Differentiable Structure of the Kabsch Algorithm

Hyeongwoo Kim

February 04, 2025

1 Introduction to Kabsch Algorithm

Kabsch algorithm, or often referred to as Kabsch alignment, is a method for aligning two sets of points to minimize their RMSD (root mean squared deviation). Kabsch algorithm is a useful method to quantitatively compare configurations of two point sets in the cases where translation and rotation do not affect their mathematical structures. This is frequently encountered in chemistry and biochemistry to compare two conformations of the same molecule or biomolecule — protein is such an example. In dynamics simulations, the Kabsch algorithm can be adopted to align successive conformations with a small time gap, excluding rigid-body motions and ensuring that only internal motions remain. [1]

This algorithm first translates two point sets to ensure their centroids coincide with the origin of the coordinate system. Then, the optimal rotation matrix is calculated by computing a cross-covariance matrix of the two point set coordinates. The Kabsch algorithm is routinely solved by the singular value decomposition (SVD) if it is available for the computed covariance matrix.

1.1 Mathematical Description of Kabsch Algorithm

Let P and Q be two point sets in Euclidean space, each containing N points in $\mathbb{R}^3$. Here, we will find the transformation from Q to P , that is, the optimal translation and rotation operations are applied to Q to be aligned to P. The sets P and Q are represented as matrices whose columns correspond to coordinates of respective points:

$$P \in \mathbb{R}^{3\times N} \text{ and } Q \in \mathbb{R}^{3\times N}.$$

For the Kabsch algorithm to be applied, the two point sets are first translated so that their centroids coincide with the origin

$$\bar{P}_{ij} := P_{ij} - \frac{1}{N}\sum_{j=1}^{N} P_{ij}, \quad \bar{Q}_{ij} := Q_{ij} - \frac{1}{N}\sum_{j=1}^{N} Q_{ij},$$

for which one can readily check $\sum_{j=1}^{N} \bar{P}_{ij} = \sum_{j=1}^{N} \bar{Q}_{ij} = 0$. The Kabsch algorithm is a method to find the solution of the following minimization problem:

$$\begin{aligned} R^* &= \underset{R}{\operatorname{argmin}}\ \mathrm{RMSD}(\bar{P}, R\bar{Q}) \\ &= \underset{R}{\operatorname{argmin}}\ \sum_{i=1}^{N} |\bar{p}_i - R\bar{q}_i|^2, \end{aligned} \tag{1}$$

where R is a rotation matrix in $\mathbb{R}^3$, an orthogonal matrix that satisfies $R^\top R = RR^\top = I \in \mathbb{R}^{3\times 3}$, $\bar{p}_i$ and $\bar{q}_i$ are i-th column vectors of $\bar{P}$ and $\bar{Q}$ matrices, respectively.

1.2 Solving Kabsch Algorithm with the SVD

The minimization problem (1) is routinely solved by SVD, if it can be applied. The cross-covariance matrix $H \in \mathbb{R}^{3\times 3}$ between the two point sets is calculated as

$$H = \bar{P}\bar{Q}^\top \in \mathbb{R}^{3\times 3}, \quad H_{ij} = \bar{P}_{ik}\bar{Q}_{jk}, \tag{2}$$

where the Einstein summation convention was used. We can then compute the singular value decomposition of the covariance matrix

$$H = USV^\top, \tag{3}$$

where U and V are $\mathbb{R}^{3\times 3}$ orthogonal matrices and S is a $\mathbb{R}^{3\times 3}$ diagonal matrix. Then, we may need to correct for improper rotations (i.e., reflections) by flipping the sign of the last column of the V matrix. We can check if this is necessary by computing the determinant of $UV^\top$:

$$d = \text{sign}(\det(UV^\top)). \tag{4}$$

If $d = -1$, we need to flip the sign of the last column of V. Otherwise, we can leave it as is. Let $B = \text{diag}(1, 1, d)$ be a diagonal matrix with d on the last diagonal element. Finally, we can compute the optimal rotation matrix R^*:

$$R^* = UBV^\top. \tag{5}$$

If $d = 1$, the rotation matrix becomes simply $R^* = UV^\top$. The rotation matrix is applied to $\bar{Q}$ to obtain RMSD-minimizing alignment:

$$\bar{Q}^* = R^*\bar{Q}. \tag{6}$$

2 Differentiating Kabsch Algorithm

We can regard the Kabsch alignment as a function f that aligns a set of N points with a fixed reference point set $\bar{P}$;

$$f : \mathbb{R}^{3\times N} \times \mathbb{R}^{3\times N} \to \mathbb{R}^{3\times N},\ f(Q; \bar{P}) = R^*\bar{Q} = \bar{Q}^*. \tag{7}$$

Then we can define the Jacobian tensor J for the function f as

$$J = \frac{\partial f(Q; \bar{P})}{\partial Q} = \frac{\partial \bar{Q}^*}{\partial Q} \in \mathbb{R}^{3\times N\times 3\times N}, \quad J_{ijkl} = \frac{\partial \bar{Q}^*_{ij}}{\partial Q_{kl}}. \tag{8}$$

Plugging the eq. (5) into eq. (8) yields

$$\begin{aligned}
J_{ijkl} &= \frac{\partial (R^*_{im}\bar{Q}_{mj})}{\partial Q_{kl}} \\
&= \frac{\partial R^*_{im}}{\partial Q_{kl}}\bar{Q}_{mj} + R^*_{im}\frac{\partial \bar{Q}_{mj}}{\partial Q_{kl}} \\
&= \frac{\partial (U_{in}V_{mn})}{\partial Q_{kl}}\bar{Q}_{mj} + R^*_{im}\frac{\partial}{\partial Q_{kl}}\left(Q_{mj} - \frac{1}{N}\sum_{j}^{N} Q_{mj}\right) \\
&= \frac{\partial U_{in}}{\partial Q_{kl}}V_{mn}\bar{Q}_{mj} + U_{in}\frac{\partial V_{mn}}{\partial Q_{kl}}\bar{Q}_{mj} + R^*_{im}(\delta_{mk}\delta_{jl} - \frac{1}{N}\delta_{mk}) \\
&= \frac{\partial U_{in}}{\partial Q_{kl}}V_{mn}\bar{Q}_{mj} + U_{in}\frac{\partial V_{mn}}{\partial Q_{kl}}\bar{Q}_{mj} + U_{in}V_{kn}(\delta_{jl} - \frac{1}{N}),
\end{aligned} \tag{9}$$

where δ is a Kronecker delta and B in eq. (5) was assumed to be the identity matrix. We then need to compute two terms: $\partial U/\partial Q$ and $\partial V/\partial Q$, which are discussed in the following sections.

2.1 Differentiating the SVD

U and V matrices are related to Q by the SVD, so $\partial U/\partial Q$ and $\partial V/\partial Q$ are related to the differentiation of the SVD. The general differentiation of the SVD, beyond its application in the Kabsch algorithm, has been covered by a few prior works. Thus, only a summary of their findings is presented here. For details, readers are recommended to refer to James Townsend's work.[2]

Here, we assume that the SVD is available for a matrix $H \in \mathbb{R}^{m\times n}$ matrix of rank k so that $H = USV^\top$, where $U \in \mathbb{R}^{m\times k}$, $S \in \mathbb{R}^{k\times k}$, $V \in \mathbb{R}^{n\times k}$. Let's define a skew-symmetric matrix F as follows:

$$F_{ij} = \begin{cases} \frac{1}{s_j^2 - s_i^2} & i \neq j \\ 0 & i = j \end{cases}, \quad F_{ij} = (1-\delta_{ij})\frac{1}{s_j^2 - s_i^2}, \quad \text{(no summation over } i, j) \tag{10}$$

where s_i is an i-th diagonal element of S, i.e. $s_i := S_{ii}$. Note that Einstein summation convention was not used in Eq.(10). Then, differentials dU, dS, dV can be computed in terms of dH, U, S, and V:

$$dU = U(F \circ [U^\top dHVS + SV^\top dH^\top U]) + (I_m - UU^\top)dHVS^{-1} \tag{11}$$

$$dS = I_k \circ [U^\top dHV] \tag{12}$$

$$dV = V(F \circ [SU^\top dHV + V^\top dH^\top US]) + (I_m - VV^\top)dH^\top US^{-1}, \tag{13}$$

where I_m and I_k are identity matrices of rank m and k, and $\circ$ denotes the Hadamard product.

2.2 Application on Differentiating Kabsch Algorithm

Let's come back to differentiating the Kabsch algorithm. Note that $U, S, V \in \mathbb{R}^{3\times 3}$. We can compute $\partial U/\partial Q$ and $\partial V/\partial Q$ by utilizing eq. (11) and eq. (13).

$$\begin{aligned} \frac{\partial U}{\partial Q_{kl}} &= U(F \circ [U^\top \frac{\partial H}{\partial Q_{kl}} VS + SV^\top \frac{\partial H}{\partial Q_{kl}}^\top U]) + (\cancel{I - UU^\top})\frac{\partial H}{\partial Q_{kl}} VS^{-1} \\ &= U(F \circ [U^\top \frac{\partial H}{\partial Q_{kl}} VS + SV^\top \frac{\partial H}{\partial Q_{kl}}^\top U]) \end{aligned} \tag{14}$$

since U is an orthogonal matrix. For $\partial V/\partial Q$, similarly,

$$\begin{aligned} \frac{\partial V}{\partial Q_{kl}} &= V(F \circ [SU^\top \frac{\partial H}{\partial Q_{kl}} V + V^\top \frac{\partial H^\top}{\partial Q_{kl}} US]) + (\cancel{I - VV^\top})\frac{\partial H^\top}{\partial Q_{kl}} US^{-1} \\ &= V(F \circ [SU^\top \frac{\partial H}{\partial Q_{kl}} V + V^\top \frac{\partial H^\top}{\partial Q_{kl}} US]) \end{aligned} \tag{15}$$

since V is an orthogonal matrix. In eq. (14) and eq. (15), there are repeating and symmetric terms. For simplicity, let's define a tensor T as

$$T := U^\top \frac{\partial H}{\partial Q} V, \quad T_{ijkl} = \left(U^\top \frac{\partial H}{\partial Q_{kl}} V\right)_{ij} = U_{mi}\frac{\partial H_{mn}}{\partial Q_{kl}} V_{nj}. \tag{16}$$

By plugging eq. (2) into eq. (16), T can be computed as

$$\begin{aligned} T_{ijkl} &= U_{mi} \frac{\partial H_{mn}}{\partial Q_{kl}} V_{nj} \\ &= U_{mi} \frac{\partial}{\partial Q_{kl}} \left(\bar{P}_{mo} \bar{Q}_{no} \right) V_{nj} \\ &= U_{mi} \bar{P}_{mo} \delta_{kn} (\delta_{lo} - 1/N) V_{nj} \\ &= U_{mi} \bar{P}_{mo} (\delta_{lo} - 1/N) V_{kj}. \end{aligned} \tag{17}$$

Now, $\partial U/\partial Q$ can be computed as

$$\frac{\partial U}{\partial Q} = U(F \circ [TS + ST^\top]) = U(F \circ [TS + (TS)^\top]) \tag{18}$$

where the transpose operations in eq. (18) are applied to the first and second indices, and the second equality holds since S is a symmetric matrix. In element-wise representation,

$$\frac{\partial U_{ij}}{\partial Q_{kl}} = U_{im} F_{mj} [T_{mnkl} S_{nj} + S_{mn} T_{jnkl}], \tag{19}$$

where m and n are dummy indices that are summed over. $\partial V/\partial Q$ can be computed in the same way,

$$\frac{\partial V}{\partial Q} = V(F \circ [ST + T^\top S]) = V(F \circ [ST + (ST)^\top]) \tag{20}$$

$$\frac{\partial V_{ij}}{\partial Q_{kl}} = V_{im} F_{mj} [S_{mn} T_{njkl} + T_{nmkl} S_{nj}], \tag{21}$$

where m and n are dummy indices that are summed over. By utilizing eq. (19) and eq. (21), the Jacobian tensor can be computed as eq. (9).

3 Hessian of Kabsch Algorithm

With the same notion of the Kabsch alignment as a function, we can define the Hessian tensor, $\mathcal{H}$,

$$\mathcal{H} = \frac{\partial^2 f(Q; \bar{P})}{\partial Q \partial Q} = \frac{\partial J}{\partial Q} \in \mathbb{R}^{3\times N\times 3\times N\times 3\times N}, \quad \mathcal{H}_{ijklmn} = \frac{\partial J_{ijkl}}{\partial Q_{mn}}. \tag{22}$$

By utilizing eq. (9) for J_{ijkl},

$$\begin{aligned} \mathcal{H}_{ijklop} &= \frac{\partial}{\partial Q_{op}} \left(\frac{\partial U_{in}}{\partial Q_{kl}} V_{mn} \bar{Q}_{mj} + U_{in} \frac{\partial V_{mn}}{\partial Q_{kl}} \bar{Q}_{mj} + U_{in} V_{kn} (\delta_{jl} - \frac{1}{N}) \right) \\ &= \frac{\partial}{\partial Q_{op}} \left(\frac{\partial U_{in}}{\partial Q_{kl}} V_{mn} \bar{Q}_{mj} \right) + \frac{\partial}{\partial Q_{op}} \left(U_{in} \frac{\partial V_{mn}}{\partial Q_{kl}} \bar{Q}_{mj} \right) + \frac{\partial}{\partial Q_{op}} \left(U_{in} V_{kn} (\delta_{jl} - \frac{1}{N}) \right) \\ &= \left(\frac{\partial^2 U_{in}}{\partial Q_{op} \partial Q_{kl}} \right) V_{mn} \bar{Q}_{mj} + \frac{\partial U_{in}}{\partial Q_{kl}} \left(\frac{\partial V_{mn}}{\partial Q_{op}} \right) \bar{Q}_{mj} + \frac{\partial U_{in}}{\partial Q_{kl}} V_{mn} \left(\frac{\partial \bar{Q}_{mj}}{\partial Q_{op}} \right) \\ &\quad + \left(\frac{\partial U_{in}}{\partial Q_{op}} \right) \frac{\partial V_{mn}}{\partial Q_{kl}} \bar{Q}_{mj} + U_{in} \left(\frac{\partial^2 V_{mn}}{\partial Q_{op} \partial Q_{kl}} \right) \bar{Q}_{mj} + U_{in} \frac{\partial V_{mn}}{\partial Q_{kl}} \left(\frac{\partial \bar{Q}_{mj}}{\partial Q_{op}} \right) \\ &\quad + \left(\frac{\partial U_{in}}{\partial Q_{op}} \right) V_{kn} (\delta_{jl} - \frac{1}{N}) + U_{in} \left(\frac{\partial V_{kn}}{\partial Q_{op}} \right) (\delta_{jl} - \frac{1}{N}). \end{aligned} \tag{23}$$

In eq. (23), most of the partial derivatives were computed in the previous section, except for the Hessian tensors of U and V; $\frac{\partial^2 U}{\partial Q \partial Q}$ and $\frac{\partial^2 V}{\partial Q \partial Q}$. These can be computed by utilizing eq. (19) and eq. (21),

$$
\begin{aligned}
\frac{\partial}{\partial Q_{op}}\left(\frac{\partial U_{ij}}{\partial Q_{kl}}\right) &= \frac{\partial}{\partial Q_{op}}\left(U_{im}F_{mn}[T_{mnkl}S_{nj}+S_{mn}T_{jnkl}]\right) \\
&= \frac{\partial U_{im}}{\partial Q_{op}}\left(F_{mn}[T_{mnkl}S_{nj}+S_{mn}T_{jnkl}]\right) \\
&\quad + U_{im}\frac{\partial}{\partial Q_{op}}\left(F_{mn}[T_{mnkl}S_{nj}+S_{mn}T_{jnkl}]\right), \qquad (24) \\
\text{(second term)} &= U_{im}\left(\frac{\partial F_{mn}}{\partial Q_{op}}\right)[T_{mnkl}S_{nj}+S_{mn}T_{jnkl}] \\
&\quad + U_{im}F_{mn}\left(\frac{\partial}{\partial Q_{op}}[T_{mnkl}S_{nj}+S_{mn}T_{jnkl}]\right) \\
&= U_{im}\left(\frac{\partial F_{mn}}{\partial Q_{op}}\right)[T_{mnkl}S_{nj}+S_{mn}T_{jnkl}] \\
&\quad + U_{im}F_{mn}\left(\frac{\partial T_{mnkl}}{\partial Q_{op}}S_{nj}+T_{mnkl}\frac{\partial S_{nj}}{\partial Q_{op}}+\frac{\partial S_{mn}}{\partial Q_{op}}T_{jnkl}+S_{mn}\frac{\partial T_{jnkl}}{\partial Q_{op}}\right). \qquad (25)
\end{aligned}
$$

In eq. (25), the following three terms need to be computed: $\frac{\partial S}{\partial Q}$, $\frac{\partial F}{\partial Q}$ and $\frac{\partial T}{\partial Q}$.

For $\frac{\partial S}{\partial Q}$, by utilizing eq. (12),

$$
\begin{aligned}
\frac{\partial S_{ij}}{\partial Q_{op}} &= \left(I \circ \left[U^{\top}\left(\frac{\partial H}{\partial Q_{op}}\right)V\right]\right)_{ij} \\
&= I_{ij}\left(U^{\top}\left(\frac{\partial H}{\partial Q_{op}}\right)V\right)_{ij} \quad \text{(no summation over } i,j\text{)} \\
&= I_{ij}T_{ijop} \quad \text{(no summation over } i,j\text{)} \\
&= \begin{cases} T_{ijop} & i=j \\ 0 & i\neq j \end{cases}, \qquad (26)
\end{aligned}
$$

which has been already calculated. For $\frac{\partial F}{\partial Q}$, by utilizing eq. (10) and eq. (26),

$$
\begin{aligned}
\frac{\partial F_{ij}}{\partial Q_{op}} &= \frac{\partial}{\partial Q_{op}}\left(\frac{1}{s_j^2-s_i^2}\right) \\
&= \frac{2}{(s_j^2-s_i^2)^2}\left(s_i\frac{\partial s_i}{\partial Q_{op}}-s_j\frac{\partial s_j}{\partial Q_{op}}\right) \\
&= 2(F_{ij})^2\left(S_{ii}\frac{\partial S_{ii}}{\partial Q_{op}}S_{jj}\frac{\partial S_{jj}}{\partial Q_{op}}\right) \\
&= 2(F_{ij})^2\left(S_{ii}T_{iiop}-S_{jj}T_{jjop}\right), \qquad (27)
\end{aligned}
$$

where the Einstein summation convention was not used for indices i and j. For $\frac{\partial T}{\partial Q}$, by utilizing eq. (17),

$$\begin{aligned}
\frac{\partial T_{ijkl}}{\partial Q_{op}} &= \frac{\partial}{\partial Q_{op}} \left(U_{mi}\bar{P}_{mo}(\delta_{lo} - 1/N)V_{kj}\right) \\
&= \left(\frac{\partial U_{mi}}{\partial Q_{op}}\right) \bar{P}_{mo}(\delta_{lo} - 1/N)V_{kj} \; + \; U_{mi}\bar{P}_{mo}(\delta_{lo} - 1/N) \left(\frac{\partial V_{kj}}{\partial Q_{op}}\right) \\
&= \left[\left(\frac{\partial U_{mi}}{\partial Q_{op}}\right) V_{kj} \; + \; U_{mi} \left(\frac{\partial V_{kj}}{\partial Q_{op}}\right)\right] \bar{P}_{mo}(\delta_{lo} - 1/N).
\end{aligned} \tag{28}$$

Now the Hessian tensor of U can be computed as eq. (24). Similarly, the Hessian tensor of V can be computed as

$$\begin{aligned}
\frac{\partial}{\partial Q_{op}} \left(\frac{\partial V_{ij}}{\partial Q_{kl}}\right) &= \frac{\partial}{\partial Q_{op}} \left(V_{im}F_{mn}[S_{mn}T_{njkl} + T_{nmkl}S_{nj}]\right) \\
&= \frac{\partial V_{im}}{\partial Q_{op}} \left(F_{mn}[S_{mn}T_{njkl} + T_{nmkl}S_{nj}]\right) \\
&\quad + V_{im}\frac{\partial}{\partial Q_{op}} \left(F_{mn}[S_{mn}T_{njkl} + T_{nmkl}S_{nj}]\right),
\end{aligned} \tag{29}$$

$$\begin{aligned}
\text{(second term)} &= V_{im} \left(\frac{\partial F_{mn}}{\partial Q_{op}}\right) [S_{mn}T_{njkl} + T_{nmkl}S_{nj}] \\
&\quad + V_{im}F_{mn} \left(\frac{\partial}{\partial Q_{op}}[S_{mn}T_{njkl} + T_{nmkl}S_{nj}]\right) \\
&= V_{im} \left(\frac{\partial F_{mn}}{\partial Q_{op}}\right) [S_{mn}T_{njkl} + T_{nmkl}S_{nj}] \\
&\quad + V_{im}F_{mn} \left(\frac{\partial S_{mn}}{\partial Q_{op}}T_{njkl} + S_{mn}\frac{\partial T_{njkl}}{\partial Q_{op}} + \frac{\partial T_{nmkl}}{\partial Q_{op}}S_{nj} + T_{nmkl}\frac{\partial S_{nj}}{\partial Q_{op}}\right).
\end{aligned} \tag{30}$$

Finally, eq. (24) and eq. (29) are plugged into eq. (23) to complete the calculation of the Hessian tensor of the Kabsch algorithm, $\mathcal{H}$. All terms that are necessary to calculate the Hessian tensor come from the differentiation of the SVD. Hence, readers are kindly recommended to read James Townsend's work[2].

References

[1] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

[2] James Townsend. Differentiating the singular value decomposition. *Technical Report*, 2016.