Molecular Distance Measures[*]

Lydia E. Kavraki

Abstract

Given a set of structures of the same molecule, it is often necessary to decide which are more similar or less similar to each other. This module presents a few ways to approach that problem, including root mean squared distance (RMSD), least RMSD, and intramolecular distance measures.

Topics in this Module

- Comparing Molecular Conformations (Section 1: Comparing Molecular Conformations)
- RMSD and lRMSD (Section 2: RMSD and lRMSD)
- Optimal Alignment for lRMSD Using Rotation Matrices (Section 3: Optimal Alignment for lRMSD Using Rotation Matrices)
- Optimal Alignment for lRMSD Using Quaternions (Section 4: Optimal Alignment for lRMSD Using Quaternions)
  - Introduction to Quaternions (Section 4.1: Introduction to Quaternions)
  - Quaternions and Three-Dimensional Rotations (Section 4.2: Quaternions and Three-Dimensional Rotations)
  - Optimal Alignment with Quaternions (Section 4.3: Optimal Alignment with Quaternions)
- Intramolecular Distance and Related Measures (Section 5: Intramolecular Distance and Related Measures)

1 Comparing Molecular Conformations

Molecules are not rigid. On the contrary, they are highly flexible objects, capable of changing shape dramatically through the rotation of dihedral angles. We need a measure to express how much a molecule changes going from one conformation to another, or alternatively, how different two conformations are from each other. Each distinct shape of a given molecule is called a **conformation**. Although one could conceivably compute the volume of the intersection of the alpha shapes for two conformations (see Molecular Shapes and Surfaces[1] for an explanation of alpha shapes) to measure the shape change, this is prohibitively computationally expensive. Simpler measures of distance between conformations have been defined, based on variables such as the Cartesian coordinates for each atom, or the bond and torsion angles within the molecule. When working with Cartesian coordinates, one can represent a molecular conformation as a vector whose components are the Cartesian coordinates of the molecule's atoms. Therefore, a conformation for a molecule with N atoms can be represented as a 3N-dimensional vector of real numbers.

[*] Version 1.23: Jun 11, 2007 4:52 am -0500

[†] http://creativecommons.org/licenses/by/1.0

[1] "Molecular Shapes and Surfaces" <http://cnx.org/content/m11616/latest/>

2 RMSD and lRMSD

One of the most widely accepted difference measures for conformations of a molecule is **least root mean square deviation (lRMSD)**. To calculate the RMSD of a pair of structures (say x and y), each structure must be represented as a 3N-length (assuming N atoms) vector of coordinates. The RMSD is the square root of the average of the squared distances between corresponding atoms of x and y. It is a measure of the average atomic displacement between the two conformations:

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}|x_i - y_i|^2}$$

However, when molecular conformations are sampled from molecular dynamics or other forms of sampling, it is often the case that the molecule drifts away from the origin and rotates in an arbitrary way. The lRMSD distance aims at compensating for these facts by representing the minimum RMSD over all possible relative positions and orientations of the two conformations under consideration. Calculating the lRMSD consists of first finding an optimal alignment of the two structures, and then calculating their RMSD. Note that aligning two conformations may require both a translation and rotation. In other words, before computing the RMSD distance, it is necessary to remove the translation of the centroid of both conformations and to perform an "optimal alignment" or "optimal rotation" of them, since these two factors artificially increase the RMSD distance between them.

Finding the optimal rotation to minimize the RMSD between two point sets is a well-studied problem, and several algorithms exist. The **Kabsch Algorithm***[1]*[4]*[2]*[5], which is implemented in several molecular modeling packages, solves a matrix equation for the three dimensional rotation matrix corresponding to the optimal rotation. An alternative approach, discussed in detail after the matrix method, uses a compact representation of rotational transformations called **quaternions***[3]*[3]*[4]*[1]. Quaternions are currently the preferred representation for global rotation in calculating lRMSD, since they require less numbers to be stored and are easy to re-normalize. In contrast, re-normalization of orthonormal matrices is quite expensive and potentially numerically unstable. Both quaternions and their application to global alignment of conformations will be presented after the next section.

3 Optimal Alignment for lRMSD Using Rotation Matrices

This section presents a method for computing the optimal rotation between 2 datasets as an orthonormal rotation matrix. As stated earlier, this approach is slightly more numerically unstable (since guaranteeing the orthonormality of a matrix is harder than the unit length of a quaternion) and requires taking care of the special case when the resulting matrix may not be a proper rotation, as discussed below.

As stated earlier, the optimal alignment requires both a translation and a rotation. The translational part of the alignment is easy to calculate. It can be proven that the optimal alignment is obtained by translating one set so that its centroid coincides with the other set's centroid (see section 2-C of *[3]*[?] for proof). The centroid of a point set a is simply the average position of all its points:

Centroid of a Point Set

$$a_c = \frac{1}{n}\sum_{i=1}^{n} a_i$$

Figure 1: The centroid of a point set is the average position over all the points.

We can then redefine each point in two sets A and B as a deviation from the centroid:

Redefining Point Sets in Terms of Centroids

$$a_i' = a_i - a_c$$

$$b_i' = b_i - b_c$$

Figure 2: Each point is now expressed as a deviation from its set's centroid.

Given this notation relative to the centroid, we can explicitly set the centroids to be equal and proceed with the rotational part of the alignment.

One of the first references to the solution of this problem in matrix form is from Kabsch *[1]*[4]*[2]*[5]. The Kabsch method uses Lagrange multipliers[2] to solve a minimization problem to find the optimal rotation. Here, we present a slightly more intuitive method based on matrix algebra and properties, that achieves the same result. This formulation can be found in *[4]*[1] and *[5]*[2]. Imagine we wish to align two conformations composed of N atoms each, whose Cartesian coordinates are given by the vectors x and y. The main idea behind this approach is to find a 3x3 orthonormal matrix U such that the application of U to the atom positions of one of the data vectors, x, aligns it as best as possible with the other data vector, y, in the sense that the quantity to minimize is the distance $\mathrm{d}\left(\mathrm{Ux},\mathrm{y}\right)$, where x and y are assumed to be **centered**, that is, both their centroids coincide at the origin (centering both conformations is the first step). Mathematically, this problem can be stated as the **minimization** of the following quantity:

$$E = \frac{1}{N}\sum_{i=1}^{N} \left|Ux_i - y_i\right|^2$$

When E is a minimum, the square root of its value becomes the least RMSD (lRMSD) between x and y. Being an orthonormal rotation matrix, U needs to satisfy the orthonormality property $UU^T = I$, where I is the identity matrix. The orthonormality contraint ensures that the rows and columns are mutually orthogonal, and that their length (as vectors) is one. Any orthonormal matrix represents a rigid orientation (transformation) in space. The only problem with this approach as is, is that all orthonormal matrices

[2] http://en.wikipedia.org/wiki/Lagrange_multipliers

encode a rigid transformation, but if the rows/columns of the matrix do not constitute a **right handed system**, then the rotation is said to be **improper**. In an improper rotation, one of the three directions may be "mirrored". Fortunately, this case can be detected easily by computing the determinant of the matrix U, and if it is negative, correcting the matrix. Denoting Ux as x', and moving the constant factor N to the left, the formula for the error becomes:

$$NE = \sum_{i=1}^{N} |x'_i - y_i|^2$$

An alternative way to represent the two point sets, rather than a one-dimensional vector or as separate atom coordinates, is using two 3xN matrices (N atoms, 3 coordinates for each). Using this scheme, x is represented by the matrix X and y is represented by the matrix Y. Note that column $1 \leq i \leq N$ in these matrices stands for point (atom) x_i and y_i, respectively. Using this new representation, we can write:

$$NE = \sum_{i=1}^{N} |x'_i - y_i|^2 = Tr((X' - Y)^T (X' - Y))$$

where X' $= UX$ and Tr (A) stands for the trace[3] of matrix A, the sum of its diagonal elements. It is easy to see that that the trace of the matrix to the right amounts precisely to the sum on the left (simply carrying out the multiplication of the first row/column should convince the reader). The right-hand side of the equation can be expanded into:

$$Tr((X' - Y)^T (X' - Y)) = Tr(X'^T X') + Tr(Y^T Y) - 2Tr(Y^T X')$$

Which follows from the properties of the trace operator, namely: Tr (A + B) = Tr (A) + Tr (B) , Tr (AB) = Tr (BA), Tr ($A^T = Tr\,(A)$, and Tr (kA) = kTr (A). Furthermore, the first two terms in the expansion above represent the sum of the squares of the components x_i and y_i, so it can be rewritten as:

$$NE = \sum_{i=1}^{N} (|x_i|^2 + |y_i|^2) - 2Tr(Y^T X')$$

Note that the x components do not need to be primed (i.e., x') since the rotation U around the origin does not change the length of x_i. Note that the summation above does not depend on U, so **minimizing** E is equivalent to **maximizing**Tr (Y^TX'. For this reason, the rest of the discussion focuses on finding a proper rotation matrix U that maximizes Tr (Y^TX'. Remembering that X' $= UX$, the quantity to maximize is then Tr ($\left(Y^T U\right) X$. From the property of the trace operator, this is equivalent to Tr ($\left(XY^T\right) U$. Since XY^T is a

[3] http://en.wikipedia.org/wiki/Trace_%28linear_algebra%29

square 3x3 matrix, it can be decomposed through the Singular Value Decomposition[4] technique (SVD) into $XY^T = \mathrm{VSW}^T$, where V and W^T are the matrices of left and right eigenvectors (which are orthonormal matrices), respectively, and S is a diagonal 3x3 matrix containing the eigenvalues s_1, s_2, s_3 in decreasing order. Again from the properties of the trace operator, we obtain that:

$$Tr(Y^T X') = Tr(VSW^T U) = Tr(SW^T UV) = \sum_{i=1}^{3} s_i w_i^T U v_i$$

If we introduce the 3x3 matrix T as the product $T = W^T\mathrm{UV}$, we can rewrite the above expression as:

$$Tr(Y^T X') = \sum_{i=1}^{3} s_i T_{ii} \le \sum_{i=1}^{3} s_i$$

Since T is the product of orthonormal matrices, it is itself an orthonormal matrix and $\det(\mathrm{T}) = +/-1$. This means that the absolute value of each element of this matrix is no more than one, from where the last equality follows. It is obvious that the maximum value of the left hand side of the equation is reached when the diagonal elements of T are equal to 1, and since it is an orthonormal matrix, all other elements must be zero. This results in $T = I$. Moreover, since $T = W^T\mathrm{UV}$, we can write that $W^T\mathrm{UV} = I$, and because W and V are orthonormal, $WW^T = I$ and $VV^T = I$. Multiplying $W^T\mathrm{UV}$ by W to the left and V^T to the right yields a solution for U:

$$U = WV^T$$

Where V and W^T are the matrices of left and right eigenvectors, respectively, of the covariance matrix $C = XY^T$. This formula ensures that U is orthonormal (the reader should carry out the high-level matrix multiplication and verify this fact).

The only remaining detail to take care of is to make sure that U is a **proper** rotation, as discussed before. It could indeed happen that $\det(\mathrm{U}) = -1$ if its rows/columns do not make up a right-handed system. When this happens, we need to compromise between two goals: maximizing $\mathrm{Tr}\,(Y^T\mathrm{X}'$ and respecting the constraint that $\det(\mathrm{U}) = +1$. Therefore, we need to settle for the second largest value of $\mathrm{Tr}\,(Y^T\mathrm{X}'$. It is easy to see what the second largest value is; since:

$$Tr(Y^T X') = \sum_{i=1}^{3} s_i T_{ii} = s_1 T_{11} + s_2 T_{22} + s_3 T_{33}$$

$$\text{where } s_1 \ge s_2 \ge s_3 \ge 0 \text{ and } |T_{ii}| \le 1$$

[4] http://en.wikipedia.org/wiki/Singular_value_decomposition

then the second largest value occurs when $T_{11} = T_{22} = +1$ and $T_{33} = -1$. Now, we have that T cannot be the identity matrix as before, but instead it has the lower-right corner set to -1. Now we finally have a unified way to represent the solution. If $\det(\mathrm{C}) > 0$, T is the identity; otherwise, it has a -1 as its last element. Finally, these facts can be expressed in a single formula for the optimal rotation U by stating:

$$U = W \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{pmatrix} V^T$$

where $d = \mathrm{sign}(\det(\mathrm{C}))$. In the light of the preceding derivation, all the facts that have been presented as a proof can be succinctly put as an algorithm for computing the optimal rotation to align two data sets x and y:

Optimal rotation

1. Build the 3xN matrices X and Y containing, for the sets x and y respectively, the coordinates for each of the N atoms after centering the atoms by subtracting the centroids.
2. Compute the covariance matrix $C = XY^T$
3. Compute the SVD (Singular Value Decomposition) of $C = VSW^T$
4. Compute $d = \mathrm{sign}(\det(\mathrm{C}))$
5. Compute the optimal rotation U as

$$U = W \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{pmatrix} V^T$$

4 Optimal Alignment for lRMSD Using Quaternions

Another way of solving the optimal rotation for the purposes of computing the lRMSD between two conformations is to use **quaternions**. These provide a very compact way of representing rotations (only 4 numbers as compared to 9 or 16 for a rotation matrix) and are extremely easy to normalize after performing operations on them. Next, a general introduction to quaternions is given, and then they will be used to compute the optimal rotation between two point sets.

4.1 Introduction to Quaternions

Quaternions are an extension of complex numbers. Recall that complex numbers are numbers of the form a + bi, where a and b are real numbers and i is the canonical imaginary number, equal to the square root of -1. Quaternions add two more imaginary numbers, j and k. These numbers are related by the set of equalities in the following figure:

Equation Relating the Imaginary Elements i, j and k

$$i^2 = j^2 = k^2 = i \cdot j \cdot k = -1$$

Figure 3: Properties of quaternion arithmetic follow directly from these equalities.

These equalities give rise to some unusual properties, especially with respect to multiplication.

Multiplication Table for the Imaginary Elements i, j and k

$$\begin{array}{ll} i \cdot j = k & j \cdot i = -k \\ j \cdot k = i & k \cdot j = -i \\ k \cdot i = j & i \cdot k = -j \end{array}$$

Figure 4: Note that multiplication of i, j, and k is **anti-commutative**.

Given this definition of i, j, and k, we can now define a quaternion.

Definition of a Quaternion

$$a + b \cdot i + c \cdot j + d \cdot k$$

Figure 5: A quaternion is a number of the above form, where a, b, c, and d are real-valued scalars and i, j, and k are imaginary numbers as defined above.

Based on the definitions of i, j and k, we can also derive rules for addition and multiplication of quaternions. Assume we have two quaternions, p and q, defined as follows:

Quaternions p and q

$$p = a + bi + cj + dk$$
$$q = r + si + tj + uk$$

Figure 6: Definition of quaternions p and q for later use.

Addition of p and q is fairly intuitive:

Addition of Quaternions p and q

$$p + q = (a + r) + (b + s)i + (c + t)j + (d + u)k$$

Figure 7: Quaternion addition closely resembles vector addition. Corresponding coefficients are added to yield the sum quaternion. This operation is associative and commutative.

The dot product and magnitude of a quaternion also closely resemble those operations for vectors. Note that a **unit quaternion** is a quaternion with magnitude 1 under this definition:

Dot (Inner) Product of p and q

$$p \cdot q = ar + bs + ct + du$$

Figure 8: The dot product of quaternions is analogous to the dot product of vectors.

Magnitude of Quaternion p

$$||p||^2 = p \cdot p = a^2 + b^2 + c^2 + d^2$$

Figure 9: As with vectors, the square of the magnitude of p is the dot product of p with itself.

Multiplication, however, is not, due to the definitions of i, j, and k:

Multiplication of Quaternions p and q

$$pq = (ar - bs - ct - du) + (br + as + cu - dt)i + (cr + at + ds - bu)j + (dr + au + bt - cs)k$$

Figure 10: This result can be confirmed by carrying out long multiplication of p and q. There is no analog in vector arithmetic for quaternion multiplication.

Quaternion multiplication also has two equivalent matrix forms which will become relevant later in the derivation of the alignment method:

Multiplication of Quaternions p and q, Matrix Forms

$$pq = \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix} q$$

$$qp = \begin{bmatrix} a & -b & -c & -d \\ b & a & d & -c \\ c & -d & a & b \\ d & c & -b & a \end{bmatrix} q$$

Figure 11: Note that quaternions can be represented as column vectors with the imaginary components omitted. This allows vector notation to be used for many quaternion operations, including multiplication. The quaternion a + bi + cj + dk, for example, may be represented by a column vector of the form [a, b, c, d].

These useful properties of quaternion multiplication can be derived easily using the matrix form for multiplication, or they can be proved by carrying out the products:

Some properties of Quaternion Multiplication

$$\begin{aligned} qp \cdot qr &= (q \cdot q)(p \cdot r) \\ (pq)(pq) &= (p \cdot p)(q \cdot q) \text{ (magnitude of product is product of magnitudes)} \\ (pq) \cdot r &= p \cdot (rq^*) \end{aligned}$$

Figure 12: Some useful properties. q* is the quaternion conjugate, a-bi-cj-dk

4.2 Quaternions and Three-Dimensional Rotations

A number of different methods exist for denoting rotations of rigid objects in three-dimensional space. These are introduced in a module on protein kinematics. **Unit quaternions** represent a rotation of an angle around an arbitrary axis. A rotation by the angle theta about an axis represented by the unit vector v = [x, y, z] is represented by a unit quaternion:

Unit Quaternion and Rotation

$$q = cos\frac{\theta}{2} + sin\frac{\theta}{2}(xi + yj + zk)$$

Figure 13: This unit quaternion represents a rotation of theta about the axis defined by unit vector v = [x, y, z].

Like rotation matrices, quaternions may be composed with each other via multiplication. The major advantage of the quaternion representation is that it is more robust to **numerical instability** than orthonormal matrices. Numerical instability results from the fact that, because computers use a finite number of bits to represent real numbers, most real numbers are actually represented by the nearest number the computer is capable of representing. Over a series of floating point operations, the error caused by this inexact representation accumulates, quite rapidly in the case of repeated multiplications and divisions. In manipulating orthonormal transformation matrices, this can result in matrices that are no longer orthonormal, and therefore not valid rigid transformations. Finding the "nearest" orthonormal matrix to an arbitrary matrix is not a well-defined problem. Unit-length quaternions can accumulate the same kind of a numerical error as rotation matrices, but in the case of quaternions, finding the nearest unit-length quaternion to an arbitrary quaternion **is** well defined. Additionally, because quaternions correspond more directly to the axis-angle representation of three-dimensional rotations, it could be argued that they have a more intuitive interpretation than rotation matrices. Quaternions, with four parameters, are also more memory efficient than 3x3 matrices. For all of these reasons, quaternions are currently the preferred representation for three-dimensional rotations in most modeling applications.

Vectors can be represented as purely imaginary quaternions, that is, quaternions whose scalar component is 0. The quaternion corresponding to the vector v = [x, y, z] is q = xi + yj + zk.

We can perform rotation of a vector in quaternion notation as follows:

Rotation Using Unit Quaternions

$$r = xi + yj + zk$$
$$q = a + bi + cj + dk$$
$$q^* = a - bi - cj - dk$$

$$r' = qrq^*$$

Figure 14: In this figure, r is the vector [x, y, z] in quaternion form, q is a unit (rotation) quaternion, q* is the conjugate of q, and r' is r after the rotation has been performed.

4.3 Optimal Alignment with Quaternions

The method presented here is from Berthold K. P. Holm, "Closed-form solution of absolute orientation using unit quaternions." Journal of the Optical Society of America A, 4:629-642.

The alignment problem may be stated as follows:

- We have two sets of points (atoms) A and B for which we wish to find an optimal alignment, defined as the alignment for which the root mean square difference between each point in A and its corresponding point in B is minimized.
- We know which point in A corresponds to which point in B. This is necessary for any RMSD-based method.

As for the case of rotation matrices, the translational part of the alignment consists of making the centroids of the two data sets coincide. To find the optimal rotation using quaternions, recall that the dot product of two vectors is maximized when the vectors are in the same direction. The same is true when the vectors are represented as quaternions. Using this property, we can define a quantity that we want to maximize (proof here[5]):

The Objective Function for Rotational Alignment (Quaternion Form)

$$\sum_{i=1}^{n}(qa'_iq^* \cdot b'_i)$$

Figure 15: We want to find the rotation on set A that maximizes the sum of the dot products of the rotated vectors of A with the vectors of B, all expressed as offsets from the set centroids.

[5] http://cnx.org/content/m11608/latest/quaternion-proof1.png

Equivalently, using the last property from the section "Introduction to quaternions", we get:

$$\sum_{i=1}^{n}(qa_i') \cdot (b_i'q)$$

Figure 16: The objective restated.

Now, recall that quaternion multiplication can be represented by matrices, and that the quaterions a and b have a 0 real component:

$$qa_i' = \begin{bmatrix} 0 & -a'_{i,x} & -a'_{i,y} & -a'_{i,z} \\ a'_{i,x} & 0 & a'_{i,z} & -a'_{i,y} \\ a'_{i,y} & -a'_{i,z} & 0 & a'_{i,x} \\ a'_{i,z} & a'_{i,y} & -a'_{i,x} & 0 \end{bmatrix} q = A'q$$

$$b_i'q = \begin{bmatrix} 0 & -b'_{i,x} & -b'_{i,y} & -b'_{i,z} \\ b'_{i,x} & 0 & -b'_{i,z} & b'_{i,y} \\ b'_{i,y} & b'_{i,z} & 0 & -b'_{i,x} \\ b'_{i,z} & -b'_{i,y} & b'_{i,x} & 0 \end{bmatrix} q = B'q$$

Figure 17: These substitutions will be used to restate the function to be maximized.

Using these matrices, we can derive a new form for the objective function:

$$\sum_{i=1}^{n}(A'_i q)\cdot(B'_i q)$$

$$\sum_{i=1}^{n} q^T A_i^{T\prime} B'_i q$$

$$q^T\left(\sum_{i=1}^{n} A_i^{T\prime} B'_i\right) q$$

$$q^T\left(\sum_{i=1}^{n} N_i\right) q = q^T N q$$

Figure 18: The third step follows because each term in the sum is multiplied on the left and right by q, so the q factors can be moved outside the sum. The fourth step simply renames the sum of matrix products to a single matrix, N, based on which we can find q.

where:

$$N_i = A_i^{T\prime} B'_i$$

$$N = \sum_{i=1}^{n} N_i$$

Figure 19: Now the problem is stated in terms of a matrix product optimization.

The quaternion that maximizes this product is the eigenvector of N that corresponds to its most positive eigenvalue (proof here[6]). The eigenvalues can be found by solving the following equation, which is quartic in lambda:

[6] http://cnx.org/content/m11608/latest/quaternion-proof2.png

$$det(N - \lambda I) = 0$$

Figure 20: I is the 4x4 identity matrix.

This quartic equation can be solved by a number of standard approaches. Finally, given the maximum eigenvalue lambda-max, the quaternion corresponding to the optimal rotation is the eigenvector v:

$$(N - \lambda_{max} I)v = 0$$

Figure 21: This equation can be solved to find the optimal rotation.

A closed-form solution to this equation for v can be found by applying techniques from linear algebra. One possible algorithm, based on constructing a matrix of cofactors, is presented in appendix A5 of the source paper *[3]*[?].

In summary, the alignment algorithm works as follows:

- Recalculate atom coordinates as displacements from the centroid of each molecule. The optimal translation superimposes the centroids.
- Construct the matrix N based on matrices A and B for each atom.
- Find the maximum eigenvalue by solving the quartic eigenvalue equation.
- Find the eigenvector corresponding to this eigenvalue. This vector is the quaternion corresponding to the optimal rotation.

This method appears computationally intensive, but has the major advantage over other approaches of being a closed-form, unique solution.

5 Intramolecular Distance and Related Measures

RMSD and lRMSD are not ideally suited for all applications. For example, consider the case of a given conformation A, and a set S of other conformations generated by some means. The goal is to estimate which conformations in S are closest in potential energy to A, making the assumption that they will be the conformations most structurally similar to A. The lRMSD measure will find the conformations in which the overall average atomic displacement is least. The problem is that if the quantity of interest is the potential energy of conformations, not all atoms can be treated equally. Those on the outside of the protein can often move a fair amount without dramatically affecting the energy. In contrast, the core of the molecule tends to be more compact, and therefore a slight change in the relative positions of a pair of atoms could lead to overlap of the atoms, and therefore a completely infeasible structure and high potential energy. A class of distance measures and pseudo-measures based on **intramolecular** distances have been developed to address this shortcoming of RMSD-based measures.

Assume we wish to compare two conformations P and Q of a molecule with N atoms. Let p_{ij} be the distance between atom i and atom j in conformation P, and let q_{ij} be the same distance for conformation Q. Then the intramolecular distance is defined as

$$\sqrt{\frac{1}{N\cdot(N+1)}\sum_{i<j}(p_{ij}-q_{ij})^2}$$

Figure 22: Intra-molecular distance (dRMSD)

One of the main computational advantages of this class of approaches is that we do not have to compute the alignment between P and Q. On the other hand, for this metric we need to sum over a quadratic number of terms, whereas for RMSD the number of terms is linear in the number of atoms. Approximations can be made to speed up this computation, as shown in *[7]*[6]. Also, the intramolecular distance measure given above, which is sometimes referred to as the dRMSD, is subject to the problem that pairs of atoms most distant from each other are the ones that contribute the greatest amount to their measured difference.

An interesting open problem is to come up with physically meaningful molecular distance metric that allows for fast nearest neighbor computations. This can be useful for, for example, clustering conformations. One proposed method is the **contact distance**. Contact distance requires constructing a **contact map** matrix for each conformation indicating which pairs of atoms are less than some threshold separation. The distance measure is then a measure of the difference of the contact maps.

Contact Distance

$$\begin{aligned} C_{ij} = 1 \text{ if } & r_{ij} < r_c \\ & 0 \text{ otherwise} \end{aligned}$$

$$r_c = \text{contact cutoff distance}$$

$$q(A,B) = \frac{\sum_{i<j} C^a_{ij} C^b_{ij}}{max(\sum_{i<j} C^a_{ij}, \sum_{i<j} C^b_{ij})}$$

$$D(A,B) = 1 - q(A,B)$$

Figure 23: Contact maps (C) are calculated for each structure, and the differences in these contact maps used to define a distance D.

Other distance measures attempt to weight each pair in the dRMSD based on how close the atoms are, with closer pairs given more weight, in keeping with the intuition that small changes in the relative positions of nearby atoms are more likely to result in collisions. One such measure is the normalized **Holm and Sander Score**.

Holm and Sander Distance

$$D(A,B) = \sum_{i<j} \frac{|r^a_{ij} - r^b_{ij}|}{r^a_{ij} + r^b_{ij}} e^{-(r^a_{ij}+r^b_{ij})^2/4r_0^2}$$

Figure 24: This distance function is weighted to accentuate the importance of differences in structures that are relatively close to each other. These are the contacts most likely to affect the potential energy of the structure.

This score is technically a **pseudo-measure** rather than a measure because it does not necessarily obey the **triangle inequality**.

The definition of distance measures remains an open problem. For reference on ongoing work, see articles that compare several methods, such as *[5]*[**?**].

Recommended Reading:

The first two papers are the original descriptions of the Kabsch Algorithm, and use rotations represented as orthonormal matrices to find the correct rotational transformation. Many software packages use this alignment method. The third and fourth papers use quaternions. The alignment method presented in the previous section comes from the third paper:

- W. Kabsch. (1976). A Solution for the Best Rotation to Relate Two Sets of Vectors[7] . Acta Crystallographica, 32, 922-923.
- W. Kabsch. (1978). A Discussion of the Solution for the Best Rotation to Relate Two Sets of Vectors[8] . Acta Crystallographica, 34, 827-828.
- Berthold K. P. Horn. (1986). Closed-form solution of absolute orientation using unit quaternions.[9] Journal of the Optical Society of America, 4:629-642.
- E. A. Coutsias and C. Seok and K. A. Dill. (2004). Using quaternions to calculate RMSD.[10] Journal of Computational Chemistry, 25, 1849-1857.
- Wallin, S., J. Farwer and U. Bastolla. (2003). Testing similarity measures with continuous and discrete protein models [11] . Proteins, 50:144-157.

References

[1] C. Seok Coutsias, E. A. and K. A. Dill. Using quaternions to calculate rmsd. *Journal of Computational Chemistry*, 25:1849–1857, 1978.

[2] G. H. Golub and C. F. V. Loadn. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.

[3] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629–642, 1986.

[4] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, 32:922–923, 1976.

[7] http://journals.iucr.org/a/issues/1976/05/00/a12999/a12999.pdf
[8] http://journals.iucr.org/a/issues/1978/05/00/a15629/a15629
[9] http://josaa.osa.org/abstract.cfm?id=2711
[10] http://www3.interscience.wiley.com/cgi-bin/fulltext/109627463/PDFSTART
[11] http://www3.interscience.wiley.com/cgi-bin/abstract/101019757/ABSTRACT

[5] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, 34:827–828, 1978.

[6] F. Schwarzer and I. Lotan. *Approximation of protein structure for fast similarity measures.* ACM. Proceedings of the seventh annual international conference on research in computational molecular biology., 2003.

[7] J. Farwer Wallin, S. and U. Bastolla. Testing similarity measures with continuous and discrete protein models. *Proteins*, 50:144–157, 2003.