JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

# Determining Geometrically Stable Domains in Molecular Conformation Sets

Julia Romanowska,*[,†,‡] Krzysztof S. Nowiński,[‡] and Joanna Trylska[§]

[†]Department of Biophysics, Faculty of Physics, University of Warsaw, Hoża 69, 00-681 Warsaw, Poland

[‡]Interdisciplinary Centre for Mathematical and Computational Modelling (ICM), University of Warsaw, Pawińskiego 5a, 02-106 Warsaw, Poland

[§]Centre of New Technologies (CeNT), University of Warsaw, Żwirki i Wigury 93, 02-089 Warsaw, Poland

Supporting Information

**ABSTRACT:** Detecting significant conformational changes occurring in biomolecules is a challenging task, especially when considering tens to hundreds of thousands of conformations. Conformational variability can be described by dividing a biomolecule into dynamic domains, i.e., by finding compact fragments that move as coherent units. Typical approaches, based on calculating a dynamical cross-correlation matrix, are limited by their inability to reveal correlated rotations and anticorrelated motions. We propose a geometric approach for finding dynamic domains, where we compare traces of atomic movements in a pairwise manner, and search for their best superposition. A quaternion representation of rotation is used to simplify the complex calculations. The algorithm was implemented in a Java graphical program: Geometrically Stable Substructures (GeoStaS). The program processes PDB and DCD binary files with large structural sets for proteins, nucleic acids, and their complexes. We demonstrate its efficiency in analyzing (a) ensembles of structures generated by NMR experiments and (b) conformation sets from biomolecular simulations, such as molecular dynamics. The results provide a clear description of the molecular movements even for large biomolecules. Compared to a standard dynamic cross-correlation matrix, our algorithm detects the correlations in both translational and rotational motions.

## 1. INTRODUCTION

Biomolecules are flexible, and their internal motions are often related to function.[1,2] Different molecular conformations can be probed by X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy but direct experimental observations of internal motions are both difficult and costly. There are, however, computational methods such as molecular dynamics (MD)[3] and Monte Carlo (MC)[4,5] stochastic algorithms that complement the experimental techniques by extensively sampling the conformational phase space of biopolymers. MD simulations are now a well-established computational tool for studying the flexibility,[6,7] conformational changes,[8,9] or even reaction paths[10,11] and folding[12−14] of biomolecules. MC methods are used when one is not interested in the timeline of changes but in generating a diverse set of available conformations, e.g., when analyzing folding or unfolding of biomolecules.[15]

The recent improvements in computer resources make these computational techniques more extensively used, producing a vast amount of data that must be analyzed. One of the outcomes of these simulations is a collection of biomolecular conformations. Unless one applies the so-called enhanced sampling procedures,[16,17] these conformations are often similar to each other, because the biomolecular energy landscape has multiple minima and the sampling in classical techniques is often restricted to only a few basins. Therefore, when analyzing such large amount of data, one often decreases their dimensionality and complexity, e.g., by clustering of similar conformations[18] or finding the common directions of motions.[19]

When analyzing conformations, one often sees that the molecule does not move as one unit but can be divided into several "dynamic domains", i.e., fragments that remain internally rigid in all conformations but move relative to each other. Various algorithms for finding the dynamic domains (or "core atoms") have been proposed: Hingefind,[20] DomainFinder,[21,22] FindCore,[23] DynDom server,[24,25] find.core,[26] PiSQRD server,[27] and CYRANGE method.[28] However, the automatic partition of the molecule into dynamically coherent parts is a challenging problem. Dividing molecules into dynamic domains based on their accessible conformations is important, since it facilitates observation of conformational transitions, which are often related to the function of the biomolecule.[29,30]

Hingefind, created by Willy Wriggers and Klaus Schulten,[20] was the first proposed method to distinguish protein domains. Using the least-squares fitting of two protein conformations, Hingefind finds parts of the molecule that do not change conformation and defines a rotation axis to simplify the description of internal motions. The algorithm was implemented as a script to the popular visualization package VMD,[31] but it is no longer maintained. Because of major changes in VMD, the script has limited functionality. The more commonly used algorithm, DomainFinder, proposed by Konrad Hinsen,[21,22] uses normal-mode analysis, which finds the directions of harmonic motions of atoms around their energetic minimum.[32] Hinsen's method enables fast search for the dynamic domains but accepts as an input only one or two static structures and cannot derive information from a diverse and large set of conformations, such as numerous trajectory snapshots.

**Received:** March 10, 2012
**Published:** July 19, 2012

FindCore[23] can deal with many molecular conformations. This algorithm is based on the observation of the variance of the interatomic distances. It defines the "core atoms" but does not provide a clear division of a molecule into domains. In contrast, the CYRANGE program[28] predicts the domain decomposition and it is also easy to use through a webpage interface, but it was optimized to analyze the data from NMR experiments, not MD simulations. The DynDom database[24,25] finds the domains and provides a visualization tool, but it was specifically designed to deal with molecules for which two different conformations were experimentally resolved. The method implemented in PiSQRD server[27,33] is based on the information derived from the lowest energy modes of motion through calculation of the covariance matrix. The residues are then grouped in blocks so that the fluctuations of a residue pair from one block are smaller than the fluctuations of a pair from different blocks. This approach easily captures the conformational changes that could be biologically important.

Nevertheless, all the above methods can analyze only protein coordinates and read exclusively the Protein Data Bank (PDB)[34] format, which is a drawback if one wants to analyze nucleic acids and large datasets from simulations. A method that reads any type of molecule and binary files is find.core from the R package *bio3d*.[26] It finds a "core" of a molecule, which is defined as a set of residues that are most invariant throughout a trajectory or common to a set of molecules having some sequence identity. This tool can be used to superimpose a set of structures but it does not predict the number of different domains in a molecule. Much effort has been put into creating tools that help define the molecular fragments that move as rigid bodies. However, the majority of the mentioned methods are applicable only to proteins and experimentally resolved conformations. Therefore, there is a need to create a similar tool to describe the dynamics of nucleic acids also. The structures of nucleic acids, especially of RNA, which forms complicated tertiary architectures, are equally complex, diverse, and dynamic as proteins.

Here, we present a simple yet powerful methodology for finding the dynamic domains based on data from either experiments or simulations. This generic algorithm takes, as an input, a set of conformations of a molecular system and decomposes it into domains with no a priori assumptions on the system's nature. Thus, it can be easily implemented to analyze various biomolecules. Our current implementation, Geometrically Stable Substructures (GeoStaS), processes files containing the coordinates of protein or nucleic acid atoms. [GeoStaS is an open source, and the GeoStaS software is freely available under the GNU General Public License; it can be downloaded from bitbucket.org/jrom/geostas or bionano.cent.edu.pl/Software/GeoStaS.] It is fast and automatically gives the optimal division into dynamic domains detecting the correlations in both translational and rotational motions. The software can read the PDB data files and binary trajectories. Moreover, the entire program is very light (in terms of bytes) and has a user-friendly graphical interface.

## 2. ALGORITHM AND IMPLEMENTATION

**Overview of the Algorithm.** The algorithm is based on a simple idea that observing correlations is equivalent to looking for similar motions. Instead of searching for similar global conformations of a molecule, we focus on each of its atoms and represent each individual atomic trace as one trajectory. The similarity matrix is created and the atoms are then clustered based on the geometric similarity of the trajectories (described in detail below).

The standard method to detect a common mode of motion of two atoms consists of a simple evaluation of the correlation coefficients of the 3N vectors of all coordinates from the trajectory. The formed dynamic cross-correlation matrix (DCCM) is defined as

$$C_{ij} = \frac{\left\langle \vec{d}_i \cdot \vec{d}_j \right\rangle}{\sqrt{\left\langle \vec{d}_i \right\rangle^2 \left\langle \vec{d}_j \right\rangle^2}}$$

where $\vec{d}_i = \vec{r}_{i(k)} - \langle \vec{r}_i \rangle$ denotes a deviation of atom $i$ in a configuration $k$ from its mean position. This method properly detects the common *translations* but consistently fails to detect common *rotations*. As a proof, let us analyze a simple case of two points $a = r_a(\cos\alpha, \sin\alpha)$ and $b = r_b(\cos\beta, \sin\beta)$, rotating rigidly around the origin of a plane (see Figure S1 in the Supporting Information). Their trajectories can be described with a sequence of angles $(\phi_1, \phi_2, ..., \phi_n)$ such that $a_i = r_a(\cos(\alpha + \phi_i), \sin(\alpha + \phi_i))$, and $b_i = r_b(\cos(\beta + \phi_i), \sin(\beta + \phi_i))$. The correlation coefficient for these two trajectories would be $C_{ab} \approx \cos(\alpha - \beta)$ and therefore it cannot be used to determine whether the motions of $a$ and $b$ are geometrically correlated.

To overcome this difficulty, we propose to estimate the similarity of motion of $a$ and $b$ by finding an isometry (translation plus rotation) of the trajectory of $a$ that maximizes the correlation with the trajectory of $b$—we search for an isometry that would bring the two trajectories as close as possible. The translational part of such an isometry is easy to find—it is sufficient to match the initial coordinates of both trajectories.

The rotation of an object in the three-dimensional Cartesian space ($\mathbb{R}^3$) is described by multiplication by a $3 \times 3$ matrix that contains trigonometric functions of the rotation (Eulerian) angles. The optimization problem of finding the best rotation outlined above can be solved in the space of the matrix coefficients (i.e., in the Cartesian coordinates). However, the conditions of orthonormality of the rotation matrix act as optimization constraints, which forces one to use more-complicated and less-efficient optimization algorithms. On the other hand, in the unconstrained optimization, the representation of the rotation matrix by the Eulerian angles $(\phi,\psi,\omega)$ is not unique and singular (e.g., for $\omega = \pi/2$—the so-called "gimbal lock" phenomenon). Therefore, we have adopted the representation of rotations in an abstract space by using quaternions, because of its functional and algorithmic uniformity.

**Rotation in quaternion space.** The quaternion is a four-tuple[35] $q = q_0 + \vec{q} = q_0 + q_1 \cdot \vec{i} + q_2 \cdot \vec{j} + q_3 \cdot \vec{k}$, where $q_0$ is named the scalar part and $\vec{q}$ is the vector part. It was shown[36] that applying a quaternion operator to a vector can be interpreted as a rotation in $\mathbb{R}^3$:

$$\vec{v}' = q\vec{v}q^{-1} \tag{1}$$

$$q = q_0 + \vec{q} = \cos\left(\frac{\alpha}{2}\right) + \vec{u}\cdot\sin\left(\frac{\alpha}{2}\right) \tag{2}$$

where $\vec{v}'$ is $\vec{v}$ rotated by the angle $\alpha$; $\vec{u}$ is the direction of $\vec{q}$, i.e., the normalized vector part of $q$; and $q^{-1} = q^*/|q|^2$. Thanks to this concept, the rotation in the three-dimensional Cartesian space can be described with the use of only one quaternion, i.e., four parameters.

Our implementation of rotation in quaternion space is based on the theory presented by Kneller et al.[37] The problem of finding the rotation that would best superimpose the two objects is reduced to solving an eigenproblem of the following form:

$$\mathbf{M}\cdot q = \lambda q \tag{3}$$

where $q$ is the quaternion that describes the rotation and $\lambda$ is a diagonal matrix. Matrix $\mathbf{M}$ is of the following form:

$$\mathbf{M} = \sum_{\alpha=1}^{N} \begin{bmatrix} (\mathbf{x}_\alpha - \mathbf{x}'_\alpha)^2 & \mathbf{u}_\alpha^{\mathrm{T}} \\ \mathbf{u}_\alpha & \mathbf{P}_\alpha \end{bmatrix} \tag{4a}$$

$$\mathbf{u}_\alpha = \mathbf{x}_\alpha \times \mathbf{x}'_\alpha \tag{4b}$$

$$\mathbf{P}_\alpha = \mathbf{x}_\alpha \cdot \mathbf{x}'^{\mathrm{T}}_\alpha + \mathbf{x}_\alpha^{\mathrm{T}} \cdot \mathbf{x}'_\alpha \tag{4c}$$

where $\mathbf{x}_\alpha$ and $\mathbf{x}'_\alpha$ describe the positions of the two atoms, respectively, in a conformation $\alpha$.

Our program calculates the correlation of the translated sequences of atomic positions $(\mathbf{x}_\alpha)$ and $(\mathbf{x}'_\alpha)$ by evaluating the sum proportional to $\sum \mathbf{x}_\alpha \mathbf{x}'_\alpha$. For an MD trajectory, the index in the sequence is just the time step. Thus, in the case of different frequencies, the correlation will be close to 0 by a standard Fourier orthogonality argument. In our method, the two superimposed atomic trajectories are treated as two molecules, so that the positions from subsequent conformations are compared. Therefore, we are able to differentiate between correlated and noncorrelated movements, even though there is no time dependence in the algorithm.

As a result of solving eq 3, we obtain four eigenvalues $(\lambda_i)$: the smallest one symbolizes the best superposition, while the largest one is the worst superposition. In addition, the largest eigenvalue gives also the orientational distance between the two compared trajectories:[37]

$$\Delta_\Omega = \left(\frac{M_{11}}{\lambda_4}\right)^{1/2} \tag{5}$$

and $\Delta_\Omega \in [0;1]$. With this normalized distance, one can also determine whether the objects are superimposed in a parallel fashion $(\Delta_\Omega \approx 0)$ or an antiparallel fashion $(\Delta_\Omega \approx 1)$. Here, we focus on the atoms whose movements are well-correlated with each other (as opposed to *anti*-correlated). Therefore, we define a similarity coefficient as $SC = 1 - \Delta_\Omega$, so that higher values would describe better correlations.

**Assigning Atoms to Domains.** After having compared each pair of atomic trajectories, we obtain a matrix of similarity coefficients, which we call atomic movement similarity matrix (AMSM). In order to find the domains, we need to cluster the AMSM. We have tested two clustering algorithms, which represent two different classes: a hierarchical merging (HM) of AMSM columns, and a graphlike algorithm, based on the common nearest-neighbors (NN) algorithm.[38]

In the HM algorithm, initially each column (vector) of AMSM forms a separate cluster. In every step of the algorithm, the two clusters that have the smallest mutual distance are merged. The distance is calculated as the difference between the average vectors from the two clusters. Therefore, in each step, only the most similar clusters are merged. The disadvantage of this algorithm is that the desired number of domains (clusters) is required a priori to make the algorithm stop when this specific number of clusters is reached. Fortunately, one can deduce this limit, e.g., by plotting the distance between the merged clusters versus the step of the algorithm. Sudden large changes in the distance indicate merging of unrelated clusters.[18] Therefore, when testing the HM algorithm, we first performed complete merging (i.e., until all the clusters were merged into one), and after studying the distance-versus-step plot, we chose a limit. The advantage of HM is that the user can observe how the clusters were merged and stop merging at any level.

In the NN algorithm, the atoms are represented by the nodes of the graph (see Figure S2 in the Supporting Information). Each two atoms (nodes) that are more than four residues apart in sequence are connected with each other in the graph by an edge weighted with the corresponding similarity coefficient $(w_i = SC_i)$. The nodes that are connected by an edge are called neighbors. The atoms that are less than four residues away from each other, naturally have a high similarity coefficient, and connecting them with an edge in the graph would bias the clustering. This especially holds true for the force-field-based simulations where the interactions up to the third neighbor in the atom sequence contribute to the so-called bonded potential energy terms — bond, planar, and dihedral angle (see, e.g., page 213 in ref 5). Chart 1 presents the pseudo-code for the nearest neighbor algorithm.

The nearest neighbors of $g$ are defined as nodes that are connected to $g$ by an edge with the weight of more than a given $w_{\min} = w_{\max} - \Delta w$ (we call $\Delta w$ the threshold). We set $n_{\mathrm{comm}}$ equal

**Chart 1**

```
for each edge V not yet assigned do
  find the edge V_max with the maximal weight, w_max = SC_max;
  take the nodes, a and b, that are connected by V_max and create a new cluster C;
  for each node ∈ {g is a neighbor of a or b} not yet assigned do
    if g has minimum n_comm common nearest neighbors with any g_c ∈ C then
      assign g to the cluster C;
    endif
  endfor
endfor
```

to 0.8 of the number of the nearest neighbors of $g$ (we also checked smaller and larger values of $n_{comm}$), and we test several values of $\Delta w$: 0.3, 0.25, 0.2, 0.15, and 0.1. We are aware that the chosen set is limited and sometimes the best solution would be gained with another threshold, so we added an option to the program to manually enter a desired threshold. We have chosen this clustering algorithm because it is well-suited to our problem. By clustering the AMSM, one seeks groups of atoms that have the highest similarity of movement. This is achieved here by adding to the cluster only those atoms that move in a similar manner as the majority of atoms that are already in the cluster.

**Automatic Mode.** When the user does not know in advance which threshold to use, the automatic mode of NN clustering can be chosen. In this mode, a set of divisions into dynamic domains is generated for different thresholds. The divisions are then compared based on the range of the root-mean-square deviation (rmsd) within each domain:

$$R_{\text{division}} = \frac{1}{N_{\text{res}}} \sum_{D} \left\langle \sqrt{\frac{1}{N} \cdot \sum_{i} [\vec{x}_{i(k)} - \vec{x}_{i(0)}]^2} \right\rangle_K \qquad (6)$$

where $N_{res}$ and $N$ are the numbers of all residues and all atoms in the molecule, respectively; $D = \{d_i\}$ are the domains that have more than two residues; $i$ traverses all the atoms in the domain $d_i$; $\vec{x}_{i(k)}$ describes the position of an atom $i$ in the conformation $k$; $\langle\cdot\rangle_K$ denotes an average over the set of conformations $K$. Prior to calculations, the domains are least-squares-fitted on the average conformation. The minimal value of $R_{division}$ points to the optimal division of the molecule into dynamic domains. Sometimes, however, the optimal division does not mean the best one—the assessment depends on how much detail one needs to characterize to answer the studied biological problem. The user can adjust the level of detail by using the manual mode.

The final step of the algorithm is to smooth out the division. This is needed because sometimes one dynamic domain that includes many subsequent residues is divided by one or two residues in the middle that were assigned to another domain. As long as these "interrupting" residues are not numerous, we consider them as noise in the final output of the program. For a more visually appealing, and also a more informative output, we eliminate these too-short domains. This is done by scanning the molecule's sequence and looking for segments less than four residues in length that are assigned to one domain. These small segments are simply reassigned to a previous domain.

**Implementation.** The Java programming language was chosen because of its modularity, portability, and ease of user interface design. The overall scheme of one run of GeoStaS is illustrated in Figure 1 and is described below. The modules include (a) the PDB reader, (b) the trajectory reader (text or binary format), and (c) the domain finder.

The *PDB Reader* simply processes the text file formatted according to the latest PDB specifications—version 3.3 (www.wwpdb.org/documentation/format33/v3.3.html). Then, it creates an instance of the Molecule object, storing all the information about atoms and bonds.

Next, the *Trajectory Reader* processes either the text file (again in the PDB format) or the binary file (in the DCD format, used, e.g., by the popular molecular dynamics softwares NAMD[39] or CHARMM[40]) and creates the Dynamics object. Global translational and rotational movement is removed, based on the algorithm for fast finding of the optimal rotation matrix described in Appendix A1 in ref 41. The tests performed without this initial superimposition

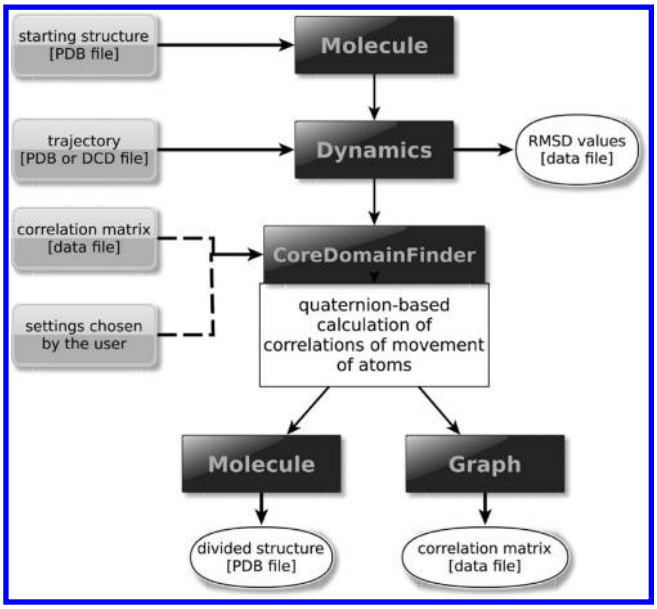


**Figure 1.** Scheme of the workflow of the GeoStaS program, including the user input, the most important Java objects that are created, and the possible output. The dashed arrows depict the optional input. For a detailed description, see the Implementation section.

yielded AMSM with the same pattern as that for the superimposed trajectory. However, without the superimposition, the correlation coefficients were scaled down, which made clustering less sensitive to smaller differences between atomic movements, while these can still be important. Thus, this initial transformation of coordinates does not introduce artificial motions: it only simplifies the AMSM clustering process. In addition, while reading the trajectory, the rmsd values are calculated, including the deviation of the atomic coordinates from the first frame, and from the previous frame. These data can be saved to a text file.

The inputted and transformed data are then analyzed by the *Core Domain Finder*. The user must choose whether to consider only the C$\alpha$ atoms (or phosphorus, in the case of nucleic acids) or three atoms per residue (i.e., C$\alpha$−C−N for proteins and P−C3′−C4′ for nucleic acids). In addition, two clustering algorithms are currently available: NN or HM; for the NN algorithm, the user can select the "automatic" or "manual mode". We recommend using the default settings, i.e., "only C-alpha/P" and "automatic mode", if the user has no a priori knowledge about the studied molecule. The module calculates the AMSM, which can be saved to a text file, for later use. In addition, in the case of HM clustering, the merging tree can be also saved to a text file, in a commonly used Newick tree format (described in http://evolution.genetics.washington.edu/phylip/newicktree.html). The last step is the creation of the Molecule object, where each different chain represents a different dynamic domain. The molecule can be written to a PDB file and visualized in third-party software.

The most computationally intensive calculations have been parallelized with the use of standard Java concurrency tools. Good parallelization efficiency has been observed for up to eight parallel threads. The source code, manual, and tutorial can be downloaded from the software web page.

## 3. RESULTS AND DISCUSSION

To evaluate the performance and ability of GeoStaS to correctly annotate dynamic domains of biomolecules, we applied our

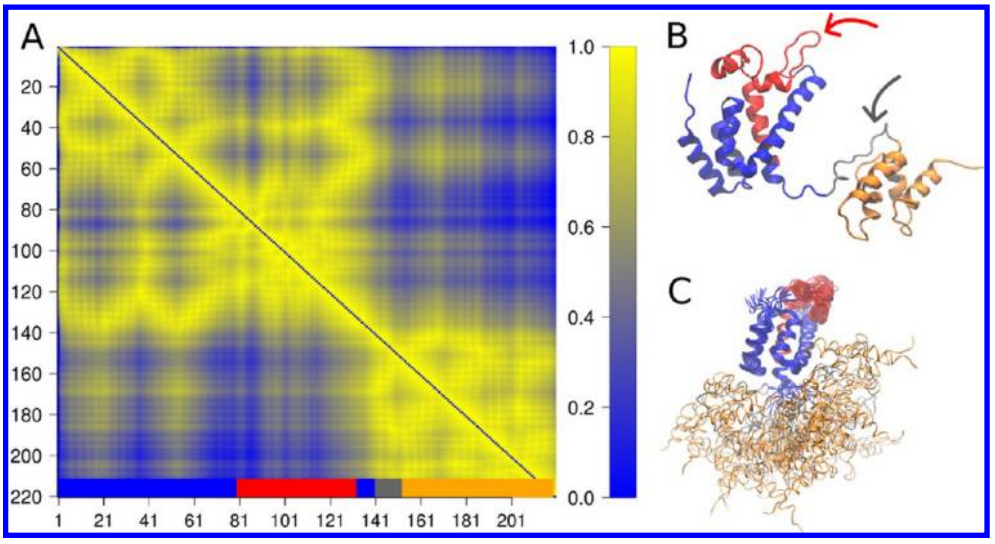


**Figure 2.** Visualization of the results obtained from GeoStaS, for the NMR ensemble 1D1D (with the NN clustering algorithm): (A) the atomic movement similarity matrix (AMSM), the color bars in the bottom depict which residues are assigned to different domains found, and the axes show residue numbers; (B) one molecular conformation colored according to different dynamic domains found by GeoStaS, and (C) the whole NMR ensemble superimposed on the blue domain. In all panels in this figure, matrices were plotted in the R environment[57] and molecules were visualized in VMD.[31]

program to several test cases originating from (a) NMR ensembles downloaded from the PDB and (b) several MD-generated trajectories of proteins and nucleic acids. In order to check the quality of the different divisions, we calculated $R_{division}$, of which the minimal value points to the optimal division (see the Algorithm and Implementation section for details). We compared the results with the CYRANGE software,[28] which is dedicated for the refinement of the NMR ensembles. The outcome of our analysis of the simulations was also compared with the division of these biomolecules into "static", structural, or functional domains.

**Tests on NMR Ensembles.** Figure 2 shows exemplary results from GeoStaS: the atomic movement similarity matrix (AMSM), and the molecule colored according to predicted dynamic domains. This test case presents an NMR ensemble of conformations of the *Rous sarcoma* virus capsid protein (PDB id 1D1D).[42] The division into dynamic domains found by GeoStaS agrees with the analysis of the NOE signals,[42] which shows that the N-terminal (red and blue fragments in Figure 2B) and C-terminal (orange fragment) domains move independently. In addition, our analysis demonstrated that the N-terminal part was subdivided into two dynamic domains: red and blue. This is also consistent with the analysis of experimental relaxation data,[42] which shows that one linker between the helices (indicated by the red arrow in Figure 2B) is similarly flexible as the linker between the domains (gray arrow). Thus, the movements of the helices connected with this red-colored linker could differ.

We tested our algorithm on all NMR ensembles mentioned in the work of Kirchner et al.[28] and on some additional nontrivial sets with many conformations. In these tests, we used the NN clustering of AMSM, since it gives an optimal solution and compares the one-atom- and three-atoms-per-residue divisions. The summary of the results, and the comparison with the CYRANGE divisions, are presented in Figure 3.

Overall, for proteins, GeoStaS shows a similar domain division as the CYRANGE software (see Figure 3). However, in some cases, our software gives a more-detailed division, defining two domains in place of one identified by CYRANGE. This is probably caused by somewhat different aims of the two programs. CYRANGE was optimized to give sets of atoms that are stable in the NMR ensemble, while our objective was to define the domains that contain similarly moving atoms. Also, our method was optimized for a larger set of conformations. The division found by the automatic mode for the NMR ensembles with a small number of conformations depended on their diversity. That was the case with the 2VIL PDB structure (see Figure S3 in the Supporting Information), where the optimal division into domains found by GeoStaS seems too complex. Visual inspection shows that the overall conformation of the protein is maintained in the NMR ensemble, but a small number of conformations (only 11) and very flexible loops make the division more difficult than, for example, in a similarly small but less structurally diverse ensemble 2SPZ. However, when manually setting a larger threshold of 0.3, a visually better division of 2VIL was found (see Figure S3B in the Supporting Information). Equation 6 produces an average of the rmsd of each domain over the number of conformations; therefore, the more conformations one has, the better the average would represent the input set.

In general, analyzing the movement of only the C$\alpha$ or phosphorus (P) atoms suffices to identify the dynamic domains correctly. However, in some cases, we have found that AMSM calculated for C$\alpha$ only is too uniform for the algorithm to distinguish the domains, even for a very small threshold parameter. These domains are detected only if one takes into account the two additional atoms per residue, as was the case for the 2K3C structure (Figure 3). Interestingly, we have also found an example where the analysis of the C$\alpha$ AMSM led to a too-complex division, and only with three atoms per backbone, the algorithm found a reasonable set of domains (PDB id 2L14; see Figure S4 in the Supporting Information). In most cases, the lower $R_{division}$ value indicated a more sensible division of the molecule.

The analysis of the NMR ensemble 1BF8 shows that, although GeoStaS, using the C$\alpha$ representation, finds almost the same division as CYRANGE, the superimposition is not the best one (see Figure S5 in the Supporting Information). However, the three-atom-per-residue representation gives a different result, which shows that the domains colored yellow and blue move in

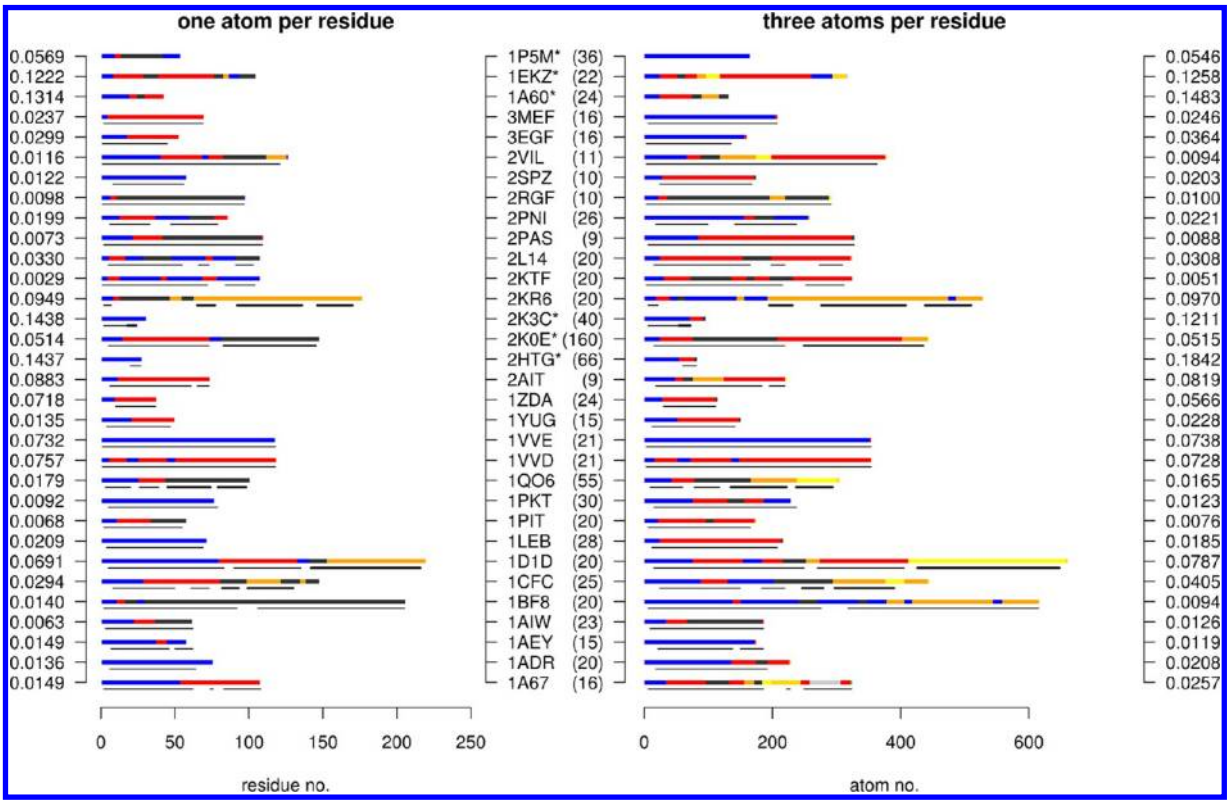


**Figure 3.** The optimal divisions for all tested NMR ensembles for (left) the one-atom-per-residue representation (C$\alpha$ or P) or (right) the three-atoms-per-residue representation (C$\alpha$−C−N or P−C3′−C4′) (NN clustering method); each dynamic domain is colored differently. The NMR ensembles are identified by their PDB codes, with the number of conformations given in brackets. The numbers on the far right and left sides show the minimal $R_{division}$ value for each of the ensembles. For comparison, the divisions obtained with the CYRANGE program are displayed as black lines below every color bar. CYRANGE gives a range of residues that belong to each domain, and these can have gaps; here, different domains, if identified, are marked with different line widths; the lines for the three-atoms-per-residue mode are scaled by a factor of 3, to match the scale of the colored bars. The structures that were not tested in ref 28 are marked with an asterisk. The structures containing nucleic acids, which CYRANGE does not process, could not be compared; these include 1EKZ (protein/RNA complex), as well as 1A60 and 1P5M (both contain RNA).

relation to each other. This division has also a smaller $R_{division}$ value (Figure 3).

We stress here that, as with all software, inspection of the results is necessary and, in certain cases, manual settings are required to obtain biologically relevant outputs. However, this is a much easier task than manually analyzing many thousands of conformations that one obtains from MD or MC simulations.

Only limited structural data are available for nucleic acid NMR ensembles with more than 20 conformations (which would be the most relevant to test). One such ensemble (1P5M) contains 36 conformations of the HCV IRES domain IIa[43]—a non-canonical RNA helix with a large internal loop region, formed by five bases (A53, A54, C55, U56, and A57). These bases are known to introduce a bend in the structure, which was correctly captured by GeoStaS, showing a wide spectrum of the movements of the two flanking helix regions (Figures 4A−C). Also, the analysis of the 1EKZ NMR ensemble[44] (complex of a double-stranded RNA with a double-stranded RNA-binding domain from an *E. coli* protein) captured the differences in the RNA conformations and illustrated how the protein adapts to these changes. As shown in Figures 4D and 4E, we observed that only a small part of the protein "moved along" with RNA (gray domain), to maintain the hydrogen bonds contributing to a stable complex.

With GeoStaS, the conformational changes can be easily identified for various NMR ensembles. However, one must be careful because many NMR ensembles are too small to provide a reasonable division into domains. To obtain dynamic domains, it is better to have a larger number of conformations, which reliably represent the accessible conformational space of the molecule.

**Tests on Simulation Datasets.** We present the performance of GeoStaS on the data derived from MD simulations, for the trajectories that contain at least thousands of conformations. We have found that the default software settings (i.e., the C$\alpha$ mode, NN clustering, and automatic mode) are, in most cases, optimal; however, for large molecules, they may give too coarse divisions. Our analyses of trajectories of an elongation factor *G* (described below)[45] and GroEL chaperone monomer complexed with ATP (Figure 5)[9,46] are examples that required manual setting of the threshold to as low as 0.05. Higher values, specifically the threshold that was found to give the smallest $R_{division}$ value, gave good indications for the coarse domains but setting a lower threshold allowed us to identify also smaller domains.

The chaperonin GroEL is a large oligomeric protein (ca. 800 kDa) that facilitates folding of non-native substrate proteins in *Escherichia coli*.[47] Together with the co-chaperonin GroES, it forms a protective chamber, where the substrate proteins regain their functional shape. Along the functional cycle, the subunits of GroEL undergo large conformational changes, which have been extensively characterized.[9,48,49] Therefore, GroEL represents an attractive system for validating and benchmarking our software.

Based on a 50-ns-long MD trajectory, our analysis reveals that the GroEL subunit can be divided into multiple dynamical


dx.doi.org/10.1021/ct300206j | J. Chem. Theory Comput. 2012, 8, 2588−2599

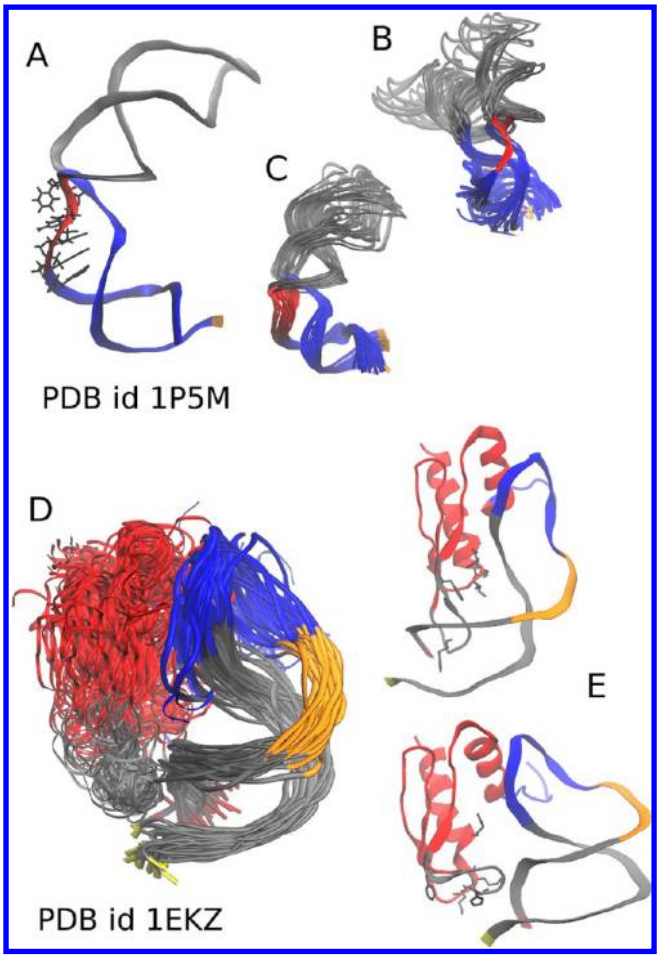


**Figure 4.** NMR ensembles containing nucleic acids, colored by domains recognized by GeoStaS (C$\alpha$ representation, automatic mode). RNA noncanonical helixes are shown in panels (A)−(C): (A) one conformation with the loop-forming bases marked as black sticks, and all NMR conformations superimposed with respect to (B) the red domain or (C) the blue domain. (D and E) Protein/RNA complexes are shown in panels (D) and (E): (D) all conformations superimposed with respect to the gray domain, and (E) two chosen conformations illustrating how the protein "follows" the nucleic helix. Amino acids whose mutations resulted in abolishing of RNA binding[44] are shown as sticks.

domains (see Figures 5B and 5C): the bottom (fragments colored blue, gray, and red), the top (gold and yellow), and the intermediate (orange). This largely corresponds to the standard separation into structural domains: equatorial (bottom), intermediate, and apical domain (top) (Figure 5A).[48] GeoStaS also provided a more-detailed dynamical division of the GroEL subunit. It showed that the majority of the yellow part of the apical domain fell into the region of helices K and L: these two helices stay close to each other during the elevation of the apical domain while forming the chamber in GroEL assembly; therefore, their movement must be more correlated with each other than with the rest of the top part. Moreover, the division of the equatorial domain suggested two subdomains (colored red and gray). The red helices are involved in ATP binding; thus, their movements should be somehow different from the rest of the bottom part of the molecule. This is in good agreement with previous structural studies highlighting the internal motions in the equatorial domain upon ATP binding.[9,50]

Apart from a simple visualization of the dynamic domains found in the GroEL monomer by GeoStaS, the analysis of the AMSM itself gave further interesting insight. Typically, in order to capture the correlated motions, one calculates the dynamical cross-correlation matrix (DCCM, see the Algorithm and Implementation section). Overall, the correlation patterns in the standard DCCM and in AMSM are in good agreement with each other, but some discrepancies can be noticed (see Figure 6). For example, AMSM showed correlations of motions of helices A, B, and C with helix M and their anticorrelations with helices K and L, which was contrary to the DCCM analysis. Helices A and C are involved in ATP binding and, together with the helix M, they move in the same direction during the first stage of conformational changes in the GroEL cycle.[49] Meanwhile, helices K and L are in the apical domain, which, during the transition, moves away from the equatorial domain (so helices A, B, and C); thus, it seems reasonable that their movements are not correlated. Interestingly, helices F and G were anticorrelated with helices Q and R in the DCCM analysis, in contrast with our observations. It is known that helices F and G must move slightly toward the equatorial domain (thus, also toward helices Q and R), in order for the apical domain to move upward and then to a fully opened conformation. An analogous case is observed between helices N and O and helix F.

Furthermore, the clustering of DCCM with the NN algorithm and automatic mode (the color bar in Figure 6A) yielded a division into 13 dynamic domains (see Figure S6 in the Supporting Information). However, this division does not correlate with the mentioned experimental and theoretical findings.[9,48−50] Only the apical domain was correctly identified as a separate dynamic entity (colored gold), while other domains contain short fragments of the monomer, most often one or two helices. This division seems too complex and not informative. Setting different thresholds ($\Delta w$ = 0.05, 0.15, 0.2, and 0.3; the optimal division was for $\Delta w$ = 0.1) resulted in either even more-complex division or almost the entire structure assigned to one domain.

Another test was performed on a 25-ns trajectory of a bacterial elongation factor, EF-Tu[45] (Figure 7). This globular protein binds GTP to acquire an active form, which, in turn, binds aminoacylated tRNA and delivers it to the ribosome. The subsequent release of cognate tRNA and dissociation of EF-Tu:GDP from the ribosome is controlled by GTP→GDP hydrolysis. The MD studies of EF-Tu:GDP complex[45] showed that some parts of the protein were significantly more dynamic when GDP was bound than without the nucleotide—these fragments are termed switch I and II.[51] The dynamics of switch I and II may be involved in triggering the release of EF-Tu from the ribosome.[52,53] As shown in Figure 7B, our analysis identified switch I as a distinct domain (colored orange), which suggests that the movements of this fragment are independent. Indeed, the visualization of the trajectory showed that this switch changes its local structure from an $\alpha$-helix to a $\beta$-turn, which corroborates experimental observations.[52,53] The switch II was distributed into two domains, with the majority of its residues assigned to fragments of the red domain, which suggests that its movements differ from the movements of the surrounding residues (i.e., the gray domain; see Figure 7B).

The comparison of the DCCM and AMSM matrices (Figure 8) indicates some differences. The frame marked with number "1" encloses correlations of movements between switch I and domain II of EF-Tu (see Figure 7A). The DCCM analysis showed a positive correlation between the motions of these

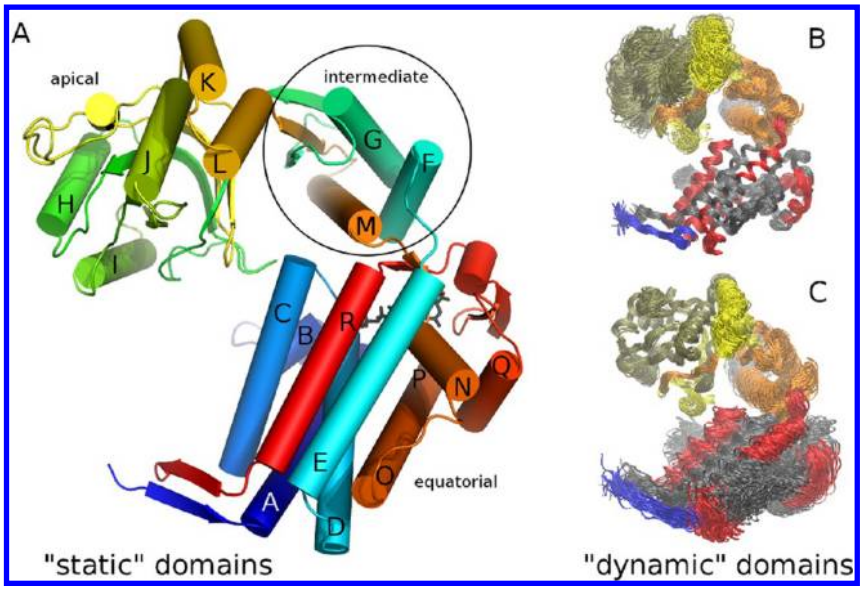


**Figure 5.** (A) Cartoon model of the monomer from the *E. coli* GroEL complex; the colors reflect the primary structure: from blue (C-terminus) to red (N-terminus). (B and C) Visualization of the domains found by GeoStaS for the NN clustering in the manual mode (threshold, $\Delta w = 0.05$)—several trajectory conformations are superimposed relative to the domains colored (B) gray and (C) gold.

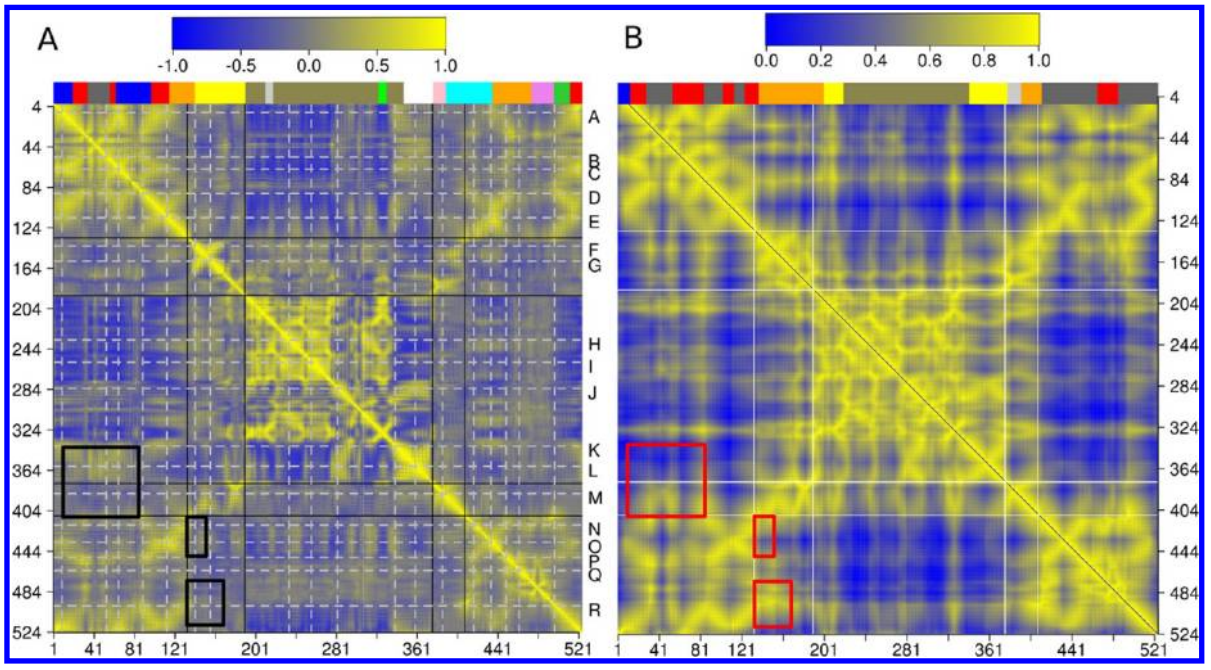


**Figure 6.** Comparison between (A) the standard dynamic cross-correlation matrix (DCCM) and (B) the atomic movement similarity matrix (AMSM), calculated by GeoStaS. Some of the areas that differ are marked by red and black squares and are discussed in the text; the meaning of the color scales is the same in both matrices: dark blue shows strong anticorrelations, and yellow indicates strong correlations. Above the matrices, the color bars represent different dynamic domains (the coloring above AMSM is based on NN clustering and is the same as in Figures 5B and 5C). The axes show residue numbers, and the letters to the right of the DCCM mark the helices shown in Figure 5A.

fragments, while, in AMSM, these motions were anticorrelated, which seems more reasonable since switch I unfolded during the simulation. The area marked with number "2" shows correlations between the fragments of domain I and domain III, which were positive according to DCCM and negative according to AMSM. Monitoring the distances between the centers of mass of these domains demonstrated that domain I moves away from domain III,[45] and our AMSM analysis correctly captured this trend. Finally, according to AMSM, the movement of domains II and III was correlated, contrary to DCCM predictions (the area marked with number "3" in Figure 8). This again agrees with the original analysis, where researchers reported that these fragments did not recede.

We have checked how these differences between DCCM and AMSM are transferred onto division into dynamic domains by applying the NN clustering algorithm also to DCCM. The color bars below the matrices in Figure 8 show that the dynamic domains found in each of these analyses differ. This was expected because of the differences in correlations outlined above. Switch I was assigned to the dynamic domain colored blue, which also


dx.doi.org/10.1021/ct300206j | J. Chem. Theory Comput. 2012, 8, 2588−2599

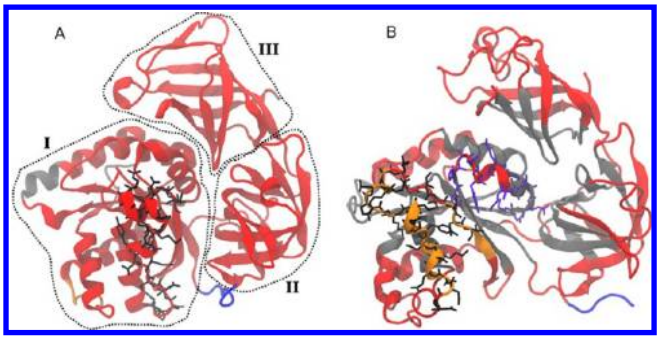


**Figure 7.** Division into dynamic domains of (A) EF-Tu (first frame of the simulation) and (B) EF-Tu:GDP complex (last frame of the simulation; ligand is not shown, for the sake of clarity). Each structure is colored independently, based on dynamic domains found with the default settings of GeoStaS. Structural domains are encircled and numbered in panel A. Residues that form switch I are depicted as sticks and is colored black in both panels; in panel B, switch II also is marked as sticks and is colored violet.

contains domains I and II. Therefore, this suggests that the movements of switch I did not differ from these domains, which is in contrast with the mentioned studies.[45,51] Similarly, domain III was assigned to a different dynamic domain than domain I (yellow and blue, respectively), although their movements were not anticorrelated.

The differences between DCCM and AMSM arise from the fact that the standard DCCM averages the deviations of the atomic positions over the entire Cartesian space, which can lead to the loss of information about rotational correlations or anticorrelations. AMSM describes the motions abstracting from an external coordinate space and, moreover, it keeps the information about the relative direction of each pair of atomic motions. Therefore, AMSM can capture correlations that are not seen in a typical DCCM matrix.

As mentioned earlier, GeoStaS was designed to work also with nucleic acid systems. Here, we show the domain decompositions performed on MD trajectories of an elbow segment of helix 38 (H38) of 23S rRNA from *Haloarcula marismortui* (200-ns-long MD simulation[54]) and of bacterial elongation factor EF-Tu complexed with tRNA (25-ns-long trajectory[45]).

Protein synthesis in bacteria requires two elongation factors (EF); EF-Tu, as mentioned above, associates with amino-acylated-tRNA and delivers it to the ribosome, while EF-G facilitates the translocation of tRNAs through the ribosome. Both structures (i.e., EF-G and EF-Tu:tRNA complex) are of similar elongated shape and are molecular mimics that bind to the same site on the ribosome exhibiting significant conformational changes. Analysis of these simulations with the default settings of GeoStaS gave good indication of the dynamic domains; however, choosing the HM clustering method provided much clearer division. EF-G was divided into five structural domains[55] (Figure 9A); however, based on the MD simulations, we find that they do not necessarily correspond to its dynamical domains (Figures 9B and 9C). Our analysis pointed to the core containing structural domains II, III, and V, which can be classified as one dynamic fragment. The G′ insertion, which is exclusive for EF-G, was classified together with domain I as a separate dynamic domain. It is known that structural domain IV bends upon binding of EF-G to the ribosome, thus assigning it to different dynamic domains by GeoStaS seems reasonable.

The division of EF-Tu:tRNA complex showed that the anticodon arm of tRNA was a separately moving part (Figures 9E and 9F), while the acceptor stem, together with the protein domains II and III, formed one dynamic domain (colored red), which shows that the motion of tRNA is coupled to the motion of the protein. Overall, the dynamic domains of tRNA correspond to its functional parts (Figure 9D). Interestingly, the visualization of the dynamic domains found by GeoStaS immediately shows the similarities between the internal movements of the

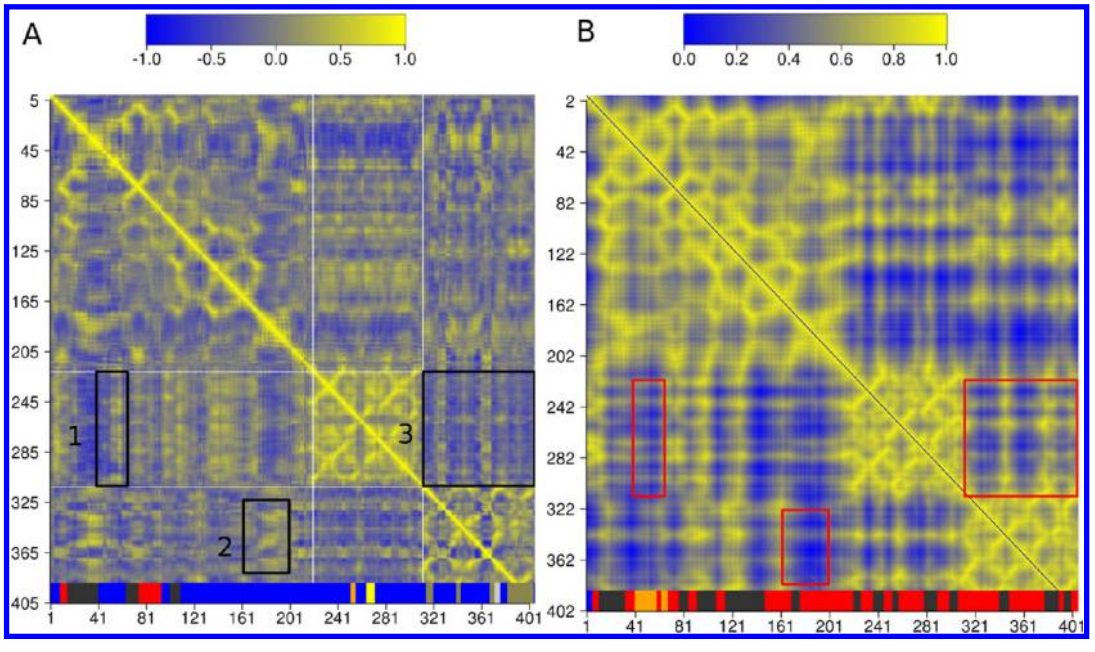


**Figure 8.** Analysis of correlation patterns of the EF-Tu:GDP complex. (A) DCCM and (B) AMSM. Some of the areas that differ are marked by red and black squares and discussed in the text. The meaning of the color scales is the same in both matrices: dark blue shows strong anticorrelations, and yellow indicates strong correlations. Below the matrices, the color bars represent different dynamic domains (the coloring below AMSM is based on NN clustering and is the same as that in Figure 7B). The axes show residue numbers.

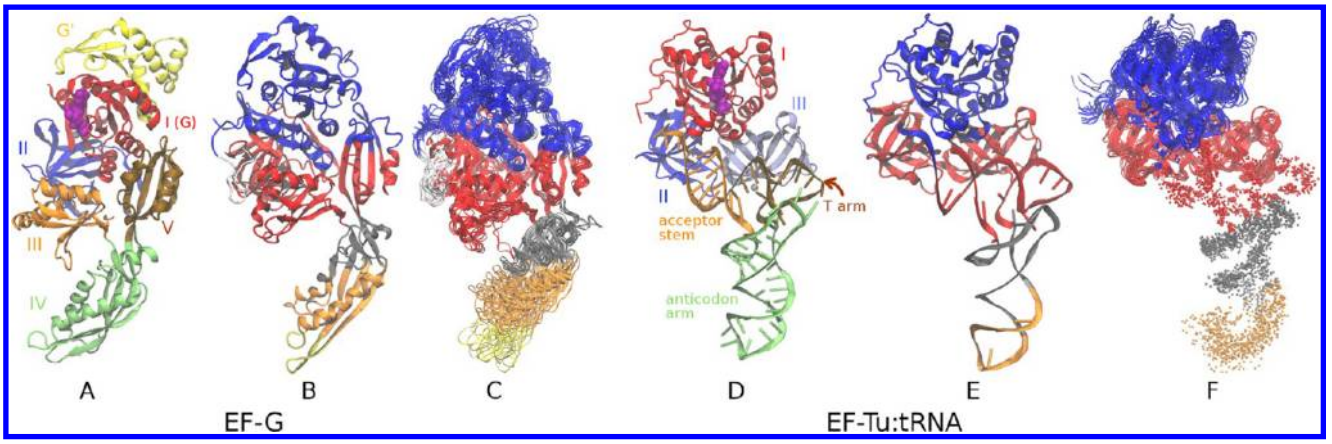


**Figure 9.** Comparison between the structural ("static", panels (A) and (D)) and dynamic domains (panels (B), (C), (E), and (F)) for the EF-G protein and the protein/RNA complex (EF-Tu:tRNA). In panels (C) and (F), several conformations from MD simulations are superimposed, relative to the red domain. The divisions were calculated with the use of HM clustering, for the C$\alpha$ representation in the case of EF-G, and three-atoms-per-residue representation in the case of EF-Tu:tRNA.

EF-Tu:tRNA complex and EF-G, which is in agreement with previous findings.[45]

Another structure examined here, the H38 RNA segment, forms a kink-turn motif that is commonly found in rRNA structures. Kink-turns are very dynamic and often initiate large conformational changes.[56] Again, our analysis emphasized high flexibility of this RNA motif (Figure 10). Apart from the bending

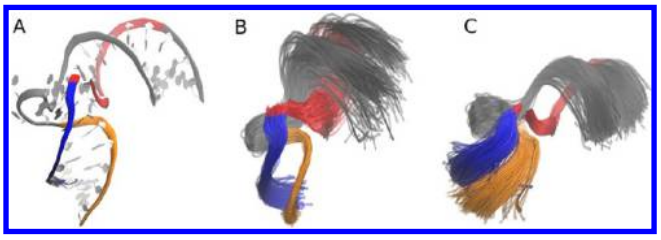


**Figure 10.** (A) Initial structure of the kink-turn region of helix 38 (h38); the backbone is colored according to the optimal division into dynamic domains found by GeoStaS with the NN clustering and one-atom-per-residue representation. Snapshots from the simulation are superimposed with regard to the domain colored (B) orange or (C) red, to emphasize the movements observed in the trajectory.

mode, which changes the angle between the flanking helices, we also observed the breathing motions of helices. In Figure 10 the upper and lower helices are divided lengthwise into two regions, each with a slightly different range of movement. GeoStaS simplified the analysis of these motions and does not require the measurement of the distances and angles between the two strands, as previously described by Réblova et al.[54]

**Software Execution Times.** The time of calculations ranged from a couple of seconds to several minutes on a 4-core Intel Core i7 desktop (maximum of eight parallel threads). The execution time depends on the number of available CPU units (since the code is parallelized), the size of the system (quadratic in relation to number of atoms taken into consideration), and the number of input conformations (linear). For example, for 8335 conformations of a protein of 686 C$\alpha$ atoms, the total calculation time of finding the optimal division into domains (one-atom-per-residue representation) was 1 min, 42 s with the use of a four-core CPU (eight parallel threads). The more conformations the better, but up to a certain limit: sometimes there is no need to take several thousand snapshots from the ~100 ns long simulations if a smaller subset is large and diverse enough. The execution time can be reduced when performing more than one calculation on the same trajectory by loading the previously calculated AMSM as an additional input.

## 4. CONCLUSIONS

The abundance of data produced nowadays by computational techniques creates a need for automatic processing that would reduce the complexity of the data and enable easier detection of functionally important events. Molecular dynamics (MD) simulations provide hundreds of thousands of molecular conformations and reducing the complexity of such conformational ensemble is of utmost importance. Our software, GeoStaS, enables fast identification of molecular fragments that remain internally rigid but differ in mobility, with respect to other parts—we call that procedure a division into dynamic domains. The implemented algorithm searches for the best pairwise superimposition of atomic trajectories, thus finding similarities in their movements. This purely geometrical approach correctly distinguishes between the anticorrelated and correlated motions, and is able to capture rotational correlations. With the use of GeoStaS, the analysis of conformational changes of biomolecules can be faster and easier.

The method presented here is generic; it does not depend on the type of the molecule and order of the conformations. This makes our algorithm easily adaptable to different sources of biomolecular data, as long as it can be described through geometrical structures. GeoStaS reads large trajectories (both in text and binary format) but also handles smaller sets of experimentally resolved structures, such as NMR ensembles. The software has a minimal number of settings and an automatic mode that suggests an optimal solution. In general, analyzing the movement of only the C$\alpha$ or phosphorus (P) atoms suffices to correctly identify the dynamic domains. However, the manual mode is needed because one cannot assess in advance whether one division is more correct than the other: it all depends on how many details one wishes to analyze.

Our method highlights the similarity of movements even between atoms that are far apart in the sequence or atoms that belong to different types of molecules (e.g., the red domain of EF-Tu:tRNA complex in Figure 9 is formed of atoms from both the protein and RNA). Apart from analyzing the conformations from simulations, this type of division can be helpful, e.g., when

creating an ensemble of structures for flexible docking or for fitting models into low-resolution data such as electron microscopy maps. The division into dynamic domains can be also useful when comparing mutational variants of one molecule, highlighting the differences in conformations. In the future, we plan to widen the range of structures that can be analyzed to include any type of molecule, not just proteins or nucleic acids.

## ■ ASSOCIATED CONTENT

### Supporting Information

Figures S1−S6 illustrate the algorithm and additional tests performed on NMR and simulation data. This material is available free of charge via the Internet at http://pubs.acs.org.

## ■ AUTHOR INFORMATION


### Corresponding Author

*E-mail: jrom@icm.edu.pl.



### Funding

This work was supported by the University of Warsaw [BST and ICM G31-4]; Polish Ministry of Science and Higher Education [N N301 245236 and N N301 033339]; and Foundation for Polish Science (Focus program and Team project [TEAM/2009-3/8] cofinanced by European Regional Development Fund operated within Innovative Economy Operational Programme).


### Notes

The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENTS


We are grateful to Katarzyna Kulczycka-Mierzejewska (University of Warsaw), Dr. Kamila Réblová (Academy of Sciences of Czech Republic), and Dr. Lars Skjærven (EMBL Heidelberg) for sharing their MD trajectories with us.


## ■ REFERENCES


(1) Karplus, M.; Kuriyan, J. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 6679−6685.
(2) Henzler-Wildman, K.; Kern, D. *Nature* **2007**, *450*, 964−972.
(3) Karplus, M.; McCammon, J. A. *Nat. Struct. Biol.* **2002**, *9*, 646−652.
(4) Leach, A. R. *Molecular Modelling: Principles and Applications*; Pearson Education/Prentice Hall: Englewood Cliffs, NJ, 2001.
(5) Schlick, T. In *Molecular Modeling and Simulation. An Interdisciplinary Guide*, 1st ed.; Marseden, J., Sirovich, L., Wiggins, S., Antman, S., Eds.; Springer: New York, 2006.
(6) Romanowska, J.; Setny, P.; Trylska, J. *J. Phys. Chem. B* **2008**, *112*, 15227−15243.
(7) Grant, B. J.; Gorfe, A. A.; McCammon, J. A. *Curr. Opin. Struct. Biol.* **2010**, *20*, 142−147.
(8) Vásquez, V.; Sotomayor, M.; Cordero-Morales, J.; Schulten, K.; Perozo, E. *Science* **2008**, *321*, 1210−1214.
(9) Skjaerven, L.; Grant, B. J.; Muga, A.; Teigen, K.; McCammon, J. A.; Reuter, N.; Martinez, A. *PLoS Comput. Biol.* **2011**, *7*, e1002004.
(10) Jensen, M. O.; Tajkhorshid, E.; Schulten, K. *Structure* **2001**, *9*, 1083−1093.
(11) Rao, F.; Karplus, M. *Proc. Natl. Acad. Sci. U.S.A.* **2010**, *107*, 9152−9157.
(12) Sorin, E. J.; Pande, V. S. *Biophys. J.* **2005**, *88*, 2472−2493.
(13) Andrec, M.; Felts, A. K.; Gallicchio, E.; Levy, R. M. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 6801−6806.
(14) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. *Science* **2011**, *334*, 517−520.
(15) Klenin, K.; Strodel, B.; Wales, D. J.; Wenzel, W. *Biochim. Biophys. Acta* **2011**, *1814*, 977−1000.
(16) Christen, M.; van Gunsteren, W. F. *J. Computat. Chem.* **2008**, *29*, 157−166.
(17) Zwier, M. C.; Chong, L. T. *Curr. Opin. Pharmacol.* **2010**, *10*, 745−752.
(18) Shao, J.; Tanner, S. W.; Thompson, N.; Cheatham, T. E. *J. Chem. Theory Comput.* **2007**, *3*, 2312−2334.
(19) Amadei, A.; Linssen, A. B. M.; Berendsen, H. J. C. *Proteins: Struct., Funct., Bioinf.* **1993**, *17*, 412−425.
(20) Wriggers, W.; Schulten, K. *Proteins: Struct., Funct., Bioinf.* **1997**, *29*, 1−14.
(21) Hinsen, K. *Proteins: Struct., Funct., Bioinf.* **1998**, *33*, 417−429.
(22) Hinsen, K.; Thomas, A.; Field, M. J. *Proteins: Struct., Funct., Bioinf.* **1999**, *34*, 369−382.
(23) Snyder, D. A.; Montelione, G. T. *Proteins: Struct., Funct., Bioinf.* **2005**, *59*, 673−686.
(24) Qi, G.; Lee, R.; Hayward, S. *Bioinformatics* **2005**, *21*, 2832−2838.
(25) Poornam, G. P.; Matsumoto, A.; Ishida, H.; Hayward, S. *Proteins: Struct., Funct., Bioinf.* **2009**, *76*, 201−212.
(26) Grant, B. J.; Rodrigues, A. P. C.; ElSawy, K. M.; McCammon, J. A.; Caves, L. S. D. *Bioinformatics* **2006**, *22*, 2695−2696.
(27) Aleksiev, T.; Potestio, R.; Pontiggia, F.; Cozzini, S.; Micheletti, C. *Bioinformatics* **2009**, *25*, 2743−2744.
(28) Kirchner, D. K.; Guntert, P. *BMC Bioinformatics* **2011**, *12*, 170−181.
(29) Roccatano, D.; Mark, a. E.; Hayward, S. *J. Mol. Biol.* **2001**, *310*, 1039−1053.
(30) Tozzini, V.; Trylska, J.; Chang, C.; Mccammon, J. A. *J. Struct. Biol.* **2007**, *157*, 606−615.
(31) Humphrey, W.; Dalke, A.; Schulten, K. *J. Mol. Graphics* **1996**, *14*, 33−38.
(32) Hayward, S. In *Computational Biochemistry and Biophysics*, 1st ed.; Becker, O. M., MacKerell, Alexander, D. J., Roux, B., Watanabe, M., Eds.; Taylor & Francis: New York, 2001; pp 153−168.
(33) Potestio, R.; Pontiggia, F.; Micheletti, C. *Biophys. J.* **2009**, *96*, 4993−5002.
(34) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. *Nucleic Acids Res.* **2000**, *28*, 235−242.
(35) Hamilton, S. W. R. *Elements of Quaternions (Google eBook)*; Longmans, Green, & Co., 1866.
(36) Kuipers, J. B. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*; Princeton University Press: Princeton, NJ, 2002.
(37) Kneller, G. R.; Calligari, P. *Acta Crystallogr., Sect. D: Biol. Crystallogr.* **2006**, *62*, 302−311.
(38) Keller, B.; Daura, X.; van Gunsteren, W. F. *J. Chem. Phys.* **2010**, *132*, 074110−074116.
(39) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. *J. Comput. Chem.* **2005**, *26*, 1781−1802.
(40) Brooks, B. R.; Brooks, C. L., III; Mackerell, A. D., Jr.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M. *J. Comput. Chem.* **2009**, *30*, 1545−1614.
(41) Thompson, K. C.; Jordan, M. J. T.; Collins, M. A. *J. Chem. Phys.* **1998**, *108*, 564−578.
(42) Campos-Olivas, R.; Newman, J. L.; Summers, M. F. *J. Mol. Biol.* **2000**, *296*, 633−649.
(43) Lukavsky, P. J.; Kim, I.; Otto, G. A.; Puglisi, J. D. *Nat. Struct. Biol.* **2003**, *10*, 1033−1038.
(44) Ramos, A.; Grünert, S.; Adams, J.; Micklem, D. R.; Proctor, M. R.; Freund, S.; Bycroft, M.; St. Johnston, D.; Varani, G. *EMBO J.* **2000**, *19*, 997−1009.
(45) Kulczycka, K.; Długosz, M.; Trylska, J. *Eur. Biophys. J.* **2011**, *40*, 289−303.
(46) Skjaerven, L.; Muga, A.; Reuter, N.; Martinez, A. *Proteins: Struct., Funct., Bioinf.* **2012**, in press.
(47) Horwich, A. L.; Fenton, W. A. *Q. Rev. Biophys.* **2009**, *42*, 83−116.

(48) Xu, Z.; Horwich, A. L.; Sigler, P. B. *Nature* **1997**, *388*, 741−750.
(49) Ma, J.; Sigler, P. B.; Xu, Z.; Karplus, M. *J. Mol. Biol.* **2000**, *302*, 303−313.
(50) de Groot, B. L.; Vriend, G.; Berendsen, H. J. *J. Mol. Biol.* **1999**, *286*, 1241−1249.
(51) Nissen, P.; Kjeldgaard, M.; Thirup, S.; Polekhina, G.; Reshetnikova, L.; Clark, B. F.; Nyborg, J. *Science* **1995**, *270*, 1464−1472.
(52) Abel, K.; Yoder, M. D.; Hilgenfeld, R.; Jurnak, F. *Structure* **1996**, *4*, 1153−1159.
(53) Polekhina, G.; Thirup, S.; Kjeldgaard, M.; Nissen, P.; Lippmann, C.; Nyborg, J. *Structure* **1996**, *4*, 1141−1151.
(54) Réblová, K.; Rázga, F.; Li, W.; Gao, H.; Frank, J.; Šponer, J. *Nucleic Acids Res.* **2010**, *38*, 1325−1340.
(55) Laurberg, M.; Kristensen, O.; Martemyanov, K.; Gudkov, A. T.; Nagaev, I.; Hughes, D.; Liljas, A. *J. Mol. Biol.* **2000**, *303*, 593−603.
(56) Rázga, F.; Zacharias, M.; Réblová, K.; Koča, J.; Šponer, J. *Structure* **2006**, *14*, 825−835.
(57) R Development Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2011 (ISBN 3-900051-07-0).