JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

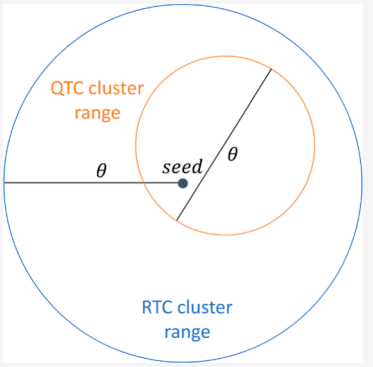
On Quality Thresholds for the Clustering of Molecular Structures

Xavier Daura* and Oscar Conchillo-Solé

Cite This: J. Chem. Inf. Model. 2022, 62, 5738−5745 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: It has been recently suggested that diametral (so-called quality) similarity thresholds are superior to radial ones for the clustering of molecular three-dimensional structures (González-Alemán et al., 2020). The argument has been made for two clustering algorithms available in various software packages for the analysis of molecular structures from ensembles generated by computer simulations, attributed to Daura et al. (1999) (radial threshold) and Heyer et al. (1999) (diametral threshold). Here, we compare these two algorithms using the root-mean-squared difference (rmsd) between the Cartesian coordinates of selected atoms as pairwise similarity metric. We discuss formally the relation between these two methods and illustrate their behavior with two examples, a set of points in two dimensions and the coordinates of the tau polypeptide along a trajectory extracted from a replica-exchange molecular-dynamics simulation (Shea and Levine, 2016). We show that the two methods produce equally sized clusters as long as adequate choices are made for the respective thresholds. The real issue is not whether the threshold is radial or diametral but how to choose in either case a threshold value that is physically meaningful. We will argue that, when clustering molecular structures with the rmsd as a metric, the simplest best guess for a threshold is actually radial in nature.



INTRODUCTION

Over 2 decades ago, Heyer et al.[1] developed an algorithm to cluster open reading frames (ORFs) based on their expression levels, using what they came to call jackknife correlation as pairwise similarity metric. The focus of the algorithm was to find large clusters that had a "quality" guarantee, that is, a minimum jackknife correlation between any two ORFs belonging to the same cluster. In other words, clusters would be guaranteed to have a maximum diameter, determined by a correlation threshold, ensuring the transitive property for the relation 'correlation > threshold' between element pairs in a cluster (i.e., if correlation(a, b) > threshold and correlation(b, c) > threshold, then correlation(a, c) > threshold, for any a, b, c belonging to the cluster). The algorithm was named quality cluster (QT_Clust) and it has since been used for the clustering of several other types of data, including molecular structures.[2] In this study, we will refer to this algorithm as QTC.

The same year, Daura et al.[3,4] introduced an algorithm to cluster molecular structures using the root-mean-squared difference (rmsd) between the Cartesian coordinates of selected atoms as pairwise similarity metric. The algorithm was meant to favor the most populated cluster and was radial in nature, that is, the rmsd threshold was applied from a configuration taken as a cluster reference, meaning transitivity was not ensured for the relation 'rmsd < threshold'. In this study, we will refer to this algorithm as RTC. The two algorithms are described under the Computational Details section and analyzed formally under the Results and Discussion section.

When applied to molecular structures using the rmsd as a metric, both QTC and RTC scan a precalculated rmsd matrix in search for the molecular configuration with the largest number of neighbors satisfying the threshold, in an iterative process that outputs a new cluster at each step and ends when there are no further configurations to cluster. The difference between the two algorithms lies in the nature of the threshold (diametral or radial) and the procedure to count the neighbors in. Recently, González-Alemán et al.[2] pointed out that, because of their similarity, these two algorithms were often confused in various software implementations commonly used in the field, misleading their users. They also evaluated the performance of the two algorithms by analyzing a trajectory of the tau polypeptide extracted from a replica-exchange molecular-dynamics simulation previously published by Shea and Levine.[5] In doing so, they used the same rmsd value for the thresholds applied in the two algorithms. They concluded that, due to its lack of a quality threshold, the RTC algorithm (referred to as Daura's algorithm in the paper) tends to cluster unrelated configurations together and gave examples in which different QTC clusters were found as composing a single RTC cluster.

Received: August 25, 2022
Published: October 20, 2022

Clearly, the results observed by González-Alemán et al. had little to do with the quality of the thresholds and much to do with using the same rmsd value for a radial and a diametral threshold. While the relation 'rmsd < threshold' is not transitive for the set of configurations conforming a cluster generated by the RTC algorithm, the alternative relation 'rmsd < diameter' is. This leads to the following question: can we generate clusters with a predetermined maximum diameter using the RTC algorithm? In other words, can we choose the radial threshold in the RTC algorithm in such a way that the maximum diameter of a cluster will be equal to the threshold we would use with the QTC algorithm? The answer is of course yes, if one would actually wish to do so.

In the following sections, we will present formally and analyze in detail the characteristics of the RTC and QTC methods. Although the two methods have been available and heavily used since over 2 decades, a detailed analysis of their properties has not been published. We will show that it is indeed possible to obtain equally sized clusters with the two algorithms and will argue that this is in fact of little importance because, first, the threshold is an arbitrary quantity that may have different "ideal" values depending on the objective of the analysis and, second, for the purposes discussed here the simplest physically based guide to decide on the value of the threshold is in fact radial in nature rather than diametral.

COMPUTATIONAL DETAILS

Clustering Algorithms. The two algorithms, QTC[1] and RTC,[3,4] require as input the matrix of rmsds between all pairs of configurations.

In its mth iteration, the QTC algorithm scans the matrix in search for the configuration with the largest number of neighbors in order to generate the mth cluster. Specifically, each configuration not clustered in the previous $m - 1$ iterations (we shall refer to them as the available configurations) will be considered both as a seed of a tentative cluster and as a potential neighbor of all other seeds. We use the term tentative cluster to refer to a seed and its neighbors before the neighborhood sizes of all seeds are compared to select the actual cluster. For each seed, its neighbors will be determined as follows: From all other available configurations, the one that upon its addition extends the diameter of the cluster the least, while fulfilling the condition that the diameter must remain smaller than the threshold, is taken as the next neighbor and included in the seed's tentative cluster. This process is repeated until no remaining available configuration fulfills the threshold, at which point the tentative cluster for that seed is complete. Note that within an iteration all available configurations are tested as potential neighbors of each one of the available seeds. Once the tentative clusters for all available seeds have been obtained, the one with the largest number of elements is promoted to constitute the mth cluster, and all elements of that cluster are removed from the pool of available configurations, thus finalizing the mth iteration. The algorithm is stopped when there are no more configurations available or new clusters fall below a preset minimum number of elements.

The RTC algorithm differs from the QTC one in the way the neighbors of a seed are determined at each iteration: All available configurations at a distance from the seed smaller than the threshold are taken as elements of the seed's tentative cluster. Thus, the RTC algorithm avoids the double loop per seed that characterizes the QTC algorithm —to find the next element of the tentative cluster (inner loop over available configurations) until no other configurations fulfilling the diameter threshold are available (outer loop).

The clusterings were performed with inhouse software reproducing exactly the algorithms described here. Results using established software implementations (available free of charge) on the tau-polypeptide example are provided as the Supporting Information (SI) for comparison. For the RTC case, results found in the SI were obtained using the McLachlan algorithm[6] as implemented in ProFit v3.3 (http://www.bioinf.org.uk/software/profit/) for the rmsd calculation and the RTC algorithm as implemented in HADDOCK v2.0[7] (cluster_struc, https://www.bonvinlab.org/software/haddock2.2/) for the clustering. For the QTC case, results found in the SI were obtained using the implementation published by González-Alemán et al.[2] (https://github.com/rglez/QT). The results are exactly the same, with small differences in the QTC case due to implementation details that are explained in the SI document and conform in both cases with the QTC algorithm.

MD Simulation Data. The trajectory of the tau polypeptide was downloaded from https://github.com/LQCT/BitQT/blob/master/examples/aligned_original_tau_6K.dcd, together with the reference PDB file https://github.com/LQCT/BitQT/blob/master/examples/aligned_tau.pdb. It corresponds to the exact same trajectory used by González-Alemán et al.[2] in their comparison of the QTC and RTC clustering algorithms. The trajectory contains 6001 configurations of the polypeptide. We used the backbone N, H, C_α, C, and O atoms of residues Lys_2 to Asp_{11}, that is, 50 atoms in total, for least-squares fitting and rmsd calculation. The two terminal residues, Gly_1 and Leu_{12} and their capping groups, were left out because they are relatively free to rotate and would introduce unnecessary noise in the clustering of the rest of the structure. Likewise, we excluded the side chains because one generally focuses on the backbone to define a fold and side chains would only introduce noise. This selection of atoms is clearly different from that used by González-Alemán et al.[2] (all atoms), but this is irrelevant for the questions addressed here.

To generate points for the example in two dimensions (not that this is important), we simply took the x and y coordinates of the backbone N atom of Lys_2 after least-squares fitting of all configurations to configuration number 2910. A subset of 1501 elements was then constructed by selecting 1 element every 4, starting with element 1.

RESULTS AND DISCUSSION

Theoretical Framework and Properties. We note that throughout this article we use the term diameter in its generalized form, that is, as the largest distance between any two points on the boundary of a closed geometric figure (in this case a cluster). Likewise, we use the term sphere as a shorthand for $(n - 1)$-sphere, defined as the $(n - 1)$-dimensional boundary of an (n-dimensional) n-ball.

Let $S_m = \{\mathbf{x}_i \equiv (x_{i,1}, ..., x_{i,n}) \in \mathbb{R}^n : i \in I_m\}$ be a set of Euclidean vectors in a Cartesian frame (for convenience, we shall also refer to $\mathbf{x}_i$ as a point in that frame) representing the N_m configurations of the molecule that are available for clustering at the mth iteration of the RTC or QTC algorithm, where n is the number of coordinates that will be used for the rmsd calculation, $I_m = \{i \in \mathbb{N} : 1 \le i \le N_1, i \notin J_m\}$ is the set of indices of the elements of S_1 that are available for clustering at the mth iteration and J_m, with $J_1 = \emptyset$ and

https://doi.org/10.1021/acs.jcim.2c01079
J. Chem. Inf. Model. 2022, 62, 5738−5745

$J_{m>1} = \{j \in \mathbb{N} : 1 \le j \le N_1,\ \mathbf{x}_j \in C_l,\ 1 \le l \le m-1\}$, is the set of indices of the elements of S_1 that have been already clustered in previous iterations, where C_l stands for the cluster set defined in iteration l (see below).

Let $\mathbf{x}_k \in S_m$ be the seed for a tentative cluster of elements of S_m and $A_{m,k}(\theta) = \{\mathbf{x}_i \in S_m : \mathrm{rmsd}_{ki} < \theta\}$ the set of elements within a rmsd-threshold $\theta \in R_{>0}$ from $\mathbf{x}_k$. Note that $\mathrm{rmsd}_{ki} = \|\mathbf{x}_i - \mathbf{x}_k\|(n_a)^{-1/2}$, where n_a is the number of atoms involved in the rmsd calculation (in principle, $n = 3 \times n_a$). Then, we define $A_m(\theta) = \{A_{m,k}(\theta) : k \in I_m\}$ as the collection of such sets for all available seeds and $B_m(\theta) = \{|A_{m,k}(\theta)| : k \in I_m\}$, where $|A_{m,k}(\theta)|$ stands for the cardinality of $A_{m,k}(\theta)$, as the collection of corresponding set sizes.

In the RTC algorithm, $A_{m,k}(\theta)$ is the tentative cluster "proposed" by seed $\mathbf{x}_k$, which shall be then compared to the tentative clusters "proposed" by all other seeds. Thus, we define $D_m(\theta) = \{A_{m,k}(\theta) \in A_m(\theta) : |A_{m,k}(\theta)| = \max(B_m(\theta))\}$ as the collection of sets with the largest number of elements and $E_m(\theta) = \{k \in I_m : A_{m,k}(\theta) \in D_m(\theta)\}$ as the collection of corresponding indices. The mth cluster (output of the mth iteration of the algorithm) is then defined as $C_m(\theta) = \{A_{m,k}(\theta) \in D_m(\theta) : k = f(E_m(\theta))\}$, where f is a function that returns one element from a set, typically the function min(), in which case $C_m(\theta)$ would be the set with the lowest index from those with the largest number of elements.

To impose the condition that the diameter of $C_m(\theta)$ is smaller than θ, as done in the QTC algorithm, we need to define a new set $F_{m,k}(\theta) = p(A_{m,k}(\theta))$, where p is an element-selection procedure, that is, $F_{m,k}(\theta) \subset A_{m,k}(\theta)$, such that $\mathrm{rmsd}_{ij} < \theta, \forall\ \mathbf{x}_i, \mathbf{x}_j \in F_{m,k}(\theta)$. $F_{m,k}(\theta)$ is, in this algorithmic context, the tentative cluster "proposed" by seed $\mathbf{x}_k$, which shall be compared to the tentative clusters "proposed" by all other seeds. Thus, as done for the tentative clusters in the RTC case, we define $F_m(\theta) = \{F_{m,k}(\theta) : k \in I_m\}$ as the collection of such sets for all available seeds and redefine $B_m(\theta) = \{|F_{m,k}(\theta)| : k \in I_m\}$ as the collection of corresponding set sizes. Accordingly, we redefine $D_m(\theta) = \{F_{m,k}(\theta) \in F_m(\theta) : |F_{m,k}(\theta)| = \max(B_m(\theta))\}$ and $E_m(\theta) = \{k \in I_m : F_{m,k}(\theta) \in D_m(\theta)\}$. The mth cluster is then defined as $C_m(\theta) = \{F_{m,k}(\theta) \in D_m(\theta) : k = f(E_m(\theta))\}$.

Note that the latter cluster definition is not specific for the QTC algorithm but general for a group of diameter-based algorithms. This is because, as it is defined, the procedure p is not unique. That is, the condition $\mathrm{rmsd}_{ij} < \theta, \forall\ \mathbf{x}_i, \mathbf{x}_j \in F_{m,k}(\theta)$, $F_{m,k}(\theta) \subset A_{m,k}(\theta)$ can be satisfied by different selection procedures p, leading to different subsets $F_{m,k}(\theta)$ of $A_{m,k}(\theta)$. For example, the tentative cluster $F_{m,k}(\theta)$ can be grown starting from its seed following the procedure by Heyer et al.[1] and described under the Computational Details section (the procedure p implemented in the QTC algorithm), or could be grown following a specific sequence of configurations (e.g., time sequence), that is, testing at each step if the next configuration in sequence satisfies the diametral threshold instead of searching for the configuration that minimizes the increase in the cluster diameter (note that this would produce different $F_{m,k}(\theta)$ subsets for different configuration sequences). Another variant of p could be searching for the subset $F_{m,k}(\theta)$ with the largest number of elements, and so forth.

While the RTC and QTC algorithms are generally presented as invariant to permutation (referring to the order in which the algorithm evaluates the seeds or the order in which the configurations are tested for inclusion in a seed's tentative cluster), they are strictly not. This is because the collection $D_m(\theta)$ defined above may contain more than one set. Indeed, when using these algorithms in practice, at any given iteration it is relatively common to see two or more seeds tie as the ones generating the tentative clusters with the largest number of elements (this is precisely why f is needed in the definition of $C_m(\theta)$), in which case both algorithms become configuration-order sensitive (for a given function f). An illustrative example of many candidate seeds forming tentative clusters of the same size in a given iteration is given in the Supporting Information (see the comment on cluster 2 in the QTC case). It is true, however, that ties tend to occur between seeds that are very close in space, thus having a relatively small impact on the clustering.

We shall use the term cluster shape to refer to the convex hull of the set of points that belong to the cluster. Thus, the cluster shape with maximum volume is for either algorithm a sphere, of radius θ for RTC clusters and $\theta/2$ for QTC ones. Note that we can define three types of geometric centers for both RTC and QTC clusters. The first type is the center of the neighbor-search volume (a sphere) and corresponds to the position of the seed (as we have seen above, both algorithms search within $A_{m,k}(\theta)$). For this reason, it is customary to refer to the seed, particularly in the RTC case, as the central element of the cluster, but this is, as we shall see, misleading if we give it a spatial significance. The second type is the geometric center of the cluster shape, which will only coincide with the first one if the cluster is spherical (RTC case) or spherical and centered on the seed (QTC case). And the third type is the geometric center of the cluster elements, which will only coincide with the second one if the spatial distribution of points in the cluster is homogeneous (which is rare). Thus, it is in fact common for the seeds of even RTC clusters to be relatively distant from the geometric center of the cluster shape and/or the geometric center of the cluster elements.

The seed of a QTC cluster tends in fact to be close to the cluster's boundary. This is due to the specific procedure p that selects the elements of $F_{m,k}(\theta)$ from $A_{m,k}(\theta)$. As described under the Computational Details section, at each step in the process of recruiting new elements for the tentative cluster $F_{m,k}(\theta)$, the point that minimizes the increase in diameter while fulfilling the diametral threshold θ is selected. Thus, the first steps of the procedure are highly determined by the local distribution of points around the seed, which is never equal in all directions. This will introduce an early bias or dominant direction and sense for the cluster's growth, which may be more or less prominent depending on the exact spatial distribution of points. In cases in which the distribution induces a marked directionality (which could be relatively frequent as we shall see in the 2D example below) and assuming a spherical cluster, the cluster will tend to grow in eccentric spherical layers away from the seed, leaving the seed at or very close to the cluster's boundary. Note that it is the local distribution of points around the seed that primarily determines the direction and sense of growth of $F_{m,k}(\theta)$, rather than the global spatial distribution of points in $A_{m,k}(\theta)$. Therefore, $F_{m,k}(\theta)$ is not necessarily the subset of $A_{m,k}(\theta)$ with the highest cardinality. Another consequence of this is that, unlike for RTC clusters, for QTC clusters the relation $|C_m(\theta)| > |C_{m+1}(\theta)|$ does not need to hold: the direction of growth of the tentative cluster around a given seed $\mathbf{x}_k$ may in some cases be less optimal for set S_m than for set S_{m+1}, so that we may have $|F_{m,k}(\theta)| < |F_{m+1,k}(\theta)|$, which can eventually lead to a situation

https://doi.org/10.1021/acs.jcim.2c01079
J. Chem. Inf. Model. 2022, 62, 5738−5745

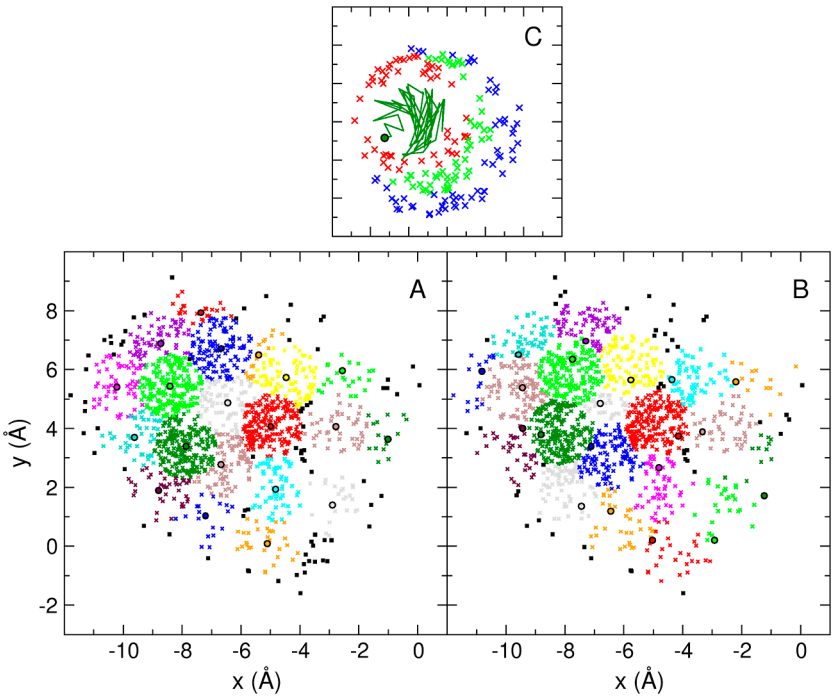


Figure 1. Representation of the first 20 clusters of 1501 points in 2D. (**A**) RTC algorithm, θ = 1.1 Å. (**B**) QTC algorithm, θ = 2.2 Å. The color sequence is the same in both panels, starting with dark green for the first cluster. The seed elements of the clusters are indicated with a black circle (with the area matching the cluster color). Points belonging to clusters with an index higher than 20 are indicated as black squares. (**C**) Detail of the process of generation of cluster 1 from panel B (dark green): to illustrate this process, the points are shown in four different colors following their sequence of inclusion in the cluster. Thus, initial growth from the seed (dark-green circle) is indicated as a dark-green trajectory. Elements in red, light green, and blue correspond to successive phases in the growth of the cluster, in this order.

in which $|F_{m,k}(\theta)| < |F_{m,j}(\theta)|$, where $F_{m,j}(\theta) = C_m(\theta)$, and $|F_{m+1,k}(\theta)| > |F_{m,j}(\theta)|$, where $F_{m+1,k}(\theta) = C_{m+1}(\theta)$. In practice, however, the fact that $C_m(\theta)$ is selected from a collection $F_m(\theta)$ of overlapping tentative clusters $F_{m,k}(\theta)$ makes this potential inversion of cluster-size order very infrequent and we have only observed it in two cases in the examples below, for very small (irrelevant) clusters.

To adequately compare the RTC and QTC algorithms in practical applications, one needs to keep in mind that θ is a radial threshold (i.e., θ_r) in the RTC algorithm and a diametral threshold (i.e., θ_d) in the QTC algorithm. If the spatial distribution of points is such that the diameters of the RTC clusters are approximately equal to two times the threshold θ_r in at least one direction, the extreme case being spherical clusters, one should use $\theta_r = \theta/2$ as the RTC threshold and $\theta_d = \theta$ as the QTC threshold for comparable results. However, as we shall see in the tau-polypeptide example, the distribution of points has rarely these characteristics when working with n-dimensional data representing molecular configurations from computer simulation. First, the region of this n-dimensional space corresponding to a conformer of the molecule (where the term conformer may be taken as a very well-defined structure or a broader structural state, depending on the purpose of the clustering) will generally have an irregular shape and its immediate surrounding may be void in many directions, so that even if the algorithm tries to mix in points corresponding to neighbor conformers (when the threshold is too big) in many spatial directions there might be simply nothing to mix in. Second, the density of points in this n-dimensional space is typically reduced in clustering exercises due to the selection of only one structure every so many simulation steps, which has the effect of blurring the underlying cluster topology (if such should physically exist). Therefore, in order to have equally sized clusters, the relation between the rmsd thresholds θ_r (RTC) and θ_d (QTC) will be in practice $\theta_d/2 \leq \theta_r < \theta_d$.

A more important question, however, is how to choose the threshold. Typically, the option of choice in the literature is trial and error: both algorithms, particularly RTC, are sufficiently fast that one can actually run them several times with different θ values, until the outcome satisfies any chosen criteria. The criterion is often visual, that is, the structures in a cluster "look the same".[2] While this may fit the purpose in some types of studies, it is actually a weak criterion from a physical standpoint. As a general rule, before clustering data points in an n-dimensional space, one may want to query this space to gather information on the distribution of data points in it. This is also what makes physical sense in this case because the space in question is the molecule's configuration space (reduced to the number of coordinates used for the rmsd calculation and the given ensemble sample). As will be shown for the tau-polypeptide example below, the simplest effective way to query this space is by calculating the distribution of rmsd values. This can be done for the full (half) rmsd matrix, but the mixing of underlying distributions generally reduces its informative value. Instead, we propose that a first tentative clustering may be performed to calculate the rmsd distribution for each of the seed elements of the most populated clusters (as representatives of the high-density regions of interest), that is, taking the respective full columns (or rows) from the rmsd matrix. In many cases, these distributions will show a first region of relatively high probability density, followed by a deep and then a second, larger increase in probability density. This deep in the probability density corresponds to the end of the

https://doi.org/10.1021/acs.jcim.2c01079
J. Chem. Inf. Model. 2022, 62, 5738−5745

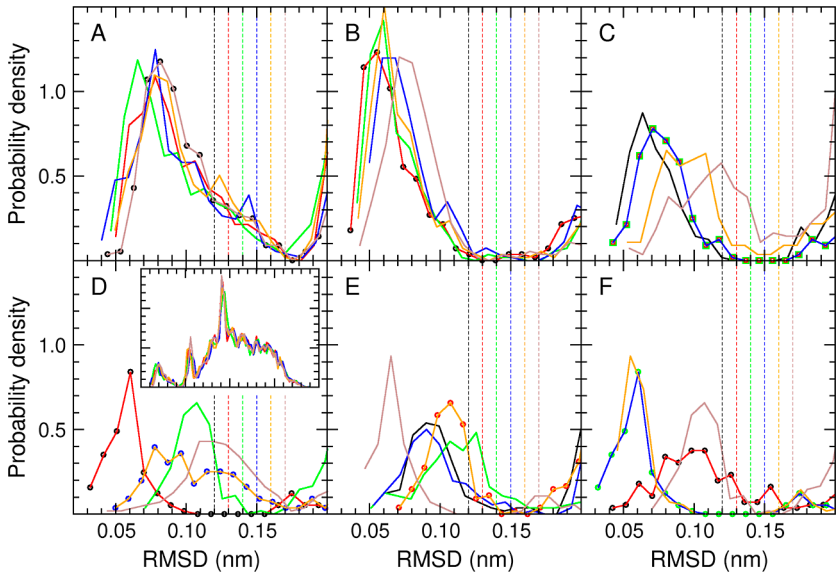


Figure 2. rmsd distributions for the seeds of the six most populated clusters, after RTC clustering with six different thresholds. Each curve corresponds to the distribution of the rmsds between the given seed and all other 6000 configurations. The distributions are cut at 0.20 nm for clarity (the inset in panel D shows the full distributions from panel A as example). Panels **A** to **F** correspond to the distributions for the seeds of clusters 1 to 6, respectively. Each panel contains the distributions for six seeds, resulting from RTC clusterings with θ_r values of 0.12 (black), 0.13 (red), 0.14 (green), 0.15 (blue), 0.16 (orange), and 0.17 nm (brown). The dashed vertical lines show the positions of the thresholds (with colors matching the corresponding distributions). When two distributions overlap (i.e., clusterings with different thresholds produce the same seeds for the given cluster number), the overlapping curves are replaced by circles of the corresponding color.

first layer of configurations around the seed and may therefore be interpreted as a (spherical) region of conformational transition. After comparing the distributions for the seed elements of the main clusters, this information can be used to assess the threshold for a second, final clustering. Note that this information is radial in nature, and it therefore leads to a value for θ_r rather than θ_d.

In the next subsections, we illustrate some of the properties discussed above of the RTC and QTC algorithms with two examples.

Example Case in 2D. Figure 1 shows the clustering of the set of 1501 points in two dimensions (see the Computational Details section) performed with the RTC (panel A) and QTC (panels B and C) algorithms. This is a good example of a major limitation of fixed-size clustering algorithms: if the data has no particular underlying structure or the threshold is completely inadequate, these algorithms will still partition the data according to the chosen threshold. The question of how to choose a threshold is, therefore, of particular significance, even if we will ignore it in this first example.

Note that, as mentioned under the Theoretical Framework and Properties section, for densely populated spaces with few void regions, a selection of thresholds such that $\theta_r = \theta_d/2$, where θ_r is the threshold used with the RTC algorithm and θ_d is the threshold used with the QTC algorithm, produces equally sized clusters in the two cases. As also mentioned, while the seeds of the RTC clusters tend to be closer to the centroids of their cluster shapes, the seeds of the QTC clusters tend to be closer to the clusters' boundaries.

Panel C illustrates the generation process of QTC clusters, using the first cluster from panel B (dark green) as an example. The initial steps are shown as a trajectory starting from the seed (dark-green circle). Consecutive phases in the growing of the cluster are illustrated with elements in red, light green, and blue, in this order. The directionality of the growth, away from the seed in eccentric circular layers, can be clearly observed.

Clustering of Structures from a MD Trajectory. Following the strategy described above to infer a physically meaningful threshold, we first used the RTC algorithm to perform a clustering of the 6001 configurations of the tau-polypeptide, in order to identify points in the 150-dimensional space (50 atoms × 3 coordinates) that are in regions of high density. In this case, we chose the seed elements of the six most populated clusters to then examine the distribution of rmsds between each of these elements and the other 6000 configurations. In order to see how the distributions may vary depending on the threshold used for this initial clustering, we chose six evenly spaced thresholds between 0.12 and 0.17 nm. The results are shown in Figure 2.

In all distributions, an initial region corresponding to a first layer of configurations around the seed can be distinguished, after which the probability density goes down to zero or close to it. The distributions for the seeds of the first, most populated clusters tend to be less sensitive to the threshold used for the clustering, as expected. This is not so much because the clusters are more populated but because they are generated first, that is, the following clusters are affected by which configurations have or have not been already taken by the previous ones. Although the rmsd value at which the probability density reaches zero differs for the different distributions, 0.17 nm stands out as a possible consensus threshold: it is a point at which the probability density either reaches zero (notably for the seeds of cluster 1) or has not yet recovered significantly from zero.

Based on these observations, we focused on the clusterings performed with θ_r values of 0.12 nm, that is, the lowest value that seems adequate for some of the distributions shown in Figure 2, and 0.17 nm. To define the corresponding θ_d thresholds for the QTC algorithm, we looked at the diameter

https://doi.org/10.1021/acs.jcim.2c01079
J. Chem. Inf. Model. 2022, 62, 5738−5745

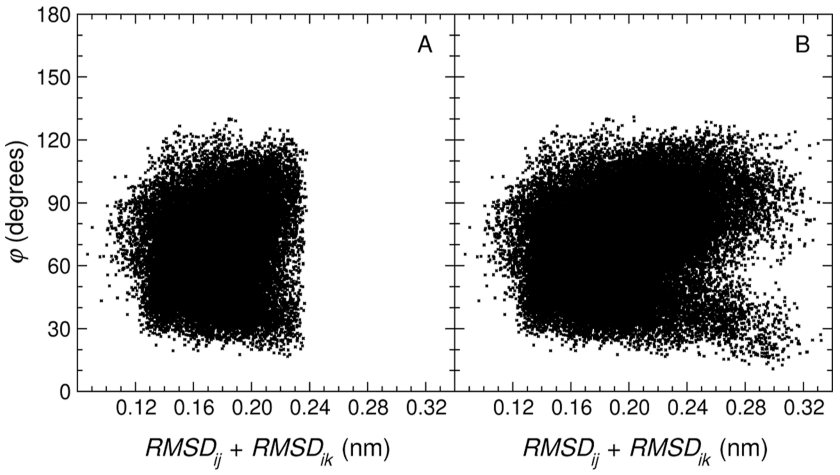


Figure 3. Relation between the distance to the seed and corresponding angle for every pair of elements of a cluster (excluding the seed). Specifically, $\text{rmsd}_{ij} + \text{rmsd}_{ik}$, for all $\mathbf{x}_j, \mathbf{x}_k \in C_1(\theta_r)$, $j, k \neq i$, where $\mathbf{x}_i$ is the seed of $C_1(\theta_r)$ and θ_r has the values 0.12 nm (**A**) and 0.17 nm (**B**), against the angle φ between the vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{ik}$, where $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$.

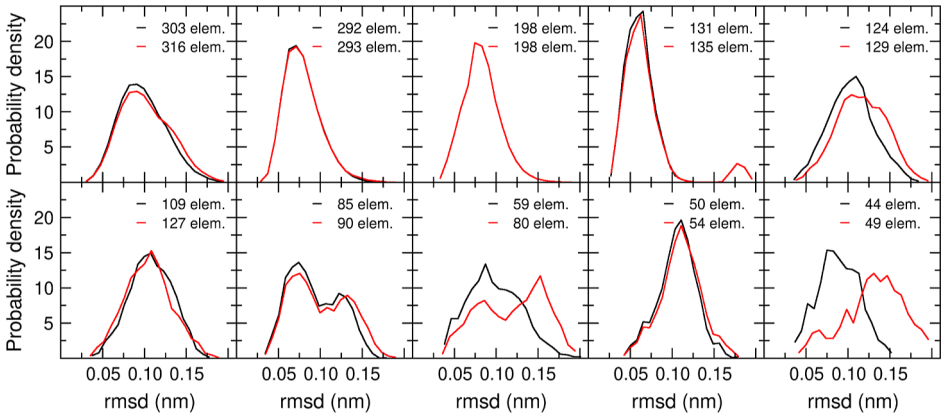


Figure 4. rmsd distributions for the first 10 clusters, using the RTC algorithm with $\theta_r = 0.12$ nm (black) and the QTC algorithm with $\theta_d = 0.20$ nm (red). The number of elements in the cluster is for each case indicated.

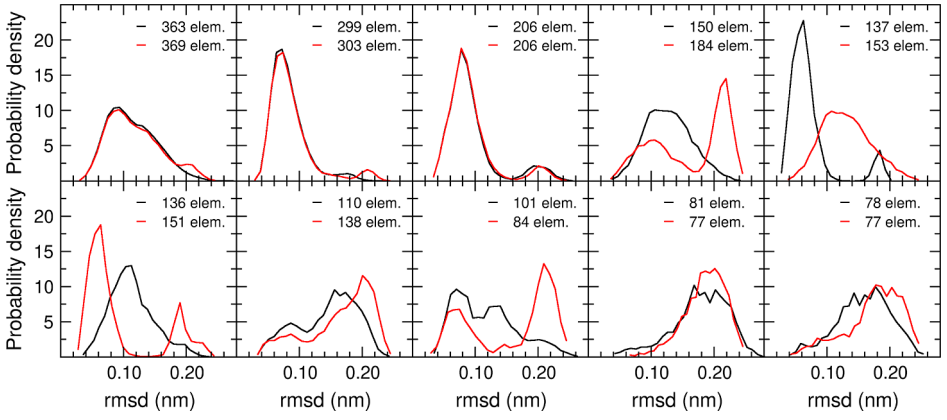


Figure 5. rmsd distributions for the first 10 clusters, using the RTC algorithm with $\theta_r = 0.17$ nm (black) and the QTC algorithm with $\theta_d = 0.25$ nm (red). The number of elements in the cluster is for each case indicated.

of cluster 1 in each of these two RTC clusterings. Note that cluster 1 is our best guide because its diameter is not conditioned by previous clusters. The diameter was 0.199 nm for $C_1^{\text{RTC}}(0.12)$ and 0.278 nm for $C_1^{\text{RTC}}(0.17)$. With these reference values and taking into account that the probability density for a rmsd distance above 0.25 nm in $C_1^{\text{RTC}}(0.17)$ is very small (only 17 elements at distances between 0.25 and 0.278 nm), we decided to choose θ_d values of 0.20 and 0.25 nm for the QTC algorithm.

Note that, as already discussed, the diameter of an RTC cluster should be generally expected to be smaller than $2\theta_r$ for actual data sets from simulation, as confirmed by the values indicated in the previous paragraph. The reason for this is illustrated in Figure 3. This figure shows, for every pair of

https://doi.org/10.1021/acs.jcim.2c01079
J. Chem. Inf. Model. 2022, 62, 5738−5745

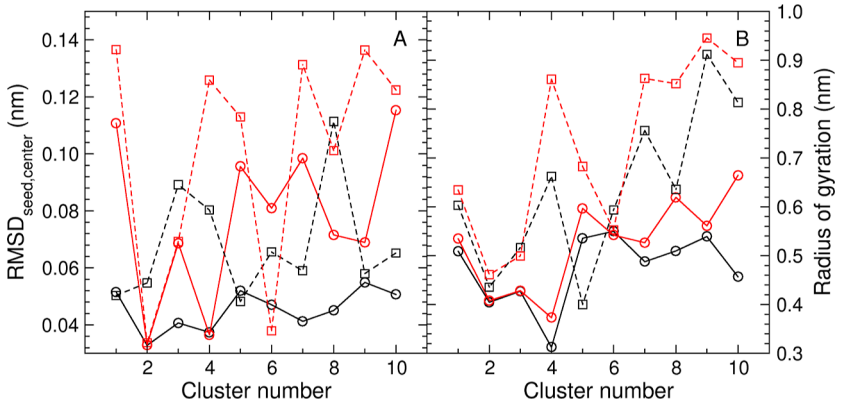


Figure 6. Distance of the seed from the center and cluster compactness (lines between points are drawn only to help visually distinguish the four data sets in each plot.) (**A**) rmsd between the seed element and the center of geometry of the elements in the cluster. (**B**) Radius of gyration of the cluster. Black circles (black solid line): RTC clustering with θ_r = 0.12 nm. Red circles (red solid line): QTC clustering with θ_d = 0.20 nm. Black squares (black dashed line): RTC clustering with θ_r = 0.17 nm. Red squares (red dashed line): QTC clustering with θ_d = 0.25 nm.

elements (excluding the seed) of clusters $C_1^{RTC}(0.12)$ (panel A) and $C_1^{RTC}(0.17)$ (panel B), the relation between the sum of their rmsd distances to the seed and the angle between the vectors associated with these distances. Thus, while there are pairs of points for which the sum of rmsds adds up indeed to $2\theta_r$, that is, 0.24 nm (panel A) and 0.34 nm (panel B), the angle between the corresponding vectors is in no case close to 180°, which precludes the maximum diameter from being reached. This figure also shows that, as expected (see panel A in Figure 2), the point density is abruptly cut (solid wall at 0.24 nm) when using the 0.12 nm threshold (panel A), while a threshold of 0.17 nm leads to a well-defined cluster (panel B).

Figure 4 shows the distribution of rmsd values within each of the first 10 clusters, for the RTC clustering with θ_r = 0.12 nm and the QTC clustering with θ_d = 0.20 nm. Seven of the clusters, including the first four, overlap almost perfectly. For three of the clusters, the QTC algorithm appears to have a higher tendency to populate the far-right side of the distribution.

Figure 5 shows the corresponding distributions for the RTC clustering with θ_r = 0.17 nm and the QTC clustering with θ_d = 0.25 nm. It becomes here more apparent that the QTC algorithm has a higher tendency to generate split distributions and populate the far-right side of the distribution. For example, it can be observed that an artificial cluster C_4^{QTC}, containing two different populations with similar weights, has displaced by one position in the ranking the QTC clusters that correspond to C_4^{RTC} and C_5^{RTC}.

We obtained for each cluster C_m, as shown in Figures 4 and 5, the center of geometry of the elements of the cluster, $\mathbf{x}_m^{(c)}$, and then calculated the rmsd between the seed element and this point, as well as the radius of gyration of the cluster. Note that the radius of gyration of the mth cluster is here defined as

$$\mathrm{rog}_m = \left(\frac{1}{|C_m|} \sum_{\mathbf{x}_i \in C_m} (\mathbf{x}_i - \mathbf{x}_m^{(c)})^2 \right)^{1/2} \quad (1)$$

The results are shown in Figure 6. As anticipated, panel A confirms that while the seeds of QTC clusters are in a majority of cases further from the center of geometry of the cluster elements than the seeds of RTC clusters, the latter are clearly off-center also. We therefore suggest to avoid, in either case, the term central element to refer to the seed element. Panel B illustrates that, except for cases in which a tight match between the rmsd distributions in Figures 4 and 5 exists, the RTC clusters tend to be more compact (lower rog) than the QTC clusters.

When looking at the total number of clusters generated by the two algorithms, we see that they differ remarkably (see the Supporting Information). Thus, while the number of clusters generated by the RTC algorithm using θ_r values of 0.12 and 0.17 nm is 1338 and 493, respectively, corresponding numbers for the QTC algorithm with θ_d values of 0.20 and 0.25 nm are 599 and 276, respectively. However, focusing on the upper part of the ranking list, we see that for RTC with θ_r = 0.12 nm the number of clusters with 100 or more elements is 6 and these cover 19% of the overall population, while for QTC with θ_d = 0.20 nm the number of clusters is also 6 and they cover 20% of the population. If we extend this to clusters with 10 or more elements, the numbers start to diverge, with RTC producing 98 clusters that cover 50% of the population and QTC producing 161 clusters covering 70% of the population. The trends are similar but the results less divergent for the comparison between RTC with θ_r = 0.17 nm and QTC with θ_d = 0.25 nm. Here, the RTC algorithm generates 8 clusters with 100 or more elements covering 25% of the population and 149 clusters with 10 or more elements covering 82% of the population, while the QTC algorithm generates 7 clusters with 100 or more elements covering 25% of the population and 150 clusters with 10 or more elements covering 91% of the population. Thus, while the upper part of the ranking looks very much the same with the two algorithms, RTC produces many more small clusters at the end of the ranking.

How can we explain such large differences in the total number of clusters? The neighbor-search volume for the RTC algorithm is strictly spherical, while for the QTC algorithm it is a volume within a sphere, with the diameter as the only shape restraint. This makes the QTC algorithm more flexible in terms of cluster shapes, allowing it to capture more of the elements that would lay just outside a cluster boundary in the RTC case. Thus, the RTC algorithm tends to generate many more artificial clusters with orphan elements in the lower part of the cluster ranking. On the other hand, the greater shape flexibility of the QTC algorithm makes it have a higher propensity to incorporate points from non-self neighbor densities in a cluster, thus mixing in it different populations.

https://doi.org/10.1021/acs.jcim.2c01079
J. Chem. Inf. Model. 2022, 62, 5738−5745

However, and as a conclusion, these differences between the two algorithms tend to be rather irrelevant in practice.

DATA AND SOFTWARE AVAILABILITY

The data used in this study was downloaded from https://github.com/LQCT/BitQT/blob/master/examples/as indicated under the Computational Details section. Although the calculations shown here were performed with inhouse software, these calculations can be performed with a variety of software packages implementing the RTC and QTC algorithms (see some of the potential choices in González-Alemán et al.[2]). We provide in the Supporting Information a comparison between our results for the tau polypeptide and results using one of the alternative software options in each case. The results for the two software choices are exactly the same, with small differences in the QTC case due to implementation details that are explained in the SI document and conform in both cases with the QTC algorithm.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.2c01079.

Tau_results_diff_software.xlsx: listing of clusters, including seed element and number of elements, from the tau-polypeptide example, with θ_r values of 0.12 nm and 0.17 nm (RTC clustering) and θ_d values of 0.20 nm and 0.25 nm (QTC clustering), corresponding to the data presented here (inhouse software) and computed with alternative available software, as indicated under the Computational Details section (XLSX)

AUTHOR INFORMATION

Corresponding Author

Xavier Daura − *Catalan Institution for Research and Advanced Studies (ICREA), Barcelona 08010, Spain; Institute of Biotechnology and Biomedicine, Universitat Autònoma de Barcelona, Cerdanyola del Vallès 08193, Spain; Centro de Investigación Biomédica en Red de Bioingeniería, Biomateriales y Nanomedicina, Instituto de Salud Carlos III, Cerdanyola del Vallès 08193, Spain;* orcid.org/0000-0001-9235-6730; Email: xavier.daura@uab.cat

Author

Oscar Conchillo-Solé − *Institute of Biotechnology and Biomedicine and Department of Genetics and Microbiology, Universitat Autònoma de Barcelona, Cerdanyola del Vallès 08193, Spain;* orcid.org/0000-0003-4266-246X

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.2c01079

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank Joan-Emma Shea and Zachary Levine for permission to use their tau-polypeptide simulation trajectories. This work received financial support from the Spanish Ministry for Science and Innovation (grant PID2019-111364RB-I00).

REFERENCES

(1) Heyer, L. J.; Kruglyak, S.; Yooseph, S. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Res.* **1999**, *9*, 1106−1115.

(2) González-Alemán, R.; Hernández-Castillo, D.; Caballero, J.; Montero-Cabrera, L. A. Quality Threshold Clustering of Molecular Dynamics: A Word of Caution. *J. Chem. Inf. Model.* **2020**, *60*, 467−472.

(3) Daura, X.; van Gunsteren, W. F.; Mark, A. E. Folding−unfolding thermodynamics of a β-heptapeptide from equilibrium simulations. *Proteins* **1999**, *34*, 269−280.

(4) Daura, X.; Gademann, K.; Jaun, B.; Seebach, D.; van Gunsteren, W. F.; Mark, A. E. Peptide Folding: When Simulation Meets Experiment. *Angew. Chem., Int. Ed.* **1999**, *38*, 236−240.

(5) Shea, J.-E.; Levine, Z. A. Studying the Early Stages of Protein Aggregation Using Replica Exchange Molecular Dynamics Simulations. *Methods Mol. Biol.* **2016**, *1345*, 225−250.

(6) McLachlan, A. D. Rapid comparison of protein structures. *Acta Cryst. A* **1982**, *38*, 871−873.

(7) de Vries, S. J.; van Dijk, A. D. J.; Krzeminski, M.; van Dijk, M.; Thureau, A.; Hsu, V.; Wassenaar, T.; Bonvin, A. M. J. J. HADDOCK versus HADDOCK: New features and performance of HADDOCK2.0 on the CAPRI targets. *Proteins* **2007**, *69*, 726−733.