Delta(i, j) - T + 1$ if $\Delta(i, j) > T$, where T is the tolerance. The deltas are given by

$$\Delta(i, j) = |d(i, j) - d(i', j')|$$

where $d(i, j)$ = (Euclidean) distance between atoms i and j in structure 1, $d(i', j')$ = distance between i' and j' in structure 2, and i is aligned with i' , j with j' . The deltas are measuring the difference in distance between equivalent pairs of positions in the two structures. It works well to choose the tolerance T to be the RMSD of the original alignment/superposition. If we have a current alignment S and add a residue position to it, then N will increase by 1, but the error term E will also increase, depending on how many deltas associated with the position are bigger than the original RMSD and also by how much. We can see that there is going to be a tendency to shrink the original alignment by removing “erroneous” residues.

The scoring function $f(S) = N - E$ is like those appearing in the classical, hard optimization problems. When trying to maximize it, there is a tension between choosing atoms to add to S to increase N and choosing to increase the error term, E . If we set the error term, $E(i, j) = \text{infinity}$ whenever $\Delta(i, j) > T$, then the problem amounts to finding a maximum independent set (MIS) on a type of graph embedded in three dimensions. The MIS on general graphs is a classical NP-complete problem, and there is no known efficient algorithm to solve it. It is not at all obvious if the MIS for general

graphs can be transformed to the problem involving our special class of graphs, so it is unclear whether our problem is NP-complete or not. Nonetheless, a reasonable approach to solving our problem is to use a heuristic. We use a Monte Carlo-type algorithm, a Gibbs sampler. To further simplify the problem, we distinguish the SSEs belonging to the structures. Conveniently, VAST computes the protein-to-protein alignments using SSEs. Corresponding to each aligned SSE, there is a contiguous segment associated with it, in each of the two structures, which may extend past the endpoints of the SSE and into adjacent loop regions. A “move” in the MC algorithm consists of replacing a segment by a different one. When we replace a segment in one structure, we also replace the corresponding segment in the other structure, so that the replacement is consistent with the original alignment. Notice that we can delete a segment simply by replacing it with the empty segment. The error term for replacing a segment is simply the sum of the error terms for the positions that are added, minus the sum of the error terms for the positions removed. Then the Gibbs sampler proceeds in the standard way: choose a segment at random, calculate the scores for all the possible replacements of that segment, assign each possible replacement the Boltzmann weight, and then choose the replacement segment probabilistically according to the weights [11].

3 Examples

An excellent example is provided by comparison between the R (relaxed) and T (tense) states of aspartate transcarbamoylase, e.g., PDB structures 4kh1 and 1rae. 4kh1 is the R-state structure of an ATCase from *E. coli* [12], whereas 1rae is a T-state CTP-ligated ATCase also from *E. coli* [13]. The ATCase assembly consists of two catalytic trimers and three regulatory dimers, for a total of 12 protein chains [14]. Comparing the individual protein chains pairwise, between the 4kh1 and 1rae structures, we see that the pairs with similar folds align very well and superpose with excellent RMSDs of under 1.5 Å.

By clustering the rotation matrices and simply combining all the alignments of the 12 component protein chains, we get an original alignment of the entire assembly involving 2613 residues with a 5.4 Å RMSD. Structure 3D graphical viewers such as iCn3D <https://github.com/ncbi/icn3d> may have an “alternate structure” feature to flip between the structures in the superposed state. When viewed in iCn3D at a good angle, by flipping between the structures, the relative motion of the more rigid halves of the assemblies is readily seen.

However, the refined alignment makes the differences between the T- and R-states even more apparent. For example, in hand,

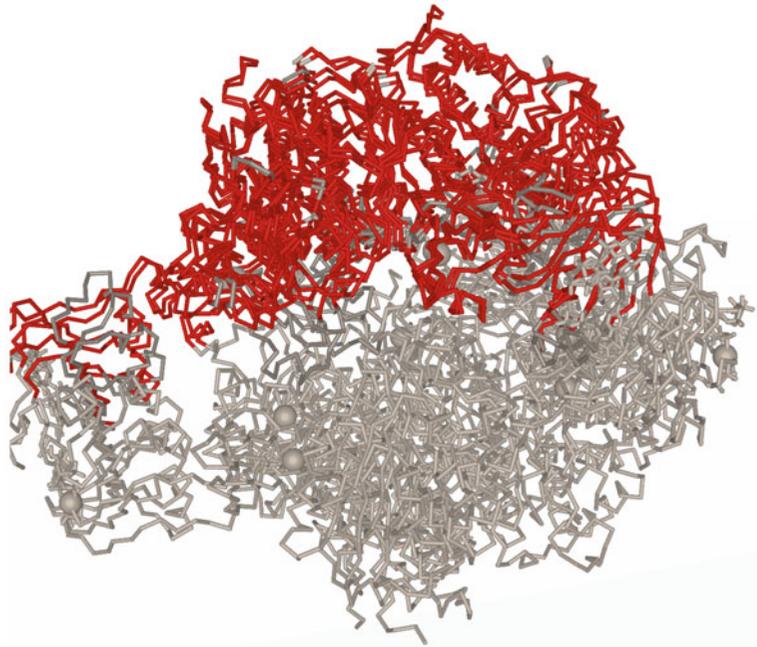


Fig. 4 Refined alignment/superposition of aspartate transcarbamoylase T- and R-state structures (PDB accessions 1rae and 4kh1). The red part displays the six-chain subcomponents that are superposed, which includes one catalytic trimer and one chain from each of the three regulatory dimers

4kh1 and 1rae, the refined alignment has 1113 aligned residues at a RMSD of 2.3 Å. The most striking difference between the T-state and the R-state structures is an expansion of about 11 Å along the threefold axis [14]. The refined alignment redefines the superposition relative to one of the catalytic trimers and includes only six chains, namely, a catalytic trimer and one chain from each of the three regulatory dimers. This helps to make the large-scale difference between the T- and R-states more obvious (*see* Fig. 4). When displayed using the iCn3D graphical viewer, alternating the structures with the “a” key now shows the lower part moving vertically relative to the fixed upper part. *See* Fig. 5 for an overview of how to access the VAST+ webserver using this particular example.

The refinement algorithm can also detect much finer-grained conformational differences. The hemagglutinin influenza virus example involves PDB structures 3sdy and 1mqm. The 3sdy structure is a broadly neutralizing antibody bound to the influenza A H3 hemagglutinin [15]. The 1mqm structure is the hemagglutinin for a potential avian progenitor of the 1968 Hong Kong pandemic influenza virus [16]. All of the protein-to-protein superpositions between the hemagglutinin molecules are excellent, complete chain alignments at under 1.0 Å RMSD. When we combine all these to get our original alignment, we have 1470 residues aligned with an



VAST+ Similar Structures

3D structural similarities among biological assemblies

VAST+ is a tool designed to identify macromolecules that have similar 3-dimensional structures, with an emphasis on finding those with similar biological assemblies ("biological units" or "biounits"). The similarities are calculated using purely geometric criteria, and therefore can identify distant homologs that cannot be recognized by sequence comparison.

Input a valid PDB ID or MMDB ID

To use VAST+, enter the PDB ID or MMDB ID of any structure that is currently in the Molecular Modeling Database (MMDB). VAST+ will display a list of similar structures, ranking them by the extent of their similarity to the query structure's biological unit. [more...](#)

Citing VAST

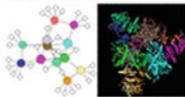
Gibrat JF, Madej T, Br
Madej T, Lanczycki C:
similarities between ma



VAST+ Similar Structures

3D structural similarities among biological assemblies

4KH1: The R State Structure Of E. Coli Atcase With Ctp,utp, And Magnesium Bound



Biological unit 1: dodecameric
Source organism: *Escherichia coli* KO11FL
Number of proteins: 12 (ASPARTATE CARBAMOYLTRANSFERASE, ASPARTATE CARBA... ▼)
Number of chemicals: 30 (Magnesium Ion (3),Zinc Ion (6),Cytidine-5'-Trip... ▼)

Similar Structures (1591)

[Original VAST](#) [Download VAST+](#)

[Invariant substructure superposed](#) [All matching molecules superposed](#)

Display filters

Found 1 structure Search within results:

PDB ID	Description	Taxonomy	Aligned Protein	RMSD	Aligned Residues	Sequence Identity
1 <input type="radio"/> 1RAE	CRYSTAL STRUCTURE OF CTP-LIGATED T STATE ASPARTATE TRANSCARBAMOYLASE AT 2.5 ANGSTROMS RESOLUTION: IMPLICATIONS FOR ATCase MUTANTS AND THE MECHANISM OF NEGATIVE COOPERATIVITY	<i>Escherichia coli</i> K-12	6	2.34Å	1113	100%

Query structure **4KH1**

- 1 ASPARTATE CARBAMOYLTRANSFERASE 271(310)
- 2 ASPARTATE CARBAMOYLTRANSFERASE REGULATORY CHAIN
- 3 ASPARTATE CARBAMOYLTRANSFERASE 266(310)
- 4 ASPARTATE CARBAMOYLTRANSFERASE 271(310)
- 5 ASPARTATE CARBAMOYLTRANSFERASE REGULATORY CHAIN
- 6 ASPARTATE CARBAMOYLTRANSFERASE REGULATORY CHAIN
- 7 ASPARTATE CARBAMOYLTRANSFERASE 0(310)
- 8 ASPARTATE CARBAMOYLTRANSFERASE REGULATORY CHAIN
- 9 ASPARTATE CARBAMOYLTRANSFERASE 0(310)
- 10 ASPARTATE CARBAMOYLTRANSFERASE 0(310)
- 11 ASPARTATE CARBAMOYLTRANSFERASE REGULATORY CHAIN
- 12 ASPARTATE CARBAMOYLTRANSFERASE 0(310)
- 13 ASPARTATE CARBAMOYLTRANSFERASE REGULATORY CHAIN
- 14 ASPARTATE CARBAMOYLTRANSFERASE REGULATORY CHAIN

Matched structure **1RAE**

- 1 Aspartate carbamoyltransferase catalytic chain 271(310)
- 2 Aspartate carbamoyltransferase regulatory chain 99(153)
- 3 Aspartate carbamoyltransferase catalytic chain 266(310)
- 4 Aspartate carbamoyltransferase catalytic chain 271(310)
- 5 Aspartate carbamoyltransferase regulatory chain 100(153)
- 6 Aspartate carbamoyltransferase regulatory chain 106(153)
- 7 Aspartate carbamoyltransferase catalytic chain 0(310)
- 8 Aspartate carbamoyltransferase regulatory chain 0(153)
- 9 Aspartate carbamoyltransferase catalytic chain 0(310)
- 10 Aspartate carbamoyltransferase regulatory chain 0(153)
- 11 Aspartate carbamoyltransferase catalytic chain 0(310)
- 12 Aspartate carbamoyltransferase regulatory chain 0(153)
- 13 Aspartate carbamoyltransferase regulatory chain 0(153)

*Select schematic circles or highlighted molecule names to view matches

Show 1 structures

Fig. 5 From the “VAST+ Similar Structures” webpage, one can enter the PDB accession for a structure of interest (topmost red circle). This goes to the main VAST+ page, which will present a list of similar assemblies in the bottom panel. There is a search box which can be used to filter the results using simple search terms, e.g., keywords appearing in the titles or taxonomy. In this example, we are specifically interested in the PDB accession 1rae and so enter it in the search box (middle red circle). When we expand the entry for that result, we see the list of protein chains from each structure (4kh1 and 1rae), and the ones that correspond in the precomputed superposition are highlighted in the same color. An interactive graphical view can be obtained by launching the iCn3D viewer (bottom red circle)

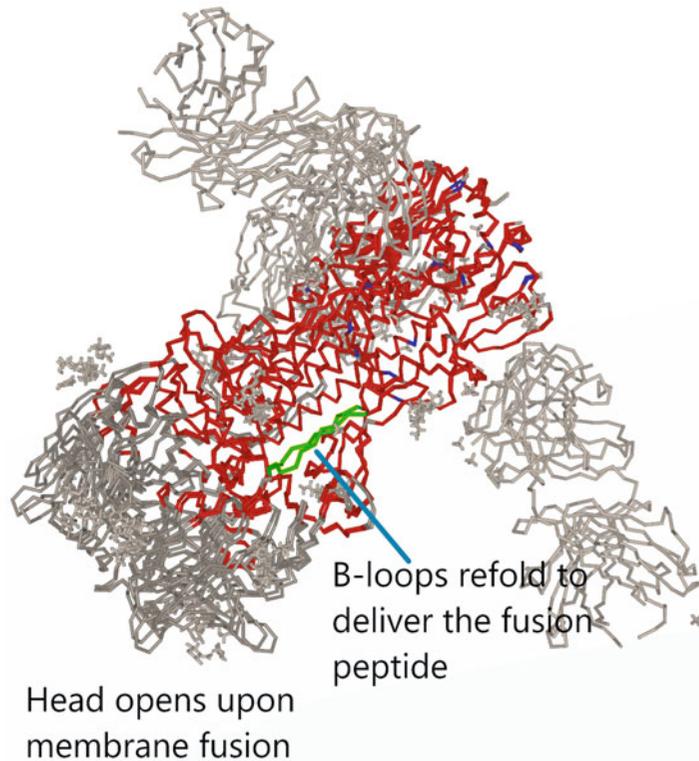


Fig. 6 Refined alignment of influenza virus hemagglutinin assemblies; PDB accessions 3sdy and 1mqm. One of the B-loops is highlighted in green. The head and B-loop annotations were added manually

excellent 1.5 Å RMSD. The refined alignment is trimmed to 865 residues at 0.6 Å RMSD. As in the aspartate transcarbamoylase example, we see that regions involving larger-scale movements become unaligned, i.e., are trimmed (*see* Fig. 6). Again in Fig. 6, the red parts are the aligned and superposed refined alignment, and the white parts are unaligned in the refinement. The head of the hemagglutinin assembly, which is that part which opens upon membrane fusion with the cell the virus is trying to infect, is not included in the refined alignment because there is a conformational difference. Note that the white chain traces at the top and right-hand side, and in the background, are the antibodies in the 3sdy structure, which are not present in 1mqm and hence not included in the original alignment. However, besides the change at the head of the hemagglutinins, finer-grained details are also detected, such as the three B-loops. The B-loops change conformation in order to deliver the fusion peptide [15]. This is a good example, where the protein-to-protein alignments are excellent and align complete chains, but by comparing the assemblies and using the refinement method, we can detect subtle but important dissimilarities that are biologically relevant.

4 Summary

The VAST+ algorithm is an efficient, simple, and elegant solution to the problem of comparing the atomic structures of biological assemblies. It uses the VAST protein–protein comparison method for the underlying structural alignments, but other methods could be used, as well. The original VAST was designed to detect similarity between protein folds and was not very concerned with subtle differences in the structures. By using VAST+ and comparing entire biological assemblies, we can automatically detect important and subtle biologically relevant differences in the structures. Moreover, as examples like the influenza hemagglutinin show, it is only by comparing entire assemblies that we can detect and clarify relationships between the structures.

Acknowledgments

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine at National Institutes of Health/DHHS. Funding for open access charge: Intramural Research Program of the National Library of Medicine, National Institutes of Health.

References

- Holm L, Sander C (1993) Protein structure comparison by alignment of distance matrices. *J Mol Biol* 233(1):123–138
- Orengo CA, Taylor WR (1996) SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol* 266:617–635
- Gibrat JF, Madej T, Bryant SH (1996) Surprising similarities in structure comparison. *Curr Opin Struct Biol* 6(3):377–385
- Shindyalov IN, Bourne PE (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng* 11(9):739–747
- Kawabata T, Nishikawa K (2000) Protein structure comparison using the Markov transition model of evolution. *Proteins* 41(1):108–122
- Zhang Y, Skolnick J (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res* 33(7):2302–2309
- Mukherjee S, Zhang Y (2009) MM-align: a quick algorithm for aligning multiple-chain protein complex structures using iterative dynamic programming. *Nucleic Acids Res* 37(11):e83
- Koike R, Ota M (2012) SCPC: a method to structurally compare protein complexes. *Bioinformatics* 28(3):324–330
- Levy ED, Pereira-Leal JB, Chothia C, Teichmann SA (2006) 3D complex: a structural classification of protein complexes. *PLoS Comput Biol* 2(11):e155
- Madej T, Lanczycki CJ, Zhang D, Thiessen PA, Geer RC, Marchler-Bauer A, Bryant SH (2014) MMDB and VAST+: tracking structural similarities between macromolecular complexes. *Nucleic Acids Res* 42(Database issue):D297–D303
- Tanner MA (1998) Tools for statistical inference: methods for the exploration of posterior distributions and likelihood functions, Springer series in statistics. Springer-Verlag, New York
- Cockrell GM, Zheng Y, Guo W, Peterson AW, Truong JK, Kantrowitz ER (2013) New paradigm for allosteric regulation of *Escherichia coli* aspartate transcarbamoylase. *Biochemistry* 52(45):8036–8047

13. Kosman RP, Gouaux JE, Lipscomb WN (1993) Crystal structure of CTP-ligated T state aspartate transcarbamoylase at 2.5 Å resolution: implications for ATCase mutants and the mechanism of negative cooperativity. *Proteins* 15(2):147–176
14. Lipscomb WN, Kantrowitz ER (2012) Structure and mechanisms of *Escherichia coli* aspartate transcarbamoylase. *Acc Chem Res* 45(3):444–453
15. Ekiert DC, Friesen RH, Bhabha G, Kwaks T, Jongeneelen M, Yu W, Ophorst C, Cox CF, Korse HJ, Brandenburg B, Vogels R, Brakehoff JP, Kompier R, Koldijk MH, Cornelissen LA, Poon LL, Peiris M, Koudstaal W, Wilson IA, Goudsmit J (2011) A highly conserved neutralizing epitope on group 2 influenza A viruses. *Science* 333(6044):843–850
16. Ha Y, Stevens DJ, Skehel JJ, Wiley DC (2003) X-ray structure of the hemagglutinin of a potential H3 avian progenitor of the 1968 Hong Kong pandemic influenza virus. *Virology* 309(2):209–218

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), 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 license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license 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.

