Special Issue
"Progress of Theoretical and Computational Biophysics"

Vol. 16, pp. 328–336 (2019)
doi: 10.2142/biophysico.16.0_328

Biophysics and Physicobiology
https://www.jstage.jst.go.jp/browse/biophysico/

Regular Article

Methodological improvements for the analysis of domain movements in large biomolecular complexes

Ruth Veevers and Steven Hayward

Computational Biology Laboratory, School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK

Received July 1, 2019; accepted August 3, 2019

Domain movements play a prominent role in the function of many biomolecules such as the ribosome and F_0F_1-ATP synthase. As more structures of large biomolecules in different functional states become available as experimental techniques for structure determination advance, there is a need to develop methods to understand the conformational changes that occur. DynDom and DynDom3D were developed to analyse two structures of a biomolecule for domain movements. They both used an original method for domain recognition based on clustering of "rotation vectors". Here we introduce significant improvements in both the methodology and implementation of a tool for the analysis of domain movements in large multimeric biomolecules. The main improvement is in the recognition of domains by using all six degrees of freedom required to describe the movement of a rigid body. This is achieved by way of Chasles' theorem in which a rigid-body movement can be described as a screw movement about a unique axis. Thus clustering now includes, in addition to rotation vector data, screw-axis location data and axial climb data. This improves both the sensitivity of domain recognition and performance. A further improvement is the recognition and annotation of interdomain bending regions, something not done for multimeric biomolecules in DynDom3D. This is significant as it is these regions that collectively control the domain movement. The new stand-alone, platform-independent implementation, DynDom6D, can analyse biomolecules comprising protein, DNA and RNA, and employs an alignment method to automatically achieve the required equivalence of atoms in the two structures.

Key words: hinge bending, hinge axis, molecular machine, allosteric mechanism

Biomolecules often comprise more than one folded chain, called subunits. Many of these multi-subunit biomolecules appear to have complex mechanisms, and early biochemical studies revealed an allosteric mechanism to be operating even in relatively small proteins such as the tetrameric haemoglobin. Larger biomolecules appear to have machine-like mechanisms. In aspartate transcarbamoylase (ATCase) [1], a 12-subunit enzyme comprising two catalytic trimers and three regulatory dimers, negative-feedback regulation occurs through a domain movement in the regulatory subunits induced by the binding of CTP, an end product of the biosynthetic pathway, which acts to cause conformational change within the catalytic trimers that switches off the catalytic process [2]. In F_0F_1-ATP synthase [3,4], ATP is

Corresponding author: Steven Hayward, Computational Biology Laboratory, School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK.
e-mail: sjh@cmp.uea.ac.uk

◀ **Significance** ▶

Biomolecules exhibit complex shape changing movements during function that are difficult to interpret at the atomic level of detail. The DynDom methodology attempts to describe conformational change in terms of relative movements of domains as quasi-rigid bodies. Here we extend the DynDom methodology by incorporating all six degrees of freedom that describe rigid body movements into the domain identification process. The new tool, DynDom6D, also determines interdomain bending regions. DynDom6D is a platform-independent implementation which can analyse biomolecules comprising protein, DNA and RNA, and employs an alignment method to automatically achieve the required equivalence of atoms in the two structures.

Special issue
Progress of theoretical and computational biophysics

synthesised from ADP in a mechanism that involves domain movements in the β subunits in F_1 being driven through rotation of the central γ-subunit which acts as a rotor shaft connecting the F_0 motor to F_1. In the 70S ribosome a ratchet-like movement between the 30S and 50S subunits occurs during mRNA and tRNA translocation [5]. These examples demonstrate how large multi-subunit biomolecules have moving parts and function very much like the machines we have built in our own macroscopic world.

At this current time cryo-electron microscopy (cryo-EM) is emerging as a dominant method to determine structures of very large biomolecular complexes at atomic or near atomic resolution. Particularly promising is single-particle cryo-EM where a heterogeneous ensemble of structures at different states in the functional process is analysed to classify individual molecules according to conformation allowing determination of the structural changes that occur upon changes in state. This method has been applied to the ribosome by Loveland, A. B., *et al.* [6] to determine multiple structures in the process of codon recognition. These latest advances suggest that there will soon be a dramatic increase in the number of large structures deposited at the Protein Data Bank (PDB) that reveal the conformational changes that occur within a biomolecule as it goes through its functional cycle. Thus there will be an increasing need for methods to be developed to analyse these conformational changes.

If only one structure is available from experiment, simulation methods such as normal mode analysis [7] can be used to find possible alternative conformations which can then be analysed in order to understand the nature of the movements that are inherent to the biomolecule concerned [8–10].

It is often the case that conformational changes involve domain movements. Although the term "domain" can have different definitions within biochemistry, a particularly useful one in the context of conformational change, is a region that moves as a quasi-rigid body [11]. These so-called, "dynamic domains", may or may not correspond to other definitions, but if one is interested in conformational change, identification of these regions and the description of their relative movements should provide insight into mechanism. Such a description is coarse-grained allowing one to focus on the relevant movements of large parts of the biomolecule avoiding the overly detailed atomic-level description.

There have been a number of different approaches to determining domains. An early but limited approach used distance-difference maps [12]. A more promising approach using distance-difference maps is the Motion Tree method [13], which finds a hierarchy of "domains". An alternative to using distance-difference maps is to recognise domains from the difference in their rotational properties [11,14,15] or to search for regions that do not deform appreciably during the movement [16]. There is a similarity between the aims of these methods and those methods that attempt to superpose two structures allowing regions of the protein to flex [17,18].

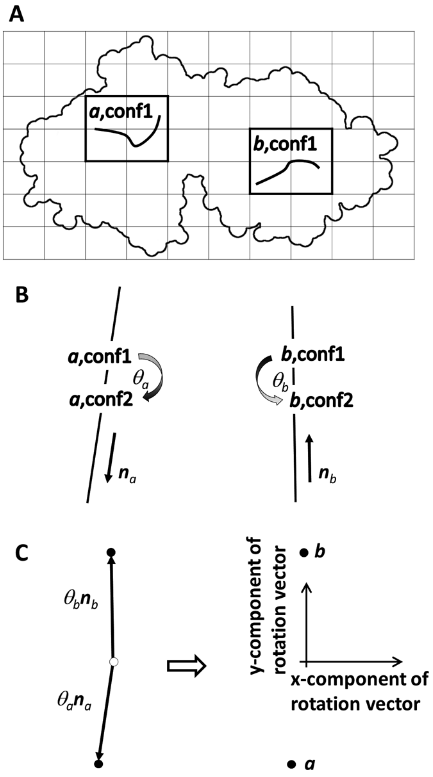


Figure 1 Schematic for the process of dynamic domain determination. (A) In DynDom main-chain segments (short curved lines) are selected by a sliding window. In DynDom3D and DynDom6D regions are selected by blocks (large bold squares) that move on a grid spanning the whole biomolecule. Indicated are two segments/blocks *a* and *b* within two different regions of the biomolecule which have a different rigid body movement in going from conformation 1 to conformation 2. (B) The rotations of atoms within the segments/blocks are analysed. Segment/block *a* rotates by an angle of θ_a about an axis that has a direction given by the unit vector $\boldsymbol{n}_a$ and segment/block *b* rotates by an angle of θ_b about an axis that has a direction given by the unit vector $\boldsymbol{n}_b$. (C) Left: Rotation vectors for segments/blocks *a* and *b*. Right: End points of the rotation vectors are indicated in a "rotation space". Neighbouring blocks/segments that have similar rotational properties will have points that cluster in the rotation space indicating dynamic domains within the biomolecule.

The original DynDom method [11] was tailored to analyse individual proteins chains, even if they were part of a larger multimeric molecule. DynDom3D [19] was developed from DynDom [11] in response to the requirement of analysing conformational change within large multimeric biomolecules. DynDom has at its heart a sliding window that selects overlapping main-chain segments and the rotation vector of each segment between the first and second structure is calculated. The components of the rotation vectors are treated as coordinates meaning that each segment has associated with it a point in a 3D "rotation space" (see Fig. 1). Segments from domains that behave as rigid bodies will have co-located points and thus segments from domains that move as quasi-rigid bodies will have clusters of rotation points. These clusters were determined using the *k*-means clustering algorithm. A crucial feature of the method is the overlapping of segments which has a smoothing effect.

DynDom has an obvious limitation when used to analyse domain movements in chains that are part of a multimeric molecules. When applied to an individual main-chain within a multimeric molecule it is not able to analyse rotational transitions between different subunits that do not occur through the main-chain. To overcome this, in DynDom3D, the rotational analysis is performed on atoms within cubic blocks situated at grid points spanning the whole biomolecule [19] (see Fig. 1). Thus in DynDom3D the grid length plays an analogous role to the distance between consecutive windows in DynDom, which is the distance between neighbouring residues in a protein. The block size is determined by an integer "block factor" which is an integer multiple of the grid length. Thus the block factor plays an analogous role to the window length in DynDom. As long as the block factor is greater than 1, the blocks overlap and the results are smoothed in a similar way to DynDom. DynDom3D was shown to be robust against changes in grid-length and block factor and was also able to reproduce the results of DynDom on individual chains.

One issue that concerns the basic methodology underlying both DynDom and DynDom3D is that domains are determined from the clustering of rotation vectors but other quantities, such as location of the axis and translation along the axis, which together contribute a further three parameters—giving the required six to describe the movement of a rigid body—are not used. Thus disconnected regions that happen to rotate identically but about different axes, would be assigned to the same cluster even though they do not form a dynamic domain. To ameliorate this a very slow and memory hungry "connected-set" algorithm [19] was used to determine if the clusters corresponded to set of atoms forming a connected region. The main methodological advance of the work presented here is to use all six parameters in the domain recognition process which means we can avoid the use of the inefficient connected-set algorithm. The approach was already alluded to in the original DynDom paper, where it was stated, "*The search in the three-dimensional rotation space for clusters of rotation vectors can be generalized to the search in the six-dimensional space containing all the six parameters governing the rigid body motion of residues or main-chain segments. Such an implementation will overcome the possible problem caused if domains happen to have identical rotation vectors.*" Hinsen, K., *et al.* [15] did incorporate translational information into their method but here we present a unique method for doing this that is consistent with the overall DynDom approach. This is achieved through the use of Chasles' theorem [20] which is applied to determine the interdomain screw axes describing the relative domain movements in the original programs. We apply this theorem to the recognition of the dynamic domains themselves which means that all six degrees of freedom are used. This should lead to a program that is more sensitive in detecting differences between the rigid-body movements of the constituent parts of the biomolecule and also be considerably more efficient in both time and memory. In contrast to DynDom, DynDom3D did not analyse rotational transitions that are annotated as (hinge-) bending regions. These regions are of critical importance as they combine to control the domain movement [21] as has been clearly demonstrated for glutamine binding protein where the domain movement could be accurately reproduced even though only 11 (out of 226 in total) residues in the two hinge bending regions were allowed to flex [22]. Within the new methodology we also determine and annotate regions between neighbouring domains where a transition between the rigid-body movements occurs. This should lead to new insights into how the relative movement of domains is controlled allowing experts on the biomolecule concerned to focus on these regions.

Here we present a new standalone, platform-independent, implementation which we call "DynDom6D", that has the features described above and which also incorporates a preprocessor to automatically align the atoms within the two structures as currently done at the DynDom3D webserver [23]. In keeping with our original intention and in contrast to the current server and previous standalone program which only work on protein molecules, it also works for biomolecules comprising DNA and RNA molecules.

Methods

Preliminary process

The two structures are superposed using least-squares best fitting and therefore there needs to be a one-to-one correspondence between atoms in the two structures. The method used applies dynamic programming for sequence alignment and has been described in detail elsewhere [23]. It allows the submitted PDB files to contain multiple chains and/or models, as for example, is often the case for "biological units" from the PDB. The DynDom6D implementation also includes the ability to analyse RNA and DNA molecules. Given that some molecules are too large to be stored as PDB-formatted files, the new program is also able to read mmCIF-formatted files.

Sliding block on grid

Given two structures, without any particular preference, one called "first" and the other "second", the second structure is superimposed on the first using an all-atom least-squares best-fit routine. The coordinate system is then changed to that of the principal axes of the first structure and a cubic grid, with cell length, g, is constructed on the first structure. Cubic blocks of length, $b\times g$, where b is the integer block factor, are then placed at each grid point and the atoms within each block in the first structure are fitted to the equivalent set of atoms in the second structure. The fitting routine employs a quaternion-based method from which it is possible to directly determine the rotation vectors, $\theta\boldsymbol{n}$, where θ

is the angle of rotation and $\boldsymbol{n}$ is the unit vector in the direction of the axis of rotation. As long as $b>1$, blocks will overlap which has a smoothing effect on the differences between the block derived quantities described below.

k-means clustering using all parameters from rigid-body movement

Up to this stage the main process is the same as for the original DynDom3D [19]. The key to the new approach presented here is the use of Chasles' theorem [20] to recognise regions of the biomolecule that move as quasi-rigid bodies. Chasles' theorem states that the movement of a rigid body between a start and end position ("position" implicitly meaning orientation as well) can be described by a screw movement about a unique axis. This means for a perfectly rigid body every block within it would have the same angle of rotation, θ, the same axis in terms of both its direction, $\boldsymbol{n}$, and location in space, and the same translation or climb along the axis, h (Fig. 2 illustrates the meaning of these quantities). Thus blocks that span a domain that moves as a *quasi*-rigid body will have *similar* values for θ, h, $\boldsymbol{n}$, and for the quantity that specifies the location of the axis. Rigid bodies undergoing different movements would therefore be distinguishable by the difference in these quantities. In the previous version of DynDom3D and DynDom only $\theta\boldsymbol{n}$ was used for this purpose but here we use all these quantities. In practice, once $\boldsymbol{n}$ has been determined by the method described above, h can be determined and then a point on the screw axis fixing its location in space. The vector $\theta\boldsymbol{n}$ specifies three of the six parameters, h, a further one, but a point on the axis fixing its location provides a further three. This brings the number to seven, one more than strictly needed. Once the direction of the axis has been fixed it would only require a further two to fix its location in space. One obvious way to do this would be to specify the point it crosses a plane. The problem with this is that a set of axes, which may cluster well in 3D space, may spread out dramatically on the selected plane. It would be preferable, therefore, to do the search for clusters of axis locations in the full 3D space. Our solution is to perform clustering on $\boldsymbol{r}_i+\boldsymbol{n}_i\tau$, $\theta_i\boldsymbol{n}_i$, and h_i, combined orthogonally, where $i=1,N$ (N being the total number of blocks) labels the result of the analysis described above on the ith block, $\boldsymbol{r}_i$ denotes a position vector to any point on the straight line that represents the axis, and τ specifies a point on the line in the usual parametric equation of a straight line (see Fig. 2). The $\boldsymbol{r}_i+\boldsymbol{n}_i\tau$ are used for clustering of axes based on their location in space, the $\theta_i\boldsymbol{n}_i$ are used to cluster rotation vectors as in the original DynDom method, and the h_i are used to cluster movements based on translation along the screw axes. Combining these orthogonally, means that they all contribute to the process of clustering and therefore of identifying dynamic domains.

The k-means clustering algorithm is an iterative process that assigns points to the cluster whose mean they are closest to. The value k refers to the number of specified clusters. It requires the calculation of the mean of a set of points and the calculation of the distance between the mean and each point. Given a subset of blocks, S_l, in cluster, l, it is clear how to calculate the means of h_j, and $\theta_j\boldsymbol{n}_j$, where $j\epsilon S_l$, but it is not immediately clear how to calculate a "mean point" that relates to the locations of a set of lines, $\boldsymbol{r}_j+\boldsymbol{n}_j\tau$. An approach is revealed by recalling a fundamental property of the mean; that is the point from which the sum of the squared distances to points in the set, is a minimum. Thus we find a point with position vector, $\boldsymbol{R}_l$, that minimises the sum of the squared *minimum* distances to the lines in the set. That is, for cluster l we would like to minimise:

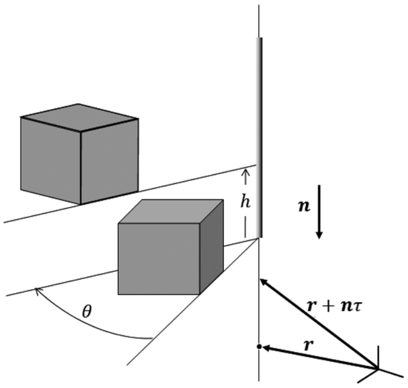


Figure 2 Schematic showing the meaning of the quantities $\boldsymbol{r}+\boldsymbol{n}\tau$, θ, and h in the movement of a rigid body as a screw about an axis. $\boldsymbol{r}$ is the position vector from the origin to a point on the axis, $\boldsymbol{n}$ is the unit vector in the direction of the axis given by the right-hand rule, τ a parameter that determines a point on the line representing the screw axis, θ is the angle of rotation, and h is the climb along the axis.

$$D_l = \sum_{j\epsilon S_l} |(\boldsymbol{r}_j - \boldsymbol{R}_l)\times\boldsymbol{n}_j|^2, \tag{1}$$

with respect to $\boldsymbol{R}_l$, where "×" indicates the vector or cross product. Given a coordinate system, and representing these quantities as column vectors, $\boldsymbol{R}_l=(X_l\ Y_l\ Z_l)^t$, $\boldsymbol{r}_j=(x_j\ y_j\ z_j)^t$ and $\boldsymbol{n}_j=(n_{xj}\ n_{yj}\ n_{zj})^t$, where t denotes the transpose, we can write Eq. (1) in matrix form as:

$$D_l = \boldsymbol{R}_l^t\boldsymbol{N}_l\boldsymbol{R}_l - 2\boldsymbol{R}_l^t\boldsymbol{P}_l + Q_l, \tag{2}$$

where, $\boldsymbol{N}_l=\sum_{j\epsilon S_l}\boldsymbol{\eta}_j$, $\boldsymbol{P}_l=\sum_{j\epsilon S_l}\boldsymbol{\eta}_j\boldsymbol{r}_j$ and $Q_l=\sum_{j\epsilon S_l}\boldsymbol{r}_j^t\boldsymbol{\eta}_j\boldsymbol{r}_j$, with

$$\boldsymbol{\eta}_j = \begin{pmatrix} n_{yj}^2+n_{zj}^2 & -n_{xj}n_{yj} & -n_{xj}n_{zj} \\ -n_{xj}n_{yj} & n_{xj}^2+n_{zj}^2 & -n_{yj}n_{zj} \\ -n_{xj}n_{zj} & -n_{yj}n_{zj} & n_{xj}^2+n_{yj}^2 \end{pmatrix}.$$

$\boldsymbol{N}_l$ is a 3×3 symmetric matrix, $\boldsymbol{P}_l$ a 3×1 matrix and Q_l a scalar. The minimum value for D_l with respect to variation in $\boldsymbol{R}_l$ is found when,

$$\nabla D_l = 2\boldsymbol{N}_l\boldsymbol{R}_l - 2\boldsymbol{P}_l = \boldsymbol{0}, \tag{3}$$

from which we find

$$\boldsymbol{R}_l = \boldsymbol{N}_l^{-1}\boldsymbol{P}_l\,. \tag{4}$$

$\boldsymbol{R}_l$ provides three components of the mean point used for *k*-means clustering.

Denoting the means of h_j and $\theta_j\boldsymbol{n}_j$ in cluster *l*, as h_l and $\theta_l\boldsymbol{n}_l$, respectively, the *k*-means clustering method implemented here aims to minimize the following expression through cluster membership, i.e. by varying S_l, l=1,*k*,

$$\textstyle\sum_{l=1}^{k}\sum_{j\epsilon S_l}(|(\boldsymbol{r}_j-\boldsymbol{R}_l)\times\boldsymbol{n}_j|^2+|\theta_j\boldsymbol{n}_j-\theta_l\boldsymbol{n}_l|^2+|h_j-h_l|^2). \tag{5}$$

This shows the difference between this new version and previous versions, where for both DynDom and the original DynDom3D only the central term, $|\theta_j\boldsymbol{n}_j-\theta_l\boldsymbol{n}_l|^2$, was used for clustering, i.e. for determination of dynamic domains. Here we have three different features contributing to the clustering.

We allow the user to use Eq. (5) directly for *k*-means clustering, but we have also implemented a form of feature scaling before *k*-means is performed. In this scaled version the quantities for clustering, $\boldsymbol{r}_i$, $\theta_i\boldsymbol{n}_i$ and h_i are respectively substituted by the following scaled versions, for i=1,*N*:

$$\boldsymbol{r}'_i=\frac{\boldsymbol{r}_i}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}|\boldsymbol{r}_i|^2}}, \tag{6a}$$

$$\theta'_i\boldsymbol{n}_i=\frac{\theta_i\boldsymbol{n}_i}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\theta_i^2}}, \tag{6b}$$

$$h'_i=\frac{h_i}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}h_i^2}}. \tag{6c}$$

The particular *k*-means algorithm we use starts by assigning all blocks to one single cluster, then two clusters, then three, and so on until it reaches a stopping criterion or the pre-set value of *k*. We refer to these below as "levels" of clustering.

Figure 3 shows a hypothetical but numerically correct 2D example which illustrates how DynDom6D would improve on DynDom3D. In this example the clusters are distinguished by axis locations not by their associated rotation vectors.

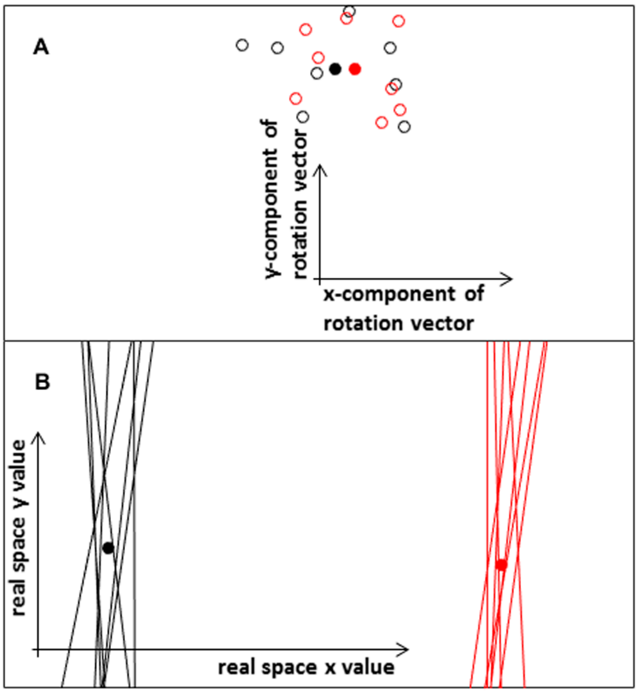


Figure 3 Hypothetical, but numerically correct 2D example that illustrates what might happen if dynamic domains are distinguishable from axis locations but not rotation vectors. To create this example, sixteen axes and rotation vectors in 2D were randomly generated such that the rotation vectors are similar and do not form clear clusters, but axis locations form two distinct clusters. (A) Rotation points (open circles). (B) Lines indicating corresponding axes. The filled circles show the two mean points determined by setting *k*=2 in the *k*-means clustering algorithm performed using the first two terms of Eq. (5). Rotation points and axes belonging to the two different clusters identified are coloured black and red. In (A) the rotation points from the two clusters do not show a clear separation, whereas in (B) the clear separation of the axis locations determines the clusters. Clustering on the rotation points alone would not have produced the same result and the two distinct dynamic domains indicated by the two separate clusters of axes would not have been correctly identified. The mean points in (B) have the minimum sum of the squared minimum distances to the lines of the same colour.

From clusters of blocks to clusters of atoms or residues

Some blocks near the surface of the molecule may have low occupancy; that is they contain very few atoms and are removed as they are likely to contribute noise. They are removed based on a threshold parameter, *occ*. If N_{max} is the number of atoms in the block containing the most atoms, then blocks with fewer atoms than $occ\times N_{max}$ are removed. Blocks are assigned to clusters but we need to assign groups of atoms or residues to domains. As in general blocks overlap, a residue/atom can be situated within a number of different blocks, each of which will be assigned to a cluster. We assign a residue/atom to a cluster based on a voting procedure according to the cluster assignment of the blocks that the residue/atom is situated within. In the previous version of DynDom3D, if v_l is the number of votes a residue/atom receives for cluster *l*, then the cluster it would be assigned to, l_{max}, would be given by $\max_l v_l$. However, here, if the condition $\frac{v_{l_{max}}}{\sum_l v_l} < bend$ is satisfied, where *bend* is a threshold parameter, the residue/atom will be assigned to a bending region, otherwise the residue/atom will be assigned to cluster, l_{max}. We suggest that a value of 0.51 is a good value for *bend* as it would mean that the residue/atom would need to have at least 51% of the votes for it to be assigned to a domain.

Definition of domain and domain-movement pair

A critical threshold parameter is the minimum domain size, *mindom*, specified as the number of atoms. If the number of atoms assigned to a particular cluster is more than *mindom* then it is called a “domain”. In contrast to the previous DynDom versions we do not test whether two domains are in contact in order to pass them on to the next stage which is to analyse their relative movement. This saves a considerable amount of time and also allows one to understand the relative movement of domains that are not connected, which in some circumstances could be of interest. A pair of domains (not including the bending regions) with the ratio of inter-domain displacement to intra-domain displacement greater than the threshold parameter *ratio*, set at 1.0, is called a “domain-movement pair”. The precise expression for *ratio* is given in the original DynDom paper [11]. All domain pairs are recorded at every level of clustering.

Termination of the clustering routine

In DynDom6D the maximum value for *k* is set very high (100), as reasonable stopping criteria prevent it reaching this value. The main aim of the program is to find the largest number of domain-movement pairs. We have found that if a cluster is created that has fewer atoms than *mindom* then this is a good clustering level to stop at as it is unlikely to proceed to find any more domains after this. Thus clustering stops when a cluster is found that has fewer atoms than *mindom* or the maximum value of *k* is exceeded, the latter being very unlikely if reasonable parameter settings are used.

Determination of screw axes

The method to determine the interdomain screw axis for each domain movement pair is the same as for the original DynDom3D and DynDom. It fixes one domain of a domain-movement pair in space and then analyses the movement of the other domain as a rigid body undergoing a screw movement in accordance with Chasles’ theorem. The screw axis, the angle of rotation and the translation along the axis are determined. This is done for all domain-movement pairs.

Output

The program outputs a PyMOL script (www.pymol.org) for display of the two structures with PyMOL. The domains are coloured for identification and screw axes are depicted as arrow molecules. The shaft of the arrow has the colour of the fixed domain and the colour of the tip of the arrow has the colour of the moving domain. This colouring scheme allows one to identify axes with domain movement pairs. The two input structures are different PDB Models allowing one to animate the difference. A text file is also output which contains details of each domain movement such as angle of rotation, translation along the axis, etc.

Implementation

DynDom6D has been implemented as a standalone Java application. In creating this tool we have made use of a number of open source programs. In particular we have used the Biojava program for reading mmCIF files.

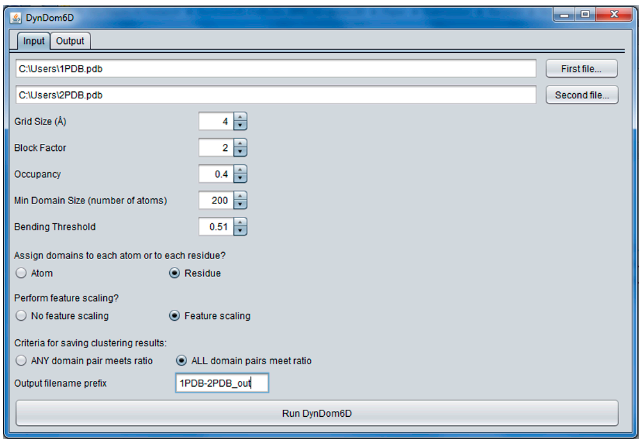
Results

Input control for DynDom6D

Figure 4 shows the input panel for DynDom6D indicating the parameters that can be set such as the grid length *g*, the block factor *b*, the occupancy, *occ*, the minimum domain size *mindom*, the bending threshold *bend* and whether one wants to use feature scaling as given in Eq. (6). There is also an option to assign individual atoms to domains and bending regions or all the atoms within a residue to a domain or bending region. The default values for *g*, *b*, *occ*, *mindom* and *bend* are 4 Å, 2, 0.4, 200 atoms, and 0.51 respectively. Although for DynDom3D a value for *occ* of 0.6 was used as a default it was found here that a value of 0.4 worked better. There are two output options. The “ANY domain pair meets ratio” option outputs the result at the clustering level prior to the exit level (when it finds a cluster with fewer atoms than the minimum domain size) irrespective of whether all domain pairs form domain-movement pairs. The “ALL domain pairs meet ratio”, outputs the result at the deepest level of clustering for which all the domain pairs are domain-movement pairs. The latter is optimal but if this condition is not satisfied then one gets a null result. For the examples below feature scaling was selected and the “ALL domain pairs meet ratio” condition was selected and satisfied.

Citrate synthase

The enzyme citrate synthase catalyses the Claisen condensation reaction between acetyl-coenzyme A and oxaloacetate yielding citrate and co-enzyme A. It has a clear domain movement and was the first enzyme to which the original DynDom was applied [11]. Figure 5 shows the DynDom6D result with default parameter settings applied



C:\Users\1PDB.pdb

C:\Users\2PDB.pdb

Grid Size (Å) 4

Block Factor 2

Occupancy 0.4

Min Domain Size (number of atoms) 200

Bending Threshold 0.51

Assign domains to each atom or to each residue?
○ Atom ● Residue

Perform feature scaling?
○ No feature scaling ● Feature scaling

Criteria for saving clustering results:
○ ANY domain pair meets ratio ● ALL domain pairs meet ratio

Output filename prefix 1PDB-2PDB_out

Figure 4 Input panel for DynDom6D showing the various parameter settings and options.

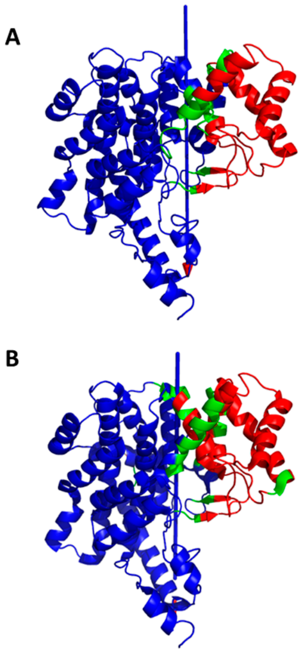


Figure 5 Comparison of DynDom6D result with the original DynDom result for the transition between the open ligand-free structure (PDB: 1CTS) to the closed citrate and co-enzyme A-bound structure (PDB: 2CTS). The structure shown is the open ligand-free structure (PDB: 1CTS) where the large domain is coloured blue, the small domain red, and bending regions, green. (A) DynDom result. (B) DynDom6D result.

to the open-free and closed ligand-bound structures. This presents a good test of the method as the result on citrate synthase is well-characterised by DynDom with the β-hairpin emanating from the large-domain forming a hinged-loop [21] with the axis passing between the bending regions on the N- and C-terminal regions of the hairpin. For comparison we show the backbone trace as a cartoon for both DynDom and DynDom6D. Although there are some differences, overall there is a good correspondence between the two results with the β-hairpin indicated in DynDom6D as being part of the small domain and its N- and C-termini indicated as bending regions with the hinge axis passing between them.

Aspartate transcarbamoylase

As referred to in the Introduction ATCase is a complex enzyme that exhibits an allosteric mechanism. We have analysed the phosphonoacetamide- and malonate-liganded R-state [25] (PDB: 1AT1) and the CTP-liganded T-state [24] (PDB: 1RAA) of *E-Coli* ATCase with DynDom6D. We needed to increase the grid length and the block factor to get a result; we used a grid length of 6.0 Å and a block factor of 5. Figure 6 shows the result. There are 5 domains forming 10 domain-movement pairs. There is a symmetry in the axes that is a reflection of the symmetry in the molecule itself. Of particular interest is the relative rotation and translation of the catalytic trimers relative to each other. The 10.7° rotation is accompanied by a 10.5 Å translation along the screw axis which moves the two catalytic trimers from close proximity in the T state to a more separated conformation in the R state. This axis is situated close to the 3-fold axis of symmetry of the molecule.

F_0F_1-ATP synthase

We analysed the structures of bovine heart mitochondria ATP synthase determined using cryo-EM [26]. The structures were classified into states: 1, 2 and 3 within which subclasses were identified resulting in states 1a, 1b, 2a, 2b, 2c, 3a, and 3b. The three main states 1, 2 and 3 are distinguished by a 120° rotation of the central rotor shaft. Here we elected to analyse the movement between states 1a and 2a (PDB: 5ARA and PDB: 5ARH, respectively). Default parameter

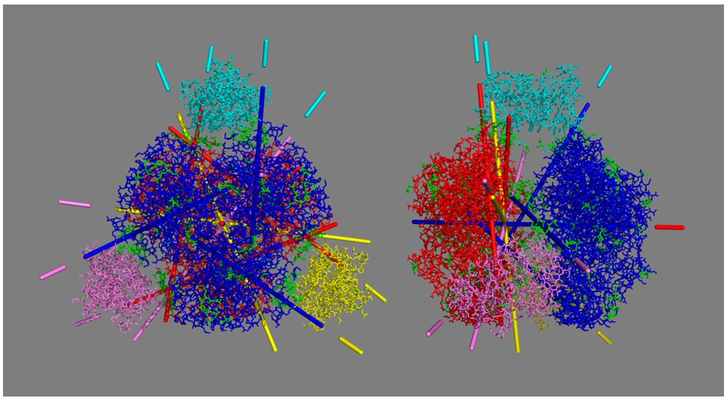


Figure 6 Face and side views of DynDom6D result on ATCase (PDB: 1AT1 vs PDB: 1RAA) where all ten domain pairs identified satisfy the *ratio* criterion, that is they are all domain-movement pairs. The three regulatory dimers form three separate domains coloured violet, cyan and yellow. The two catalytic trimers form two domains coloured blue and red. Green indicates interdomain bending regions. The relative rotation in a domain-movement pair is indicated by colour: the colour of the shaft of an axis indicates the domain fixed in space, and the colour of the separated tip of the axis indicates the moving domain. The direction of rotation is given by the right-hand rule. The structure shown is the R-state (PDB: 1AT1). There is a symmetry in the axes that is a reflection of the symmetry in the molecule itself.

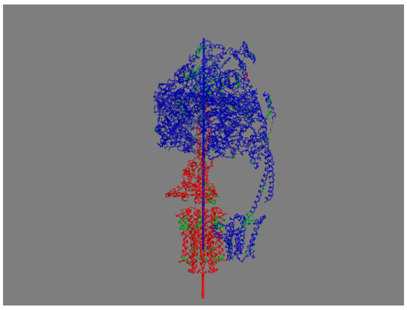


Figure 7 Result of DynDom6D applied to F_0F_1-ATP synthase (PDB: 5ARA vs PDB: 5ARH). The structure shown is state 2a (PDB: 5ARH). The red domain comprising F_0 and the rotor shaft rotates 131° relative to the blue domain comprising F_1 and the stator.

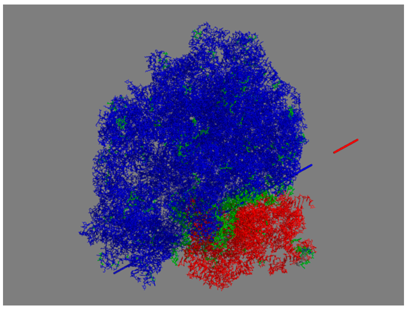


Figure 8 Result of DynDom6D applied to the E-Coli 70S ribosome structures (PDB: 5UYM vs PDB: 5UYL). The red domain forms part of the 30S subunit and the blue domain the 50S subunit and part of the 30S subunit. The structure shown is structure III (PDB: 5UYM).

settings were used. The result is shown in Figure 7. The red domain comprising F_0 and the rotor shaft rotates 131° relative to the blue domain comprising F_1 and the stator.

Ribosome

We analysed the structures of the *E-Coli* 70S ribosome, structure II (PDB: 5UYL) and structure III (PDB: 5UYM) determined using cryo-EM by Loveland, A. B., *et al.* [6]. In Structure II, the anticodon base-pairs with the codon, with Elongation Factor Tu (EF-Tu) being distant from the 50S subunit. In structure III, the anticodon base-pairs with the codon, and EF-Tu contacts the sarcin ricin loop of the 50S subunit. We used a grid length of 8 Å and a block factor of 4. Figure 8 shows the result. There are just two domains. The red domain rotates by an angle of 4.8° relative to the blue. The green region between the blue and red domains runs along part of the border between the 30S and 50S subunits.

Discussion

Here we present, DynDom6D, an implementation of the DynDom approach that brings it to a logical conclusion. The DynDom approach is that if domains move as rigid bodies then they should be identifiable from the differences in the parameters that describe these movements. These domains, which are naturally quasi-rigid and parts thereof form clusters in the parameter space, should be linked by bending regions which would also form links in this parameter space. The DynDom approach attempts to interpret the parameter space according to this model. The name of the software tool, "DynDom6D" alludes to the ability to distinguish dynamic domains using all six parameters that govern rigid body movements. In DynDom3D, where the 3D sliding block was introduced, only the three components of the rotation vectors were used for dynamic domain identification. In the original DynDom (DynDom1D), a 1-D sliding window was used although clustering was also performed in the 3D rotation space. The main improvement of DynDom6D over DynDom3D is that it also uses axis location data and axial climb data in the clustering process. To achieve this we used Chasles' theorem and developed a method for clustering lines in 3D space for use in the *k*-means clustering algorithm. We have illustrated how this works and how DynDom3D would fail in some cases to identify dynamic domains when they rotate similarly but about axes that are not co-located. Although we were aware of this problem in the development of DynDom and DynDom3D and attempted to ameliorate it by using a connected-set routine to make sure the domains identified comprised a set of connected atoms, there are certain scenarios where this could fail to identify the correct dynamic domains. A further advantage is that by circumventing use of a connected-set routine it is much faster, completing the analysis of the movement in the ribosome in a few minutes rather than many hours using DynDom3D. Another advantage of avoiding the use of a connected-set routine is that it does not require a contact distance parameter and as such it can be used on structures with missing atoms or structures that derive from coarse-grained simulation methods such as elastic network models where only C^{α} atoms may be used. DynDom6D offers a further significant improvement over DynDom3D in being able to determine interdomain bending regions which help control the domain movements. The method used is quite different to DynDom so the results differ somewhat. As we do not use a connected-set routine in DynDom6D it is possible that due to noise, parts of a single dynamic domain are disconnected from the main body of the domain. These are often located on the surface of the biomolecule. These regions might then be converted to bending regions in the voting part of the algorithm. Thus small isolated regions of colour green or regions with a different colour to the surrounding parts should be regarded as the effect of noise.

Default parameters work well for many examples but not all. It is clear that larger molecules perform better with a larger grid size and/or block factor, and molecules with thin regions separated from the main body of the biomolecule might work better with a lower occupancy threshold. More testing is required to understand how these parameters affect results but generally there is a robustness against variation in parameter values.

We have applied DynDom6D to the citrate synthase protomer, ATCase, F_0F_1-ATP synthase, and the ribosome. The results are encouraging in that application of DynDom6D results in a very large and complex set of atomic displacements being reduced to something much easier to understand. The result on ATCase is of particular note as the movement obviously reflects the symmetry present in the molecule itself. It is expected that application of DynDom6D will lead to new insights into biomolecular mechanism.

Download

The beta-test download of DynDom6D is available from the following link:
http://dyndom.cmp.uea.ac.uk/dyndom/dyndomDownload.jsp

Acknowledgements

There are no acknowledgements.

Conflicts of Interest

R. V. and S. H. declare that they have no conflict of interest.

Author Contributions

S. H. conceived the project, developed methods and co-wrote the manuscript. R. V. developed methods, co-wrote the manuscript, and produced the accompanying software tool.

References

[1] Monaco, H. L., Crawford, J. L. & Lipscomb, W. N. 3-dimensional structures of aspartate carbamoyltransferase from Escherichia-Coli and of its complex with cytidine triphosphate. *Proc. Natl. Acad. Sci. USA* **75**, 5276–5280 (1978).

[2] Krause, K. L., Volz, K. W. & Lipscomb, W. N. Structure at 2.9 Å resolution of aspartate carbamoyltransferase complexed with the bisubstrate analog N-(phosphonacetyl)-L-aspartate. *Proc. Natl. Acad. Sci. USA* **82**, 1643–1647 (1985).

[3] Boyer, P. D. The ATP synthase—A splendid molecular machine. *Annu. Rev. Biochem.* **66**, 717–749 (1997).

[4] Okuno, D., Iino, R. & Noji, H. Rotation and structure of F_0F_1-ATP synthase. *J. Biochem.* **149**, 655–664 (2011).

[5] Frank, J. & Agrawal, R. K. A ratchet-like inter-subunit reorganization of the ribosome during translocation. *Nature* **406**, 318–322 (2000).

[6] Loveland, A. B., Demo, G., Grigorieff, N. & Korostelev, A. A. Ensemble cryo-EM elucidates the mechanism of translation fidelity. *Nature* **546**, 113–117 (2017).

[7] Go, N., Noguti, T. & Nishikawa, T. Dynamics of a small globular protein in terms of low-frequency vibrational modes. *Proc. Natl. Acad. Sci. USA* **80**, 3696–3700 (1983).

[8] Gibrat, J. & Go, N. Normal mode analysis of human lysozyme: Study of the relative motion of the two domains and characterization of the harmonic motion. *Proteins* **8**, 258–279 (1990).

[9] Hayward, S. & Go, N. Collective variable description of native protein dynamics. *Annu. Rev. Phys. Chem.* **46**, 223–250 (1995).

[10] Bahar, I. & Rader, A. J. Coarse-grained normal mode analysis in structural biology. *Curr. Opin. Struc. Biol.* **15**, 586–592 (2005).

[11] Hayward, S. & Berendsen, H. J. C. Systematic analysis of domain motions in proteins from conformational change: New results on citrate synthase and T4 lysozyme. *Proteins* **30**, 144–154 (1998).

[12] Huang, E. S., Rock, E. P. & Subbiah, S. Automatic and accurate method for analysis of proteins that undergo hinge-mediated domain and loop movements. *Curr. Biol.* **3**, 740–748 (1993).

[13] Koike, R., Ota, M. & Kidera, A. Hierarchical Description and Extensive Classification of Protein Structural Changes by Motion Tree. *J. Mol. Biol.* **426**, 752–762 (2014).

[14] Hayward, S., Kitao, A. & Berendsen, H. J. C. Model free methods to analyze domain motions in proteins from simulation. A comparison of a normal mode analysis and a molecular dynamics simulation of lysozyme. *Proteins* **27**, 425–437 (1997).

[15] Hinsen, K., Thomas, A. & Field, M. J. Analysis of domain motions in large proteins. *Proteins* **34**, 369–382 (1999).

[16] Wriggers, W. & Schulten, K. Protein domain movements: Detection of rigid domains and visualization of hinges in comparisons of atomic coordinates. *Proteins* **29**, 1–14 (1997).

[17] Shatsky, M., Nussinov, R. & Wolfson, H. J. Flexible protein alignment and hinge detection. *Proteins* **48**, 242–256 (2002).

[18] Ye, Y. Z. & Godzik, A. FATCAT: a web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Res.* **32**, W582–W585 (2004).

[19] Poornam, G. P., Matsumoto, A., Ishida, H. & Hayward, S. A method for the analysis of domain movements in large biomolecular complexes. *Proteins* **76**, 201–212 (2009).

[20] Chasles, M. Note sur les propriétés générales du système de deux corps semblables entr'eux et placés d'une manière quelconque dans l'espace; et sur le déplacement fini ou infiniment petit d'un corps solide libre. *Bulletin des Sciences Mathematiques, Astronomiques, Physiques et Chimiques* **14**, 321–326 (1830).

[21] Hayward, S. Structural principles governing domain motions in proteins. *Proteins* **36**, 425–435 (1999).

[22] Hayward, S. & Kitao, A. Monte Carlo Sampling with Linear Inverse Kinematics for Simulation of Protein Flexible Regions. *J. Chem. Theory Comput.* **11**, 3895–3905 (2015).

[23] Girdlestone, C. & Hayward, S. The DynDom3D Webserver for the Analysis of Domain Movements in Multimeric Proteins. *J. Comput. Biol.* **23**, 21–26 (2016).

[24] Kosman, R. P., Gouaux, J. E. & Lipscomb, W. N. 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**, 147–176 (1993).

[25] Gouaux, J. E., Stevens, R. C. & Lipscomb, W. N. Gouaux, J. E., Stevens, R. C. & Lipscomb, W. N. Crystal-structures of aspartate carbamoyltransferase ligated with phosphonoacetamide, malonate, and CTP or ATP at 2.8 Å resolution and neutral pH. *Biochemistry* **29**, 7702–7715 (1990).

[26] Zhou, A. N., Rohou, A., Schep, D. G., Bason, J. V., Montgomery, M. G., Walker, J. E., *et al.* Structure and conformational states of the bovine mitochondrial ATP synthase by cryo-EM. *Elife* **4**, e10180 (2015).