

On the optimal rotation-translation relating two sets of vectors

Evangelos A. Coutsias[†] and Chaok Seok[‡]
Department of Pharmaceutical Chemistry,
University of California at San Francisco,
San Francisco, CA 94118

Report, 27 September 2000

Abstract

In these notes we give a simple derivation of the optimal solid body transformation (rotation-translation) to minimize the (weighted) Euclidean distance between two ordered sets of vectors in 3-space.

1 Introduction

In comparing two shapes in three dimensional space, one must ensure that they are brought as close as is possible by an orthogonal transformation (proper rotation). Kabsch [4] solved this problem for shapes described as ordered sets of vectors. The statement of the problem, formulated and solved by Kabsch is: *Given an ordered set of vectors \mathbf{y}_k (target) and a second set \mathbf{x}_k (model), $1 \leq k \leq N$ find an orthogonal transformation \mathcal{U} such that the residual*

$$R(X, Y) := \frac{1}{2} \sum_{k=1}^N w_k (\mathcal{U} \mathbf{x}_k + \mathbf{r} - \mathbf{y}_k)^2$$

is minimized.

Kabsch's solution for the $\mathbf{r} = 0$ problem (pure rotation), parametrized by the coefficients of the rotation matrix, is somewhat involved and some of the intuitions are disguised by the complexity of the formalism. The reason is that a rotation, when expressed in terms of a 3×3 rotation matrix \mathcal{U} involves 9 parameters, the coefficients u_{ij} , which are related through the condition $\mathcal{U}\mathcal{U}^T = I$ or $\sum_{j=1}^3 u_{ij}u_{kj} = \delta_{ik}$ with δ_{ik} the Kronecker delta ($\delta_{ik} = 1$ if $i = k$ and vanishing otherwise). Thus, there are 6 constraints among the 9 coefficients and this requires a constrained minimization of the residual which involves 6 Lagrange multipliers. By making use of *quaternions* one can arrive at a simpler formulation [2], [3]. Here we give a self-contained derivation of this result together with

a brief introduction to quaternions. The motivation for this work was provided by the need for an efficient computation of the quantity $\nabla_X R$, the gradient of the minimal residual with respect to the model coordinates X . This computation is required, e.g., in the parameter optimization scheme proposed by Rosen et al. [7] as part of a strategy for improving the performance of protein folding energy minimization algorithms.

For the problem of protein folding, where one needs to compare the shape of two molecular chains, a translation is needed to produce the best possible match in addition to a rotation. In order to allow a general solid-body transformation we minimize the residual with respect to variations in both the rotation \mathcal{U} and translation \mathbf{r} . The latter is unconstrained, so that the only effect it has is to require shifting the two sets to their respective barycenters:

$$\mathbf{r} = \mathcal{U}(q)\bar{\mathbf{x}} - \bar{\mathbf{y}} := \mathcal{U}(q) \frac{1}{N} \sum_{k=1}^N \mathbf{x}_k - \frac{1}{N} \sum_{k=1}^N \mathbf{y}_k .$$

The optimal rotation is generated by a quaternion q which is found as the leading eigenvector of the matrix

$$\mathcal{S} := \sum_{k=1}^N \mathcal{A}_L(\mathbf{y}_k - \bar{\mathbf{y}}) \mathcal{A}_R(\mathbf{x}_k - \bar{\mathbf{x}}) ,$$

corresponding to its maximal eigenvalue, λ_{max} , where \mathcal{A}_L and \mathcal{A}_R are defined by eq.(10). The minimal residual is found to be:

$$\min_q R = \frac{1}{2} \sum_{k=1}^N ((\mathbf{x}_k - \bar{\mathbf{x}})^2 + (\mathbf{y}_k - \bar{\mathbf{y}})^2) - \max_{i=1,2,3,4} \lambda_i .$$

This note is organized as follows: in Sec. 1 we give an overview of quaternions and some of their properties. In Sec. 2 we carry out the residual minimization for the case of pure rotation, and arrive at a result equivalent, albeit simpler to state, to the one in Kabsch [4]. In Sec. 3 we carry out the computation of $\nabla_X R$. A MATLAB routine to carry out the optimal rotation, both with and without translation, is given in appendix A. In appendix B we show that our result is equivalent to that given in Kabsch [4].

1.1 Quaternion notation and manipulations

Below, we sketch the basic properties of quaternions. We follow the notes of Coutsiaris and Romero [1], and we refer the reader to these notes for more details. The book by Rappaport [6], which discusses quaternions in the context of molecular dynamics can also be useful as an introduction.

We define a quaternion q as a 4-vector $q = (q_0, q_1, q_2, q_3) \equiv (q_0, \mathbf{q})$ with the obvious addition law and the fundamental multiplication law given by

$$a = (a_0, \mathbf{a}), \quad b = (b_0, \mathbf{b})$$

$$\begin{aligned}
a + b &= (a_0 + b_0, \mathbf{a} + \mathbf{b}) \\
ab &= (a_0 b_0 - \mathbf{a} \cdot \mathbf{b}, a_0 \mathbf{b} + b_0 \mathbf{a} + \mathbf{a} \times \mathbf{b})
\end{aligned} \tag{1}$$

We also define the *conjugate* quaternion q^c , the norm of a quaternion $N(q)$ and the inverse quaternion q^{-1} by

$$q^c = (q_0, -\mathbf{q}), \quad (ab)^c = b^c a^c \tag{2}$$

$$N(q) = q_0^2 + \mathbf{q} \cdot \mathbf{q} \tag{3}$$

$$q^{-1} = \frac{q^c}{N(q)} = \frac{(q_0, -\mathbf{q})}{N(q)} \tag{4}$$

An important class of quaternion are the *pure* quaternions given by

$$\mathcal{Q}_0 = \{q \mid q = (0, \mathbf{q})\}. \tag{5}$$

Pure quaternions can be considered as ordinary 3-vectors $\mathbf{q} = (q_1, q_2, q_3)$ which have been promoted to 4-vectors. Note for pure quaternions $a, b \in \mathcal{Q}_0$, Eq. (1) and Eq. (2) yields

$$\begin{aligned}
a &= (0, \mathbf{a}), & b &= (0, \mathbf{b}) \\
ab &= (-\mathbf{a} \cdot \mathbf{b}, \mathbf{a} \times \mathbf{b})
\end{aligned} \tag{6}$$

$$a^c = (0, -\mathbf{a}) = -a \tag{7}$$

For our purposes we are interested in writing quaternion multiplications in terms of ordinary 4×4 matrix multiplication acting on a 4-vector. For any two quaternions a and q we define the 4×4 matrices $\mathcal{A}_L(a)$ and $\mathcal{A}_R(a)$ as the action of a on q via

$$a q \equiv \mathcal{A}_L q, \quad a \text{ operates on } q \text{ from the left} \tag{8}$$

$$q a \equiv \mathcal{A}_R q, \quad a \text{ operates on } q \text{ from the right} \tag{9}$$

Recall that matrix multiplication (the rhs of the above) always acts from the left on a 4-vector $q = (q_0, q_1, q_2, q_3)^T$ (i.e. q as a column vector). In Eq. (8) the quaternion multiplication $a q$ has a acting from the left on q translating into the 4×4 matrix \mathcal{A}_L acting on q as a column 4-vector. In Eq. (9) the quaternion multiplication $q a$ has a acting from the right on q translating into the 4×4 matrix \mathcal{A}_R acting on q as a column 4-vector. For $a = (a_0, a_1, a_2, a_3)$ the matrices \mathcal{A}_L and \mathcal{A}_R are given by

$$\mathcal{A}_R(a) = \begin{pmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & a_3 & -a_2 \\ a_2 & -a_3 & a_0 & a_1 \\ a_3 & a_2 & -a_1 & a_0 \end{pmatrix}, \quad \mathcal{A}_L(a) = \begin{pmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & -a_3 & a_2 \\ a_2 & a_3 & a_0 & -a_1 \\ a_3 & -a_2 & a_1 & a_0 \end{pmatrix}. \tag{10}$$

Quaternions provide a natural coordinate system for $SU(3)$, the group of proper rotations of 3-space. Thus they can affect rotations without the singularities

of, say, the Euler angles rotation matrices. For the 3-vector \mathbf{r}' obtained by a rotation of the vector \mathbf{r} via the orthogonal rotation matrix \mathcal{U} we have

$$\begin{aligned} \mathbf{r}' &= (0, \mathbf{r}'), & \mathbf{r} &= (0, \mathbf{r}) \\ \mathbf{r}' &= q \mathbf{r} q^c, \end{aligned} \quad (11)$$

$$\begin{aligned} (0, \mathbf{r}') &= (q_0, \mathbf{q}) (0, \mathbf{r}) (q_0, -\mathbf{q}) \\ &= (0, \mathcal{U}(q)\mathbf{r}). \end{aligned} \quad (12)$$

Using the notation of eq.(10), we can express the rotation matrix as a product of two matrices, each depending linearly on q :

$$\begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathcal{U}(q) \end{pmatrix} = \mathcal{A}_L(q) \mathcal{A}_R(q^c).$$

The 3×3 rotation matrix \mathcal{U} is given by

$$\mathcal{U}(q) = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \quad (13)$$

Since \mathcal{U} is orthogonal we have $\mathcal{U}^{-1} = \mathcal{U}^T$. Note that by its construction, $\mathcal{U}(q)$ is a proper rotation whose angle θ and axis \mathbf{c} are seen in the form of q [1]:

$$q = (q_0, \mathbf{q}) = (\sin(\theta/2), \cos(\theta/2)\mathbf{c}).$$

2 The optimal rotation in terms of quaternions.

We now find the optimal rigid body transformation to minimize the residual $R(X, Y)$ given by eq.(1). Since the translation \mathbf{r} is unconstrained, while the rotation $\mathcal{U}(q)$ depends on the unit quaternion q , for which $qq^c = 1$, we introduce the Lagrange multiplier λ and consider variations of the quantity

$$E(X, Y; \lambda) := R(X, Y) - \frac{1}{2}\lambda(q_0^2 + q_1^2 + q_2^2 + q_3^2) \quad (14)$$

with respect to both \mathbf{r} and q . Considering variations in \mathbf{r} first, we find that for an extremum:

$$\sum_{k=1}^N (\mathcal{U}(q)\mathbf{x}_k - \mathbf{y}_k + \mathbf{r}) = 0$$

so that

$$\mathbf{r} = \bar{\mathbf{y}} - \mathcal{U}(q)\bar{\mathbf{x}} := \frac{1}{N} \sum_{k=1}^N \mathbf{y}_k - \mathcal{U}(q) \frac{1}{N} \sum_{k=1}^N \mathbf{x}_k. \quad (15)$$

Shifting the two sets, X and Y , to their respective barycenters $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, we introduce:

$$\tilde{\mathbf{x}}_k := \mathbf{x}_k - \bar{\mathbf{x}}, \quad \tilde{\mathbf{y}}_k := \mathbf{y}_k - \bar{\mathbf{y}}.$$

Then, the minimization problem becomes:

$$\begin{aligned}
E(X, Y; \lambda) &= \frac{1}{2} \sum_{k=1}^N (\mathcal{U}(q) \tilde{\mathbf{x}}_k - \tilde{\mathbf{y}}_k)^2 - \frac{1}{2} \lambda q^2 \\
&= \frac{1}{2} \sum_{k=1}^N (\tilde{\mathbf{x}}_k^2 + \tilde{\mathbf{y}}_k^2) - \sum_{k=1}^N \tilde{\mathbf{y}}_k^T \mathcal{U}(q) \tilde{\mathbf{x}}_k - \frac{1}{2} \lambda (q_0^2 + q_1^2 + q_2^2 + q_3^2) .
\end{aligned} \tag{16}$$

In the sequel we drop the tildes (i.e. we will assume that both sets have been shifted to bring their respective barycenters to the origin) and we consider the model set,

$$X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) ,$$

and the target set

$$Y = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N) .$$

We treat the columns of X and Y as *pure quaternions*, i.e. $x_k := (0, \mathbf{x}_k)$ with $x_k^c = -x_k$ and similarly for y_k . Then the rotation $\mathcal{U}(q)$ on x_k is written as

$$(0, \mathcal{U}(q) \mathbf{x}_k) = qx_k q^c .$$

The residual (1) is written, in terms of quaternions, as

$$R(X, Y) = \frac{1}{2} \sum_{k=1}^N (qx_k q^c - y_k) (qx_k q^c - y_k)^c . \tag{17}$$

We wish to minimize $R(X, Y)$ subject to the constraint

$$qq^c = 1 . \tag{18}$$

Expanding, eq.(17) becomes:

$$\begin{aligned}
2R(X, Y) &= \sum_{k=1}^k ((qx_k q^c)(qx_k q^c)^c + y_k y_k^c - (qx_k q^c) y_k^c - y_k (qx_k q^c)^c) \\
&= \sum_{k=1}^k (x_k x_k^c + y_k y_k^c + (qx_k q^c) y_k + y_k (qx_k q^c))
\end{aligned} \tag{19}$$

where use has been made of the normalization $qq^c = 1$ and the property of pure quaternions $x^c = -x$. Since $qx_k q^c$ and y_k are pure and for a, b pure we have $ab + ba = 2(-\mathbf{a} \cdot \mathbf{b}, \mathbf{0}) = ([ab + ba]_0, \mathbf{0})$ the last two terms in eq.(19) can be combined as:

$$(qx_k q^c) y_k + y_k (qx_k q^c) = 2([y_k (qx_k q^c)]_0, \mathbf{0}) ,$$

i.e. only the 0-th component is non-zero. Using associativity for quaternions we have $y_k (qx_k q^c) = (y_k qx_k) q^c$ and if we define $z_k := y_k qx_k$ then

$$[z q^c]_0 = z_0 q_0 + \mathbf{z} \cdot \mathbf{q} =: q_{(4)}^T z_{(4)}$$

the last expression being the usual dot product for 4-vectors. Since we can write

$$z_k = y_k q x_k = \mathcal{A}_L(y_k) \mathcal{A}_R(x_k) q_{(4)} , \quad (20)$$

with $\mathcal{A}_L(y_k)$, $\mathcal{A}_R(x_k)$ defined as in eq.(10), we have finally the matrix vector expression:

$$[y_k(qx_kq^c)]_0 = q_{(4)}^T \mathcal{A}_L(y_k) \mathcal{A}_R(x_k) q_{(4)} . \quad (21)$$

Collecting results, we find that the residual can be written as

$$R(X, Y; q) = \frac{1}{2} \sum_{k=1}^N (\mathbf{x}_k^2 + \mathbf{y}_k^2) - q^T \mathcal{S}_N(X, Y) q \quad (22)$$

where

$$\mathcal{S}_N(X, Y) := \sum_{k=1}^N \mathcal{S}(\mathbf{x}_k, \mathbf{y}_k) := \sum_{k=1}^N \mathcal{A}_L(y_k) \mathcal{A}_R(x_k) .$$

The problem has in this way been reduced to that of finding the extrema of a quadratic form $q^T \mathcal{S} q$ in the four variables q_i , $i = 0, 1, 2, 3$ subject to the constraint $q^2 = 1$.

Introducing the Lagrange multiplier λ , we arrive at the expression

$$E(X, Y; \lambda) = \frac{1}{2} \sum_{k=1}^N (\mathbf{x}_k^2 + \mathbf{y}_k^2) - q^T \mathcal{S}_N(X, Y) q - \lambda q^T q \quad (23)$$

whose minimization leads to the eigenproblem:

$$\mathcal{S}_N(X; Y) q = \left(\sum_{k=1}^N \mathcal{S}(\mathbf{x}_k; \mathbf{y}_k) \right) q = \lambda q \quad (24)$$

The explicit form of the matrix \mathcal{S} is:

$$\mathcal{S}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} 0 & -y_1 & -y_2 & -y_3 \\ y_1 & 0 & -y_3 & y_2 \\ y_2 & y_3 & 0 & -y_1 \\ y_3 & -y_2 & y_1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -x_1 & -x_2 & -x_3 \\ x_1 & 0 & x_3 & -x_2 \\ x_2 & -x_3 & 0 & x_1 \\ x_3 & x_2 & -x_1 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} x_1 y_1 + x_2 y_2 + x_3 y_3 & x_2 y_3 - x_3 y_2 & x_3 y_1 - x_1 y_3 & x_1 y_2 - x_2 y_1 \\ x_2 y_3 - x_3 y_2 & x_1 y_1 - x_2 y_2 - x_3 y_3 & x_1 y_2 + x_2 y_1 & x_1 y_3 + x_3 y_1 \\ x_3 y_1 - x_1 y_3 & x_1 y_2 + x_2 y_1 & -x_1 y_1 + x_2 y_2 - x_3 y_3 & x_2 y_3 + x_3 y_2 \\ x_1 y_2 - x_2 y_1 & x_1 y_3 + x_3 y_1 & x_2 y_3 + x_3 y_2 & -x_1 y_1 - x_2 y_2 + x_3 y_3 \end{pmatrix} .$$

We see that the Lagrange multiplier λ is equal to one of the eigenvalues of a 4×4 symmetric, traceless matrix, and the corresponding eigenvector gives one of the candidate rotations that extremize the residual, as sought. We are thus led to the following expression for the minimal residual:

$$\min_q R = \frac{1}{2} \sum_{k=1}^N (\mathbf{x}_k^2 + \mathbf{y}_k^2) - \max_{i=1,2,3,4} \lambda_i$$

3 Appendix A

We give here MATLAB code for implementing the optimal match by a rotation of two ordered sets of vectors, whose barycenters have been shifted to the origin.

```
% This MATLAB program calculates the least-square RMSD between two
% rigid structures and the matrix and quaternion for the optimal rotation.
% The algorithm uses the formalism based on the quaternion algebra.
clear;
load model; % Read in data files named 'target' and 'model'.
load target; % Each file should be of the same length and in the format:
% x11 x12 x13 % x21 x22 x23 % . .
% . .
% xN1 xN2 xN3 % coordinate of the Nth atom
N = size(model,1); % Get the number of vectors
x_c = sum(model) ./ N; % Find the barycenters
y_c = sum(target) ./ N;
x = model - ones(N,1) * x_c; % Transform to the centroidal coordinates
y = target - ones(N,1) * y_c;
norm_x = sum(sum(x .* x)); % Calculate the norms
norm_y = sum(sum(y .* y));
xT = x'; % Calculate the R matrix
R = zeros(3,3);
for i = 1 : N
R = R + xT(:,i)*y(i,:);
end
% S matrix
S = [R(1,1)+R(2,2)+R(3,3) R(2,3)-R(3,2) R(3,1)-R(1,3) R(1,2)-R(2,1); ...
R(2,3)-R(3,2) R(1,1)-R(2,2)-R(3,3) R(1,2)+R(2,1) R(1,3)+R(3,1); ...
R(3,1)-R(1,3) R(1,2)+R(2,1) -R(1,1)+R(2,2)-R(3,3) R(2,3)+R(3,2); ...
R(1,2)-R(2,1) R(1,3)+R(3,1) R(2,3)+R(3,2) -R(1,1)-R(2,2)+R(3,3)];
[V, D] = eig(S);
quaternion = V(:,4)
q0 = V(1,4);
q1 = V(2,4);
q2 = V(3,4);
q3 = V(4,4);
AR = [ q0 -q1 -q2 -q3; q1 q0 q3 -q2; q2 -q3 q0 q1; q3 q2 -q1 q0];
AL = [ q0 q1 q2 q3;-q1 q0 q3 -q2;-q2 -q3 q0 q1;-q3 q2 -q1 q0];
U = AL * AR;
U(:,1) = [];
U(1,:) = []
RMSD = sqrt((0.5 * (norm_x + norm_y) - D(4,4)) / N)
clear;
```

4 Appendix B

Kabsch [4] finds the extremal of the residual to be equal to

$$R(X, Y)_{\min} = \frac{1}{2} \sum_{k=1}^N (\mathbf{x}_k^2 + \mathbf{y}_k^2) - \sum_{j=1}^3 \sigma_j \sqrt{\mu_j}$$

where $\sigma_j = \pm 1$ with $\sigma_1 \sigma_2 \sigma_3 = 1$. Here the μ_j are the eigenvalues of the symmetric, positive definite matrix $\mathcal{R}\mathcal{R}^T$ with

$$\mathcal{R} := \sum_{k=1}^N \mathbf{x}_k \mathbf{y}_k^T \rightarrow R_{ij} = \sum_{k=1}^N x_{ki} y_{kj}, \quad i, j = 1, 2, 3$$

The 9 quantities appearing in matrix \mathcal{R} enter both in

$$\mathcal{R}\mathcal{R}^T = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{pmatrix} R_{11} & R_{21} & R_{31} \\ R_{12} & R_{22} & R_{32} \\ R_{13} & R_{23} & R_{33} \end{pmatrix}$$

and in the matrix $\mathcal{S}_N(X, Y) =$

$$\begin{pmatrix} R_{11} + R_{22} + R_{33} & R_{23} - R_{32} & R_{31} - R_{13} & R_{12} - R_{21} \\ R_{23} - R_{32} & R_{11} - R_{22} - R_{33} & R_{12} + R_{21} & R_{13} + R_{31} \\ R_{31} - R_{13} & R_{12} + R_{21} & -R_{11} + R_{22} - R_{33} & R_{23} + R_{32} \\ R_{12} - R_{21} & R_{13} + R_{31} & R_{23} + R_{32} & -R_{11} - R_{22} + R_{33} \end{pmatrix}$$

what is needed in order to show equivalence of the methods is that the set of eigenvalues of \mathcal{S}_N , λ_i with $i = 1, 2, 3, 4$ is the same as the set of values $\rho_{\sigma_1, \sigma_2, \sigma_3} := \sum_{j=1}^3 \sigma_j \sqrt{\mu_j}$. The following MATLAB script demonstrates (but does not prove!) that the two sets are identical. A proof should be easy to find, but we have not yet succeeded...

```

RR = rand(3,3);
AA = RR*RR';
QQ =
[R(1,1)+R(2,2)+R(3,3)  R(2,3)-R(3,2)  R(3,1)-R(1,3)  R(1,2)-R(2,1);...
 R(2,3)-R(3,2)  R(1,1)-R(2,2)-R(3,3)  R(1,2)+R(2,1)  R(1,3)+R(3,1);...
 R(3,1)-R(1,3)  R(1,2)+R(2,1)  R(2,2)-R(3,3)-R(1,1)  R(2,3)+R(3,2);...
 R(1,2)-R(2,1)  R(1,3)+R(3,1)  R(2,3)+R(3,2)  R(3,3)-R(1,1)-R(2,2)];

E = eig(AA);
E1 = sqrt(E);

a = E1(1)+E1(2)+E1(3)
b = E1(1)-E1(2)-E1(3)
c = E1(2)-E1(3)-E1(1)
d = E1(3)-E1(1)-E1(2)

E2 = eig(QQ)

```

References

- [†] vageli@maxwell.ucsf.edu; permanent address: Dept. of Mathematics and Statistics, University of New Mexico, Albuquerque, NM. 87131.
- [‡] chaok@maxwell.ucsf.edu
- [1] E. A. Coutsias and L. Romero, *The Quaternions with applications to Rigid Body Dynamics*, notes from Dynamical Systems Seminar, Dept. of Mathematics and Statistics and Center for Advanced Study, Univ. of New Mexico, Jan. 1999.
- [2] Diamond, R., *A note on the rotational superposition problem*, Acta Crystallographica, A44(2), 211-216, (1988).
- [3] Flower, D.L., *Rotational superposition: a review of methods*, Journal of Computational Graphics and modelling, 17(3-4), 238-244, 1999.
- [4] Kabsch, W., Acta Cryst.(1978), A34, 827-828.
- [5] A. Neumaier, *Molecular modeling of proteins and mathematical prediction of protein structure*, SIAM Rev. 39(3), 407-460, (1997).
- [6] Rappaport, D.C., *The Art of Molecular Dynamics Simulation*, Cambridge Univ. Press, New York, Chapter 8, pp. 191-221, 1995.
- [7] Rosen, J.B., A.T. Phillips, S.Y. Oh and K.A. Dill, *A Method for Parameter Optimization in Computational Biology*, preprint, May 2000.