On sufficient statistics of least-squares superposition of vector sets

Arun S. Konagurthu[1], Parthan Kasarapu[1], Lloyd Allison[1], James H. Collier[1], Arthur M. Lesk[2]

[1]Faculty of Information Technology, Monash University, Clayton VIC 3800 Australia,
[2]The Huck Institute of Genomics, Proteomics and Bioinformatics and the Department of Biochemistry and Molecular Biology, Pennsylvania State University, University Park PA 16802 USA
Correspondence: arun.konagurthu@monash.edu

Abstract. Superposition by orthogonal transformation of vector sets by minimizing the least-squares error is a fundamental task in many areas of science, notably in structural molecular biology. Its widespread use for structural analyses is facilitated by exact solutions of this problem, computable in linear time. However, in several of these analyses it is common to invoke this superposition routine a very large number of times, often operating (through addition or deletion) on previously superposed vector sets. This paper derives a set of *sufficient statistics* for the least-squares orthogonal transformation problem. These sufficient statistics are additive. This property allows for the superposition parameters (rotation, translation, and root mean square deviation) to be computable as *constant time* updates from the statistics of partial solutions. We demonstrate that this results in a massive speed up in the computational effort, when compared to the method that recomputes superpositions *ab initio*. Among others, protein structural alignment algorithms stand to benefit from our results.

1 Introduction

Optimal superposition through orthogonal transformation of vector sets forms the linchpin of macromolecular structure comparison [1, 2]. This task is ubiquitously used to analyse globular three-dimensional structures of proteins [3]. Orthogonal transformation involves finding the best rigid-body rotation and translation of two vector sets that are in one-to-one correspondence so that they can be superimposed. This superposition immediately provides a qualitative (through visual inspection) as well as a quantitative measure of shape similarity.

An almost universally used criterion to define the *best* superposition of vector sets is the one that minimizes the *sum of square errors* over the entire search space of possible rotations and translations. This results in a quantitative measure, *root mean square deviation* (or r.m.s.d.) after best superposition. This measure is central in assessing the quality of superposition with attractive metrical properties.

Superpositions pervade protein structural analyses because they provide essential information about comparisons of conformations of structures and substructures; it is remarkable and comes in handy that optimal superposition of aligned sets of points can be computed exactly and efficiently [3]. Given the importance of this routine, several approaches have been proposed to address this problem over the years [4–14]. However, among the most-widely used approach to solve this problem is the method of Kabsch [5] that solves this problem using Lagrange multipliers that constrain the search to *pure rotations* (and avoid improper ones).

An equivalent, but a more elegant, approach to solving the same problem was proposed by Kearsley [11] using the mathematical object called *quaternions* [15]. Quaternions are generalizations of complex numbers with direct applications to transformations in three dimensional space. Specifically, the space group corresponding to unit quaternions is equivalent to the group of all possible pure rotations in three dimensions (3D) defined about an arbitrary origin. That is, any 3D pure rotation by an angle θ about some normalized axis $\hat{\boldsymbol{n}}$ passing through the origin can be represented using a unit quaternion as follows: $\left[\cos\left(\frac{\theta}{2}\right), \hat{\boldsymbol{n}}\sin\left(\frac{\theta}{2}\right)\right]$. Among the key advantages of using Kearsley's quaternion method to solve the least-squares superposition problem are: (1) the problem can be solved analytically in quaternion parameters, and (2) the method avoids problems with singularities (and rotoinversions) that can result from using Kabsch's approach, where these oddities are handled explicitly after the solution is found [11, 13]. In general, the least-squares superposition involves a computational effort that asymptotically grows *linearly* with the number of corresponding points being superimposed.

Many methods that facilitate analyses involving protein structures employ least-squares superpositions. Among the primary example of this is when computing the residue-residue correspondences betweeen two or more protein structures – *the structural alignment problem*. Many popular methods build an alignment between structures using orthogonal superpositions of fragments [9, 16–23]. The general strategy involves finding aligned (contiguous) fragment pairs that are often maximally extended, one residue-residue correspondence at a time starting from some minimum fragment size, until the fragment pairs superposes within some specified threshold of r.m.s.d. This results in a library of well-fitting fragment pairs, construction effort of which grows as a cubic in the length of the structures being aligned ($O(n^2)$ number of superpositions, each taking $O(n)$ superposition effort, where n is the number of residues in the structures being aligned). Further, by computing the joint superpositions of these well-fitting maximal fragment pairs, a structural alignment is *assembled* by collecting fragment pairs that superpose consistently. This involves repeated concatenation and superposition calls using the fragment pairs in the library. Such superpositions are currently recomputed from scratch (even though the previous superpositions provide a wealth of information about the joint superposition, as we shall demonstrate in the forthcoming sections). It can be seen that the number of joint

superpositions grows (at least) quadratically in the size of the fragment library, with each joint superposition taking a linear effort in the size of the concatenated vector sets.

Although the optimal solution of the least-squares superposition problem can be computed extremely efficiently, the algorithmic complexity term hides a sizeable constant factor. This imposes a significant computational demand when performing a large number of superpositions, as required for computing pairwise structural alignments. The amount of time spent in superposing fragments quickly becomes computationally impractical when aligning multiple protein structures simultaneously, where the multiple structural alignment is commonly built using all-vs-all pairwise structural alignments, each of which makes a very large number of calls to the superposition routine.

Contribution of this work: In this paper we explore the theoretical underpinning of the orthogonal superposition problem and derive a set of statistics that are sufficient to compute the r.m.s.d of best superposition, and its corresponding rotation and translation parameters). We demonstrate that these *sufficient statistics* [24] are additive. Thus these statistics can be used to compute new superpositions as *constant time* updates using the statistics of the partial solutions. Using such an approach results in a drastic speed up in comparison with the approach that recomputes the new superposition from scratch.

Organization of this paper: Section 2 gives the basic background of the orthogonal superposition problem using the widely-used least-squares criterion. Section 3 introduces the statistical aspects of sufficient statistics, and derives the full set of sufficient statistics for the optimal orthogonal superposition problem. Section 4.1 provides the mechanics of performing constant-time updates to superpositions building on the sufficient statistics of previous (partial) superpositions. Section 5 describes an approach to speed up the diagonalization step used in the Kearsley approach. Finally, the paper ends with an experimental evaluation of computing optimal superpositions using sufficient statistics.

2 Orthogonal superposition

Formally let $\mathcal{U} = \{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n\}$ and $\mathcal{V} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n\}$ denote two vector sets with one-to-one correspondence. In this paper we consider vectors in three dimensions. Let the (x, y, z) components of each $\boldsymbol{u}_i$ be represented here as $(\boldsymbol{u}_i(x), \boldsymbol{u}_i(y), \boldsymbol{u}_i(z))$. (Similar representation holds for $\boldsymbol{v}_i$ or any other vector.)

The rigid-body least-squares superposition problem is a constrained optimization problem that involves finding the best rotation (matrix) $\mathbf{R}$ and translation (vector) $\boldsymbol{t}$ with the optimality criterion defined as:

$$\mathcal{E} = \min |\mathbf{R}\mathcal{U} + \boldsymbol{t} - \mathcal{V}|^2 = \min \sum_{i=1}^{n} |\mathbf{R}\boldsymbol{u}_i + \boldsymbol{t} - \boldsymbol{v}_i|^2 = \min \sum_{i=1}^{n} \langle \mathbf{R}\boldsymbol{u}_i + \boldsymbol{t} - \boldsymbol{v}_i, \mathbf{R}\boldsymbol{u}_i + \boldsymbol{t} - \boldsymbol{v}_i \rangle$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product between the stated terms, $\mathbf{R}$ is a 3×3 pure rotation matrix, and $\boldsymbol{t}$ is a translation vector.

Under this least-squares criterion, the translation with respect to the optimal superposition is independent of rotation. This can be easily seen by differentiating $\mathcal{E}$ with respect to $\boldsymbol{t}$ and evaluating it at its extremum:

$$\begin{aligned}
\frac{\partial \mathcal{E}}{\partial \boldsymbol{t}} &= \frac{\partial}{\partial \boldsymbol{t}} \sum_{i=1}^{n} \langle \mathbf{R}\boldsymbol{u}_i + \boldsymbol{t} - \boldsymbol{v}_i, \mathbf{R}\boldsymbol{u}_i + \boldsymbol{t} - \boldsymbol{v}_i \rangle = \sum_{i=1}^{n} 2 \frac{\partial (\mathbf{R}\boldsymbol{u}_i + \boldsymbol{t} - \boldsymbol{v}_i)}{\partial \boldsymbol{t}} (\mathbf{R}\boldsymbol{u}_i + \boldsymbol{t} - \boldsymbol{v}_i) = 0 \\
&\implies \sum_{i=1}^{n} \mathbf{R}\boldsymbol{u}_i + \boldsymbol{t} - \boldsymbol{v}_i = 0 \\
&\implies \boldsymbol{t} = \frac{\sum_{i=1}^{n} \boldsymbol{v}_i}{n} - \mathbf{R}\frac{\sum_{i=1}^{n} \boldsymbol{u}_i}{n} = \textbf{Centroid}(\mathcal{V}) - \mathbf{R}\ \textbf{Centroid}(\mathcal{U})
\end{aligned}$$

It follows that moving each of the vector sets to an origin at its centroid, about which the rotation is defined, gives us a modified (but equivalent) objective which is independent of the translation $\boldsymbol{t}$:

$$\mathcal{E} = \min \sum_{i=1}^{n} |\mathbf{R}\boldsymbol{u}_i' - \boldsymbol{v}_i'|^2$$

where, $\boldsymbol{u_i}' = \boldsymbol{u_i} - \frac{\sum_{i=1}^{n} \boldsymbol{u_i}}{n}$ and $\boldsymbol{v_i}' = \boldsymbol{v}_i - \frac{\sum_{i=1}^{n} \boldsymbol{v}_i}{n}$.

Kearsley [11] proposed an elegant method that removes the non-linear aspect to this the least-squares problem and transforms it to an eigenvalue problem of the form $\mathbf{Q}\boldsymbol{q} = \lambda \boldsymbol{q}$, where $\mathbf{Q}$ is a 4×4 square symmetric matrix

$$\begin{pmatrix}
\sum(x_m^2 + y_m^2 + z_m^2) & \sum(y_p z_m - y_m z_p) & \sum(x_m z_p - x_p z_m) & \sum(x_p y_m - x_m y_p) \\
\sum(y_p z_m - y_m z_p) & \sum(x_m^2 + y_p^2 + z_p^2) & \sum(x_m y_m - x_p y_p) & \sum(x_m z_m - x_p z_p) \\
\sum(x_m z_p - x_p z_m) & \sum(x_m y_m - x_p y_p) & \sum(x_p^2 + y_m^2 + z_p^2) & \sum(y_m z_m - y_p z_p) \\
\sum(x_p y_m - x_m y_p) & \sum(x_m z_m - x_p z_p) & \sum(y_m z_m - y_p z_p) & \sum(x_p^2 + y_p^2 + z_m^2)
\end{pmatrix}, \quad (1)$$

$\boldsymbol{q} = (q_1, q_2, q_3, q_4)^T = \left(\cos\left(\frac{\theta}{2}\right), \hat{\boldsymbol{n}}(x) \sin\left(\frac{\theta}{2}\right), \hat{\boldsymbol{n}}(y) \sin\left(\frac{\theta}{2}\right), \hat{\boldsymbol{n}}(z) \sin\left(\frac{\theta}{2}\right) \right)^T$ are the (unknown or to be solved) quaternion components associated with some rotation θ about a normalized axis $\hat{n}$, and λ is an (unknown) eigenvalue. In Equation 1, we use the notation x_m to denote the component-wise difference $\boldsymbol{v}_i'(x) - \boldsymbol{u}_i'(x)$ (and similarly y_m and z_m) and x_p to denote the component-wise sum $\boldsymbol{v}_i'(x) + \boldsymbol{u}_i'(x)$ (similarly y_p and z_p). From this point onwards, we use the term *quaternion matrix* to indicate the 4×4 square symmetric matrix in Equation 1 and denote it as $\mathbf{Q}$.

Diagonalizing this matrix yields four eigenvalues and (corresponding) eigenvectors. The eigenvector corresponding to the smallest eigenvalue, $\lambda_{\min}$, corresponds to the rotation producing the least-squares error, and the r.m.s.d is computed as $\sqrt{\frac{\lambda_{\min}}{n}}$

Time complexity The computational effort that takes to solve the rigid-body superposition problem using Kearsley's quaternion approach (or equivalently Kabsch's approach) grows linearly with the number of vectors being superimposed. In Kearsley's approach this is dominated by the computation of the $\mathbf{Q}$ where each of 10 distinct terms in the matrix requires $O(n)$ effort. The diagonalization of $\mathbf{Q}$ is independent of n and shows a rapid convergence with numerical methods such as Jacobi's diagonalization algorithm [25].

3 Sufficient Statistics

We note that this rigid-body superposition problem is a geometric instance of the general regression problem using total least-squares, where a regression line is determined that minimizes the sum of the squared errors of the observed data with respect to it.

It is widely known that solution of the regression problem produces error terms that are normally distributed as $\mathcal{N}(0, \sigma)$ where the mean μ is 0 and σ is the standard deviation which is minimized by the problem. In fact, the least squares estimator of σ is also its maximum likelihood estimator.

More formally, consider the standard normal distribution of some random variable x:

$$\mathcal{N}(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

This normal density can be reparameterized into a general form denoting the family of exponential distributions:

$$f(x|\boldsymbol{\eta}) = h(x)g(\boldsymbol{\eta}) \exp(\boldsymbol{\eta}^T \boldsymbol{U}(x))$$

where $h(x) = \frac{1}{\sqrt{\pi}}$, $g(\eta_2) = \sqrt{-\eta_2} \exp\left(\frac{\eta_1^2}{4\eta_2}\right)$, $\boldsymbol{\eta}^T = (\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2})$, $\boldsymbol{U}^T(x) = (x, x^2)$.

This transformation can be used to show certain important properties that allows efficient computation of maximum likelihood estimators of μ and σ.

Considering a sample set of observations that are normally distributed $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}$. The likelihood for these samples is given by:

$$f(\mathbf{X}|\boldsymbol{\eta}) = \left(\prod_{i=1}^{n} h(x_i)\right) (g(\boldsymbol{\eta}))^n \exp(\boldsymbol{\eta}^T \sum_{i=1}^{n} u(x_i))$$

Taking natural logarithms on both sides gives us the log likelihood:

$$\log(f(\mathbf{X}|\boldsymbol{\eta})) = \kappa + n \log(g(\boldsymbol{\eta})) + \boldsymbol{\eta}^T \sum_{i=1}^{n} \boldsymbol{U}(x_i)$$

where $\kappa = \sum_{i=1}^{n} \log(h(x_i))$ is a term independent of $\boldsymbol{\eta}$.

To find the maximum likelihood estimators $\hat{\boldsymbol{\eta}}$, take the gradient with respect to $\boldsymbol{\eta}$ and set to 0. This results in:

$$
\begin{aligned}
& n\nabla_{\hat{\boldsymbol{\eta}}}\left[\log\left(g(\hat{\boldsymbol{\eta}})\right)\right] + \sum_{i=1}^{n}\boldsymbol{U}(x_i) = 0 \\
\implies & -\nabla_{\hat{\boldsymbol{\eta}}}\left[\log\left(g(\hat{\boldsymbol{\eta}})\right)\right] = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{U}(x_i) \\
& = \frac{-1}{g(\hat{\boldsymbol{\eta}})}\nabla_{\hat{\boldsymbol{\eta}}}g(\hat{\boldsymbol{\eta}}) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{U}(x_i)
\end{aligned}
$$

Notice that maximum likelihood estimate $\hat{\boldsymbol{\eta}}$ depends on the statistic $\sum_{i=1}^{n}\boldsymbol{U}(x_i)$ rather than the individual data. This suggests that to obtain the maximum likelihood estimate we do not need the data explicitly as it can be derived from that statistic. This sufficiency to derive the maximum likelihood estimator without explicit consideration of data makes $\sum_{i=1}^{n}\boldsymbol{U}(x_i)$ a *sufficient statistic* for the exponential family of functions. For normal distribution, we saw earlier that $\boldsymbol{U}(x_i) = (x_i, x_i^2)$ gives the sufficient statistics of $\sum_{i=1}^{n}x_i$ and $\sum_{i=1}^{n}x_i^2$ [24].

Sufficient statistics for orthogonal superposition

We note that each error term, $\varepsilon_i = \mathbf{R}\boldsymbol{u_i}' - \boldsymbol{v_i}'$, is assumed to be normally distributed: *i.e.,* $\varepsilon_i \sim \mathcal{N}(\mu = 0, \sigma)$. We now derive the sufficient statistics for σ of ε_is, which is equivalent to the r.m.s.d. after least-squares superposition. The likelihood of the observed normally distributed errors after superposition, $\mathbf{E} = \{\varepsilon_1, \ldots, \varepsilon_n\}$, can be written as:

$$
\begin{aligned}
f(\varepsilon_1, \ldots, \varepsilon_n|\sigma) &= \prod_{i=1}^{n}(2\pi\sigma^2)^{-\frac{1}{2}}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{R}\boldsymbol{u_i}' - \boldsymbol{v_i}'\|^2\right) \\
&= (2\pi\sigma^2)^{-\frac{n}{2}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\|\mathbf{R}\boldsymbol{u_i}' - \boldsymbol{v_i}'\|^2\right) \qquad (2)
\end{aligned}
$$

Let's examine the decomposition of

$$
\varepsilon_i^2 = \|\mathbf{R}\boldsymbol{u_i}' - \boldsymbol{v_i}'\|^2 = \|\boldsymbol{u_i}'\|^2 + \|\boldsymbol{v_i}'\|^2 - 2\boldsymbol{v_i}'^{\mathrm{T}}\mathbf{R}\boldsymbol{u_i}' \qquad (3)
$$

From Equation 1, the matrix $\mathbf{Q}$ is made up of terms of the form

$$
A_m = v_i'(A) - u_i'(A) \text{ and } A_p = v_i'(A) + u_i'(A)
$$

where each A and B take the values $\{x, y, z\}$ denoting vector components. Rewriting, we get

$$
v_i'(A) = \frac{A_p + A_m}{2} \text{ and } u_i'(A) = \frac{A_p - A_m}{2}
$$

The first two terms on the right hand side of Equation 3 can be expanded as follows:

$$\begin{aligned}\|\boldsymbol{u_i}'\|^2 + \|\boldsymbol{v_i}'\|^2 &= (u_i'(x)^2 + u_i'(y)^2 + u_i'(z)^2) + (v_i'(x)^2 + v_i'(y)^2 + v_i'(z)^2) \\ &= \frac{1}{2}(x_m^2 + x_p^2 + y_m^2 + y_p^2 + z_m^2 + z_p^2) \\ &= \frac{1}{2}\sum_{A\in\{x,y,z\}} A_m^2 + \frac{1}{2}\sum_{A\in\{x,y,z\}} A_p^2 \end{aligned} \tag{4}$$

The last term on the right hand side of Equation 3 can be expanded as $\boldsymbol{v_i}'^{\mathrm{T}}\mathbf{R}\boldsymbol{u_i}' = \boldsymbol{v_i}'^{\mathrm{T}}[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{r_3}]\boldsymbol{u_i}'$ where $\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}$ are column vectors of the 3×3 rotation matrix $\mathbf{R}$. Therefore,

$$\boldsymbol{v_i}'^{\mathrm{T}}\mathbf{R}\boldsymbol{u_i}' = (\boldsymbol{v_i}'.\mathbf{r_1})u_i'(x) + (\boldsymbol{v_i}'.\mathbf{r_2})u_i'(y) + (\boldsymbol{v_i}'.\mathbf{r_3})u_i'(z) \tag{5}$$

Take the first term on the right hand side of Equation 5. This can be expanded as:

$$\begin{aligned}(\boldsymbol{v_i}'.\mathbf{r_1})u_i'(x) &= r_{11}v_i'(x)u_i'(x) + r_{12}v_i'(y)u_i'(x) + r_{13}v_i'(z)u_i'(x) \\ &= \frac{r_{11}}{4}(x_p + x_m)(x_p - x_m) + \frac{r_{12}}{4}(y_p + y_m)(x_p - x_m) + \frac{r_{13}}{4}(z_p + z_m)(x_p - x_m) \\ &= \frac{r_{11}}{4}(x_p^2 - x_m^2) + \frac{r_{12}}{4}(y_p x_p - y_p x_m + y_m x_p - y_m x_m) \\ &\quad + \frac{r_{13}}{4}(z_p x_p - z_p x_m + z_m x_p - z_m x_m)\end{aligned}$$

where r_{11}, r_{12}, r_{13} are the terms in the $\boldsymbol{r}_1$ column vector in $\mathbf{R}$. More generally,

$$(\boldsymbol{v_i}'.\mathbf{r_1})u_i'(x) = c_1A_p^2 + c_2A_m^2 + c_3A_pB_p + c_4A_mB_m + c_5A_mB_p \tag{6}$$

where c_k are constants in terms of components of $\boldsymbol{r}_1$.

Similarly, $(\boldsymbol{v_i}'.\mathbf{r_2})u_i'(y)$ and $(\boldsymbol{v_i}'.\mathbf{r_3})u_i'(z)$ can be expanded as above and will have the same form as (6) but with different constants. Therefore, combining Equations 4-5, the equation 3 can be written as

$$\varepsilon_i^2 = \zeta_1\sum_A A_p^2 + \zeta_2\sum_A A_m^2 + \zeta_3\sum_{\forall A\neq B} A_pB_p + \zeta_4\sum_{\forall A\neq B} A_mB_m + \zeta_5\sum_{\forall A\neq B} A_mB_p$$

where ζ_k are constants. Hence, the likelihood function can be written as

$$f(\varepsilon_1,\ldots,\varepsilon_n|\sigma) = (2\pi\sigma^2)^{-\frac{n}{2}}\exp\left(-\frac{1}{2\sigma^2}\mathbf{U}\right) \tag{7}$$

where

$$\mathbf{U} = \sum_{i=1}^{n}\left(\zeta_1\sum_A A_p^2 + \zeta_2\sum_A A_m^2 + \zeta_3\sum_{\forall A\neq B} A_pB_p + \zeta_4\sum_{\forall A\neq B} A_mB_m + \zeta_5\sum_{\forall A\neq B} A_mB_p\right)$$

and $A, B \in \{x, y, z\}$

Using Equation 7, the negative log-likelihood is given as:

$$\mathcal{L}(\varepsilon_1, \ldots, \varepsilon_n|\sigma) = \frac{n}{2}\log(2\pi) + n\log\sigma + \frac{1}{2\sigma^2}\mathbf{U} \tag{8}$$

The maximum likelihood estimate $\hat{\sigma}$ can be determined by minimising Equation 8 and evaluating the corresponding σ, *i.e.*

$$\frac{\partial\mathcal{L}}{\partial\sigma} = 0 \implies \hat{\sigma}^2 = \frac{\mathbf{U}}{n} \tag{9}$$

$\mathbf{U}$ involve statistics that do not take into account the data explicitly, and are sufficient to estimate σ (or r.m.s.d). Therefore the set of *sufficient statistics* for the least-squares superposition problem can be defined as:

$$\mathbf{\Psi} = \left\{ \sum_{i=1}^{n} A_m, \quad \sum_{i=1}^{n} A_p, \quad \sum_{i=1}^{n} A_m B_m, \quad \sum_{i=1}^{n} A_m B_p, \quad \sum_{i=1}^{n} A_p B_p \right\} \tag{10}$$

where A and B take the values $\{x, y, z\}$, $A_m = \boldsymbol{v_i}'(A) - \boldsymbol{u_i}'(A)$ is the component-wise difference (similarly B_m), and $A_p = \boldsymbol{v_i}'(A) + \boldsymbol{u_i}'(A)$ is the component-wise sum (similarly B_p). Altogether, the set $\mathbf{\Psi}$ consists of 24 distinct statistics.

In addition, using the same notation, the statistics required to compute the centroid are of the form $\sum_{i=1}^{n} \boldsymbol{u_i}'(A)$ and $\sum_{i=1}^{n} \boldsymbol{v_i}'(A)$, and these are equivalent to $\sum_{\forall A} A_m$ and $\sum_{\forall A} A_p$.

4 Updating sufficient statistics

4.1 Addition operation on vector sets using sufficient statistics

Consider two pairs of corresponding vector sets: $\mathcal{Q} \leftrightarrow \mathcal{R}$ containing n_1 correspondences and $\mathcal{S} \leftrightarrow \mathcal{T}$ containing n_2 correspondences. Let $\mathcal{U}$ be defined as a combination of vectors $\mathcal{Q}$ and $\mathcal{S}$) and similarly $\mathcal{V}$ as a combination of $\mathcal{R}$ and $\mathcal{T}$. Let $\mathbf{\Psi}_1$ denote the sufficient statistics of superposing the first pair and $\mathbf{\Psi}_2$ denote the same for the second pair. Define these as:

$$\mathbf{\Psi}_1 = \left\{ \sum_{i=1}^{n_1} C_m, \sum_{i=1}^{n_1} C_p, \sum_{i=1}^{n_1} C_m D_m, \sum_{i=1}^{n_1} C_m D_p, \sum_{i=1}^{n_1} C_p D_p \right\} \tag{11}$$

$$\mathbf{\Psi}_2 = \left\{ \sum_{i=1}^{n_2} E_m, \sum_{i=1}^{n_2} E_p, \sum_{i=1}^{n_2} E_m F_m, \sum_{i=1}^{n_2} E_m F_p, \sum_{i=1}^{n_2} E_p F_p \right\} \tag{12}$$

Where C, D, E and F are all either $\{x, y, z\}$ denoting the components of the corresponding vectors in the vector sets under consideration. Consistent with

the previous notation (see Equation 10), C_p and C_m (similarly D_p and D_m) are the component-wise sums and differences between corresponding vectors in $\mathcal{Q}$ and $\mathcal{R}$. The same definitions hold for E_m (and E_p) and F_m (and F_p), with respect to corresponding vectors in $\mathcal{S}$ and $\mathcal{T}$.

We want to use $\mathbf{\Psi_1}$ and $\mathbf{\Psi_2}$ to compute a new set of sufficient statistics $\mathbf{\Psi}$ (defined in Equation 10) for the superposition of vector sets $\mathcal{U} = \mathcal{Q} + \mathcal{S}$ with $\mathcal{V} = \mathcal{R} + \mathcal{T}$. Below we derive the construction of the new sufficient statistics.

The statistics involved in computing the new centroids of the sets $\mathcal{U}$ and $\mathcal{V}$, $\sum_{i=1}^{n=n_1+n_2} \boldsymbol{u}(A)$ and $\sum_{i=1}^{n=n_1+n_2} \boldsymbol{v}(A)$, can be trivially updatated using the statistics $\sum_{i=1}^{n_1} \boldsymbol{q}(C)$, $\sum_{i=1}^{n_1} \boldsymbol{r}(D)$, $\sum_{i=1}^{n_2} \boldsymbol{s}(E)$, and $\sum_{i=1}^{n_2} \boldsymbol{t}(F)$.

To compute the remaining statistics in $\mathbf{\Psi}$, define vectors:

$$\begin{array}{ll} \boldsymbol{\alpha}_1 = \textbf{Centroid}(\mathcal{U}) - \textbf{Centroid}(\mathcal{Q}) & \boldsymbol{\beta}_1 = \textbf{Centroid}(\mathcal{V}) - \textbf{Centroid}(\mathcal{R}) \\ \boldsymbol{\alpha}_2 = \textbf{Centroid}(\mathcal{U}) - \textbf{Centroid}(\mathcal{S}) & \boldsymbol{\beta}_2 = \textbf{Centroid}(\mathcal{V}) - \textbf{Centroid}(\mathcal{T}). \end{array}$$

These vectors define the corrections that are required to be made to the previous centroids to recover the updated ones.

Lemma 1. $\sum_{i=1}^{n=n_1+n_2} A_m = \left[\sum_{i=1}^{n_1} C_m + n_1 \Delta_m^C\right] + \left[\sum_{i=1}^{n_2} E_m + n_2 \Delta_m^E\right]$ *, where* $\Delta_m^C = \boldsymbol{\beta_1}(C) - \boldsymbol{\alpha_1}(C)$ *and* $\Delta_m^E = \boldsymbol{\beta_2}(E) - \boldsymbol{\alpha_2}(E)$ *and* $A = C = E \in \{x, y, z\}$

Proof.

$$\begin{aligned} \sum_{i=1}^{n=n_1+n_2} A_m &= \left[\sum_{i=1}^{n_1} [(\boldsymbol{r}'(C) + \boldsymbol{\beta_1}(C)) - (\boldsymbol{q}'(C) + \boldsymbol{\alpha_1}(C))]\right] \\ &\quad + \left[\sum_{i=1}^{n_2} [(\boldsymbol{t}'(E) + \boldsymbol{\beta_2}(E)) - (\boldsymbol{s}'(E) + \boldsymbol{\alpha_2}(E))]\right] \\ &= \left[\sum_{i=1}^{n_1} (\boldsymbol{r}'(C) - \boldsymbol{q}'(C)) + (\boldsymbol{\beta_1}(C) - \boldsymbol{\alpha_1}(C))\right] \\ &\quad + \left[\sum_{i=1}^{n_2} (\boldsymbol{t}'(C) - \boldsymbol{s}'(C)) + (\boldsymbol{\beta_2}(E) - \boldsymbol{\alpha_2}(E))\right] \\ &= \left[\sum_{i=1}^{n_1} C_m + \sum_{i=1}^{n_1} \Delta_m^C\right] + \left[\sum_{i=1}^{n_2} E_m + \sum_{i=1}^{n_2} \Delta_m^E\right] \\ &= \left[\sum_{i=1}^{n_1} C_m + n_1 \Delta_m^C\right] + \left[\sum_{i=1}^{n_2} E_m + n_2 \Delta_m^E\right] \end{aligned}$$

Corollary 1. $\sum_{i=1}^{n} A_p = \left[\sum_{i=1}^{n_1} C_p + n_1 \Delta_p^C\right] + \left[\sum_{i=1}^{n_2} E_p + n_2 \Delta_p^E\right]$

Lemma 2.

$$\sum_{i=1}^{n=n_1+n_2} A_m B_m = \left[\sum_{i=1}^{n_1} C_m D_m + \Delta_m^C \sum_{i=1}^{n_1} D_m + \Delta_m^D \sum_{i=1}^{n_1} C_m + n_1 \Delta_m^C \Delta_m^D\right] + \left[\sum_{i=1}^{n_2} E_m F_m + \Delta_m^E \sum_{i=1}^{n_2} F_m + \Delta_m^F \sum_{i=1}^{n_2} E_m + n_2 \Delta_m^E \Delta_m^F\right]$$

where $\Delta_m^C = \boldsymbol{\beta_1}(C) - \boldsymbol{\alpha_1}(C)$, $\Delta_m^D = \boldsymbol{\beta_1}(D) - \boldsymbol{\alpha_1}(D)$, $\Delta_m^E = \boldsymbol{\beta_2}(E) - \boldsymbol{\alpha_2}(E)$, *and* $\Delta_m^F = \boldsymbol{\beta_2}(F) - \boldsymbol{\alpha_2}(F)$ $A = C = E \in \{x, y, z\}$ *and* $B = D = F \in \{x, y, z\}$

Proof.

$$\text{Updated}(\sum_{i=1}^{n_1} C_m D_m) = \sum_{i=1}^{n_1} [(\boldsymbol{r}'(C) + \boldsymbol{\beta_1}(C)) - (\boldsymbol{q}'(C) + \boldsymbol{\alpha_1}(C))] [(\boldsymbol{r}'(D) + \boldsymbol{\beta_1}(D)) - (\boldsymbol{q}'(B) + \boldsymbol{\alpha_1}(D))]$$

$$\begin{aligned}
&= \sum_{i=1}^{n_1} [(\boldsymbol{r}'(C)\boldsymbol{r}'(D) - \boldsymbol{q}'(C)\boldsymbol{q}'(D) - \boldsymbol{r}'(C)\boldsymbol{q}'(D) + \boldsymbol{q}'(C)\boldsymbol{r}'(D)] \\
&+ \sum_{i=1}^{n_1} [(\boldsymbol{\beta_1}(C)\boldsymbol{r}'(D) - \boldsymbol{\alpha_1}(C)\boldsymbol{r}'(D) - \boldsymbol{\beta_1}(C)\boldsymbol{q}'(D) + \boldsymbol{\alpha_1}(C)\boldsymbol{q}'(D)] \\
&+ \sum_{i=1}^{n_1} [(\boldsymbol{r}'(C)\boldsymbol{\beta_1}(D) - \boldsymbol{r}'(C)\boldsymbol{\alpha_1}(D) - \boldsymbol{q}'(C)\boldsymbol{\beta_1}(D) + \boldsymbol{q}'(C)\boldsymbol{\alpha_1}(D)] \\
&+ \sum_{i=1}^{n_1} [(\boldsymbol{\beta_1}(C)\boldsymbol{\beta_1}(D) - \boldsymbol{\alpha_1}(C)\boldsymbol{\beta_1}(D) - \boldsymbol{\beta_1}(C)\boldsymbol{\alpha_1}(D) + \boldsymbol{\alpha_1}(C)\boldsymbol{\alpha_1}(D)] \\
&= \sum_{i=1}^{n_1} C_m D_m + \sum_{i=1}^{n_1} \Delta_m^C D_m + \sum_{i=1}^{n_1} \Delta_m^D C_m + \sum_{i=1}^{n_1} \Delta_m^C \Delta_m^D \\
&= \sum_{i=1}^{n_1} C_m D_m + \Delta_m^C \sum_{i=1}^{n_1} D_m + \Delta_m^D \sum_{i=1}^{n_1} C_m + n_1 \Delta_m^C \Delta_m^D
\end{aligned}$$

Similarly, we can show that:

$$\text{Updated}(\sum_{i=1}^{n_2} E_m F_m) = \sum_{i=1}^{n_2} E_m F_m + \Delta_m^E \sum_{i=1}^{n_2} F_m + \Delta_m^F \sum_{i=1}^{n_2} E_m + n_2 \Delta_m^E \Delta_m^F$$

Adding the two updated statistics, the lemma follows.

Corollary 2.

$$\sum_{i=1}^{n=n_1+n_2} A_m^2 = \sum_{i=1}^{n=n_1+n_2} A_m A_m = \left[\sum_{i=1}^{n_1} C_m C_m + 2\Delta_m^C \sum_{i=1}^{n_1} C_m + n_1 \left(\Delta_m^C\right)^2\right] + \left[\sum_{i=1}^{n_2} E_m E_m + 2\Delta_m^E \sum_{i=1}^{n_2} E_m + n_2 \left(\Delta_m^E\right)^2\right]$$

Corollary 3.

$$\sum_{i=1}^{n=n_1+n_2} A_p B_p = \left[\sum_{i=1}^{n_1} C_p D_p + \Delta_p^C \sum_{i=1}^{n_1} D_p + \Delta_p^D \sum_{i=1}^{n_1} C_p + n_1 \Delta_p^C \Delta_p^D\right] + \left[\sum_{i=1}^{n_2} E_p F_p + \Delta_p^E \sum_{i=1}^{n_2} F_p + \Delta_p^F \sum_{i=1}^{n_2} E_p + n_2 \Delta_p^E \Delta_p^F\right]$$

Corollary 4.

$$\sum_{i=1}^{n=n_1+n_2} A_p^2 = \sum_{i=1}^{n=n_1+n_2} A_p A_p = \left[\sum_{i=1}^{n_1} C_p C_p + 2\Delta_p^C \sum_{i=1}^{n_1} C_p + n_1 \left(\Delta_p^C\right)^2\right] + \left[\sum_{i=1}^{n_2} E_p E_p + 2\Delta_p^E \sum_{i=1}^{n_2} E_p + n_2 \left(\Delta_p^E\right)^2\right]$$

Lemma 3.

$$\sum_{i=1}^{n=n_1+n_2} A_m B_p = \left[\sum_{i=1}^{n_1} C_m D_p + \Delta_m^C \sum_{i=1}^{n_1} D_p + \Delta_p^D \sum_{i=1}^{n_1} C_m + n_1 \Delta_m^C \Delta_p^D\right] + \left[\sum_{i=1}^{n_2} E_m F_p + \Delta_m^E \sum_{i=1}^{n_2} F_p + \Delta_p^F \sum_{i=1}^{n_2} E_m + n_2 \Delta_m^E \Delta_p^F\right]$$

where $\Delta_m^C = \boldsymbol{\beta_1}(C) - \boldsymbol{\alpha_1}(C)$, $\Delta_m^D = \boldsymbol{\beta_1}(D) - \boldsymbol{\alpha_1}(D)$, $\Delta_m^E = \boldsymbol{\beta_2}(E) - \boldsymbol{\alpha_2}(E)$, *and* $\Delta_m^F = \boldsymbol{\beta_2}(F) - \boldsymbol{\alpha_2}(F)$ $A = C = E \in \{x, y, z\}$ *and* $B = D = F \in \{x, y, z\}$

Proof.

$$\text{Updated}(\sum_{i=1}^{n_1} C_m D_p) = \sum_{i=1}^{n_1} [(\boldsymbol{r'}(C) + \boldsymbol{\beta_1}(C)) - (\boldsymbol{q'}(C) + \boldsymbol{\alpha_1}(C))] \\ [(\boldsymbol{r'}(D) + \boldsymbol{\beta_1}(D)) + (\boldsymbol{q'}(B) + \boldsymbol{\alpha_1}(D))]$$

$$
\begin{aligned}
&= \sum_{i=1}^{n_1} [(\boldsymbol{r}'(C)\boldsymbol{r}'(D) - \boldsymbol{q}'(C)\boldsymbol{q}'(D) + \boldsymbol{r}'(C)\boldsymbol{q}'(D) - \boldsymbol{q}'(C)\boldsymbol{r}'(D)] \\
&+ \sum_{i=1}^{n_1} [(\boldsymbol{\beta_1}(C)\boldsymbol{r}'(D) - \boldsymbol{\alpha_1}(C)\boldsymbol{r}'(D) + \boldsymbol{\beta_1}(C)\boldsymbol{q}'(D) - \boldsymbol{\alpha_1}(C)\boldsymbol{q}'(D)] \\
&+ \sum_{i=1}^{n_1} [(\boldsymbol{r}'(C)\boldsymbol{\beta_1}(D) - \boldsymbol{r}'(C)\boldsymbol{\alpha_1}(D) + \boldsymbol{q}'(C)\boldsymbol{\beta_1}(D) - \boldsymbol{q}'(C)\boldsymbol{\alpha_1}(D)] \\
&+ \sum_{i=1}^{n_1} [(\boldsymbol{\beta_1}(C)\boldsymbol{\beta_1}(D) - \boldsymbol{\alpha_1}(C)\boldsymbol{\beta_1}(D) + \boldsymbol{\beta_1}(C)\boldsymbol{\alpha_1}(D) - \boldsymbol{\alpha_1}(C)\boldsymbol{\alpha_1}(D)] \\
&= \sum_{i=1}^{n_1} C_m D_p + \sum_{i=1}^{n_1} \Delta_m^C D_p + \sum_{i=1}^{n_1} \Delta_m^D C_p + \sum_{i=1}^{n_1} \Delta_m^C \Delta_p^D \\
&= \sum_{i=1}^{n_1} C_m D_p + \Delta_p^C \sum_{i=1}^{n_1} D_m + \Delta_m^D \sum_{i=1}^{n_1} C_p + n_1 \Delta_m^C \Delta_p^D
\end{aligned}
$$

Similarly, we can show that:

$$
\text{Updated}(\sum_{i=1}^{n_2} E_m F_p) = \sum_{i=1}^{n_2} E_m F_p + \Delta_m^E \sum_{i=1}^{n_2} F_p + \Delta_p^F \sum_{i=1}^{n_2} E_m + n_2 \Delta_m^E \Delta_p^F
$$

Adding the two updated statistics, the lemma follows.

4.2 Deletion operation of vector sets using sufficient statistics

Let us consider the case where we want to find a superposition under a deletion operation. That is, let $\mathcal{Q} \leftrightarrow \mathcal{R}$ and $\mathcal{S} \leftrightarrow \mathcal{T}$ denote two pairs of vector sets that are in correspondence. Let $\mathcal{S} \subset \mathcal{Q}$ and $\mathcal{T} \subset \mathcal{R}$. Under this assumption, let us define $\mathcal{U} = \mathcal{Q} - \mathcal{S}$ and $\mathcal{V} = \mathcal{R} - \mathcal{T}$.

Using the same notations as in the previous section, it is straightforward to see that the sufficient statistics $\boldsymbol{\Psi}$ of the superposition of $\mathcal{U}$ with $\mathcal{V}$ can be derived from the sufficient statistics $\boldsymbol{\Psi_1}$ (of $\mathcal{Q} \leftrightarrow \mathcal{R}$) and $\boldsymbol{\Psi_2}$ (of $\mathcal{S} \leftrightarrow \mathcal{T}$). The update rules defining the deletion operation are similar to the ones described above, so we leave these rules to the reader as an exercise.

5 Computing the r.m.s.d. from updated sufficient statistics

It is easy to see that Kearsley's 4×4 quaternion matrix $\mathbf{Q}$ given in Equation 1 can be constructed using the updated sufficient statistics $\boldsymbol{\Psi}$ derived from $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$. The matrix $\mathbf{Q}$ contains 10 distinct elements (given that $\mathcal{Q}$ is square symmetric) which can be computed in constant time.

In practice, $\mathbf{Q}$ is diagonalized using the Jacobi's iterative rotation approach, which with each rotation annihilates an off-diagonal element. This approach has a fast convergence, and requiring no additional optimization. However, in many cases the updated superposition shows only a marginal change from the previous one. For example, if we were to extend a current superposition by one pair of residues, the resultant new transformation will often, in practice, be very close to the previously computed one. This allows the diagonalisation to build on the previous solution.

Let $\mathbf{Q}$ denote the Kearsley's 4×4 matrix corresponding to the superposition of corresponding vector sets $\mathcal{U}$ and $\mathcal{V}$. From eigen decomposition theorem, we get $\mathbf{Q} = S\Lambda S^{-1}$, where S is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues. Also note that $\mathbf{Q}$ is positive semidefinite matrix with the property $\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T$. This implies that all the eigenvectors are orthogonal to each other. This further simplifies the decomposition to $\mathbf{Q} = S\Lambda S^T$. Also, since S is an orthogonal matrix, $\mathbf{Q} = S\Lambda S^T \implies \Lambda = S^T\mathbf{Q}S$.

Now, assume that the corresponding vector sets are augmented from $\mathcal{U}$ and $\mathcal{V}$ to $\mathcal{U}'$ and $\mathcal{V}'$, resulting in an updated Kearsley's matrix $\mathbf{Q}'$. We want to diagonalize this matrix into $S'\Lambda'S'^T$. Instead of starting the Jacobi's iterative process from scratch, we use the previously computed eigenvectors (before the vector sets were augmented), S, and compute $\tilde{\Lambda}$ as $S^T\mathcal{Q}'S$. Notice that if the augmentation does not include drastic changes, then $\tilde{\Lambda}$ is nearly diagonal (that is, $\tilde{\Lambda} \approx \Lambda'$), thus requiring very few iterations to fully diagonalize $\tilde{\Lambda}$. This provides a further optimization to the diagonalization step under update operations on vector sets.

6 Experiments

C++ programs were developed to compare the performance gain using sufficient statistics, when compared with the approach which recomputes the superposition *ab initio*.

8992 ASTRAL SCOP [26, 27] domains were as the source structures from which superposable fragments are randomly sampled. The general procedure of sampling is as follows. From the list of source structures, uniformly randomly choose a particular structure. Within this structure choose 2 random fragments of lengths l_1 and l_2, where the length is between 10 and 40 residues. These chosen fragments form the sets $\mathcal{Q}$ and $\mathcal{S}$. Yet another structure is again randomly chosen, and two fragments are sampled from it such that their lengths are strictly l_1 and l_2 respectively. These form the sets $\mathcal{R}$ and $\mathcal{T}$.

Assuming one-to-one correspondence between $\mathcal{Q} \leftrightarrow \mathcal{R}$ we compute the sufficient statistics $\mathbf{\Psi}_1$ of their orthogonal superposition. Similarly the sufficient statistics $\mathbf{\Psi}_2$ is computed for the orthogonal superposition between $\mathcal{R} \leftrightarrow \mathcal{T}$. Define $\mathcal{U} = \mathcal{Q} + \mathcal{S}$ and $\mathcal{V} = \mathcal{R} + \mathcal{T}$.

Iterating this process over 100 million such random samples, we compute:

1. The time it takes to superpose $\mathcal{U} \leftrightarrow \mathcal{V}$ and compute r.m.s.d from scratch.

2. The time it takes to superpose the same and compute r.m.s.d using the sufficient statistics $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$
3. The difference between the two r.m.s.d values. (This is performed to ascertain the numerical stability involved in computing the r.m.s.d. values from sufficient statistics.)

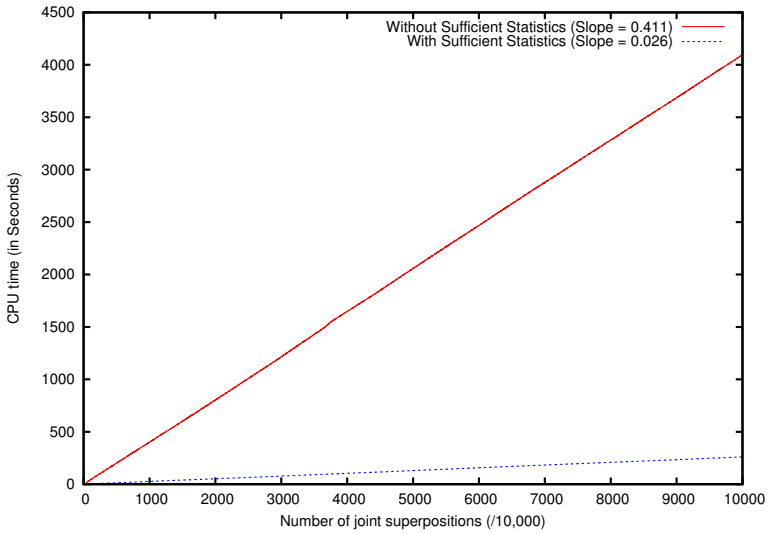

Fig. 1. The CPU times (in seconds) performing joint superpositions from scratch (Red line) compared against the same using sufficient statistics (Blue line) over 100 million random fragment data sets derived from ASTRAL SCOP domains. The X-axis reports the number of joint superpositions divided by 10,000.

Figure 1 compares the run times for the data set discussed above. Without sufficient statistics the run times takes 1.15 hours to conduct 100 million joint superpositions, while the same task is be achieved in 261 seconds ($\approx$4 minutes) using sufficient statistics. This shows a drastic improvement in the run time.

These empirical runtime results demonstrate what we have shown in Section 4.1, that the updates using sufficient statistics can be performed in constant time. If $|J|$ is the number of joint superpositions and n is the (average) number of points being superposed, then the first method grows as $O(n|J|)$. Since $n \ll |J|$ we see a linear trend (with a steeper gradient accounting for the multiplier n in the complexity term). In comparison, the results with sufficient statistics grow simply as $O(|J|)$ with a small gradient, made possible due to constant time computation of r.m.s.d values (using sufficient statistics) in each iterations.

To assess the numerical stability of our approach, we computed the r.m.s.d. values using the two approaches. The mean and standard deviation of the *difference* between the two r.m.s.d values were then computed. Both the mean and

the standard deviation are zero Å up to double precision. This demonstrates the numerical stability of computing r.m.s.d. using sufficient statistics.

7 Conclusion

Optimal superpositions of vector sets provide the foundation to determine similarities and differences between spatial objects, especially for macromolecular structures. We derived a set of sufficient statistics for the orthogonal superposition problem minimizing the sum of squares error. These statistics provide a highly efficient method to operate (via addition and deletion of vectors) on the existing superpositions. Our results demonstrate a drastic improvement in the computational effort required to compute r.m.s.d. using sufficient statistics. These results are relevant to many analyses involving structural data. These include the plethora of algorithms to construct pairwise and multiple protein structural alignments by assembling fragment pairs. Source code (written in C++) to undertake superpositions of vector sets using sufficient statistics can be downloaded from `http://www.csse.monash.edu.au/~karun/suffStatSuperpose.html`

References

1. Lesk, A.M.: Introduction to protein architecture: the structural biology of proteins. Oxford University Press (2001)
2. Eidhammer, I., Jonassen, I., Taylor, W.R.: Protein Bioinformatics: An algorithmic approach to sequence and structure analysis. J. Wiley & Sons (2004)
3. Lesk, A.M.: The unreasonable effectiveness of mathematics in molecular biology. The Mathematical Intelligencer **22**(2) (2000) 28–37
4. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography **32**(5) (1976) 922–923
5. Kabsch, W.: A discussion of the solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography **34**(5) (1978) 827–828
6. McLachlan, A.D.: Rapid comparison of protein structures. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography **38**(6) (1982) 871–873
7. KenKnight, C.: Comparison of methods of matching protein structures. Acta Crystallographica Section A: Foundations of Crystallography **40**(6) (1984) 708–712
8. Mackay, A.L.: Quaternion transformation of molecular orientation. Acta Crystallographica Section A: Foundations of Crystallography **40**(2) (1984) 165–166
9. Lesk, A.: A toolkit for computational molecular biology. II. on the optimal superposition of two sets of coordinates. Acta Crystallographica Section A: Foundations of Crystallography **42**(2) (1986) 110–113
10. Diamond, R.: A note on the rotational superposition problem. Acta Crystallographica Section A: Foundations of Crystallography **44**(2) (1988) 211–216

11. Kearsley, S.K.: On the orthogonal transformation used for structural comparisons. Acta Crystallographica Section A: Foundations of Crystallography **45**(2) (1989) 208–210
12. Cohen, G.: ALIGN: a program to superimpose protein coordinates, accounting for insertions and deletions. Journal of applied crystallography **30**(6) (1997) 1160–1161
13. Coutsias, E.A., Seok, C., Dill, K.A.: Using quaternions to calculate RMSD. Journal of Computational Chemistry **25**(15) (2004) 1849–1857
14. Koehl, P.: Protein structure similarities. Current opinion in structural biology **11**(3) (2001) 348–353
15. Hamilton, W.R., Hamilton, W.E.: Elements of quaternions. Longmans, Green, & Company (1866)
16. Shindyalov, I.N., Bourne, P.E.: Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. Protein engineering **11**(9) (1998) 739–747
17. Ye, Y., Godzik, A.: Flexible structure alignment by chaining aligned fragment pairs allowing twists. Bioinformatics **19**(suppl 2) (2003) ii246–ii255
18. Shatsky, M., Nussinov, R., Wolfson, H.J.: A method for simultaneous alignment of multiple protein structures. Proteins: Structure, Function, and Bioinformatics **56**(1) (2004) 143–156
19. Konagurthu, A.S., Whisstock, J.C., Stuckey, P.J., Lesk, A.M.: MUSTANG: a multiple structural alignment algorithm. Proteins: Structure, Function, and Bioinformatics **64**(3) (2006) 559–574
20. Shatsky, M., Nussinov, R., Wolfson, H.J.: Flexible protein alignment and hinge detection. Proteins: Structure, Function, and Bioinformatics **48**(2) (2002) 242–256
21. Vriend, G., Sander, C.: Detection of common three-dimensional substructures in proteins. Proteins: Structure, Function, and Bioinformatics **11**(1) (1991) 52–58
22. Lackner, P., Koppensteiner, W.A., Sippl, M.J., Domingues, F.S.: Prosup: a refined tool for protein structure alignment. Protein Engineering **13**(11) (2000) 745–752
23. Kolodny, R., Koehl, P., Levitt, M.: Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. Journal of molecular biology **346**(4) (2005) 1173–1188
24. Hogg, R.V., Craig, A.: Introduction to mathematical statistics. Prentice Hall (1994)
25. Jacobi, C.G.J.: Über ein leichtes Verfahren, die in der Theorie der Säkularstörungen vorkommenden Gleichungen numerisch aufzulösen. Journal für die Reine und Angewandte Mathematik **30** (1846) 51–95
26. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: SCOP: a structural classification of proteins database for the investigation of sequences and structures. Journal of molecular biology **247**(4) (1995) 536–540
27. Chandonia, J.M., Hon, G., Walker, N.S., Conte, L.L., Koehl, P., Levitt, M., Brenner, S.E.: The ASTRAL compendium in 2004. Nucleic acids research **32**(suppl 1) (2004) D189–D192