

Mathematical Framework for Molecular Alignment in SeamStress

SeamStress Documentation

January 22, 2026

Abstract

This document provides a comprehensive mathematical description of the alignment algorithms used in SeamStress for molecular geometry alignment and analysis. We describe the Kabsch algorithm, weighted alignment schemes, permutation search strategies, and the two-stage alignment process used for optimal molecular superposition.

Contents

1	Introduction	2
2	The Kabsch Algorithm	2
2.1	Standard Kabsch Algorithm	2
2.1.1	Problem Statement	2
2.1.2	Algorithm Steps	2
2.2	Weighted Kabsch Algorithm	3
2.2.1	Weight Definition	3
2.2.2	Weighted Algorithm	3
3	RMSD Calculation	4
3.1	Definition	4
3.2	Implementation	4
4	Permutation Search	4
4.1	Problem Statement	4
4.2	Brute Force Search Algorithm	4
4.2.1	Factorization by Atom Type	5
5	Fragment-Based Permutation Optimization	5
5.1	Applicability Condition	5
5.2	Fragment Definition	5
5.3	Complexity Reduction	6
5.4	Fragment Permutation Algorithm	6
6	Two-Stage Alignment Process	6
6.1	Rationale	6
6.2	Mathematical Formulation	7
6.3	Algorithm	7

7	Complete Alignment Workflows	7
7.1	Mode 1: Multi-Family Alignment	7
7.1.1	Workflow	7
7.2	Mode 2: Align-All-to-Centroid	8
7.2.1	Workflow	8
7.3	Visualization in Dimensionality Reduction	8
8	Computational Complexity	8
8.1	Kabsch Algorithm	8
8.2	Permutation Search	8
8.3	Complete Workflow	9
9	Numerical Stability	9
9.1	Weight Normalization	9
9.2	SVD Stability	9
9.3	Reflection Detection	9
10	Implementation Notes	9
10.1	Heavy Atom Factor Selection	9
10.2	RMSD Warning Thresholds	10
11	References	10

1 Introduction

SeamStress aligns molecular geometries using the Kabsch algorithm combined with optimal atom permutation search. The workflow involves:

1. **Connectivity grouping:** Molecules are grouped by connectivity (SMILES hash)
2. **Permutation search:** Finding optimal atom correspondence between molecules
3. **Kabsch alignment:** Optimal rotation and translation for superposition
4. **Heavy atom weighting:** Optional prioritization of heavy atoms in alignment
5. **RMSD calculation:** Quantifying structural similarity

2 The Kabsch Algorithm

2.1 Standard Kabsch Algorithm

The Kabsch algorithm finds the optimal rotation matrix \mathbf{R} and translation vector \mathbf{t} to align two sets of points.

2.1.1 Problem Statement

Given two sets of N points:

- Reference coordinates: $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\} \in \mathbb{R}^{N \times 3}$
- Target coordinates: $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N\} \in \mathbb{R}^{N \times 3}$

Find rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$ that minimize:

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_i - (\mathbf{R}\mathbf{q}_i + \mathbf{t})\|^2} \quad (1)$$

2.1.2 Algorithm Steps

Step 1: Compute centroids

$$\bar{\mathbf{p}} = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i, \quad \bar{\mathbf{q}} = \frac{1}{N} \sum_{i=1}^N \mathbf{q}_i \quad (2)$$

Step 2: Center the coordinates

$$\mathbf{P}' = \mathbf{P} - \bar{\mathbf{p}}, \quad \mathbf{Q}' = \mathbf{Q} - \bar{\mathbf{q}} \quad (3)$$

Step 3: Compute the covariance matrix

$$\mathbf{H} = \mathbf{P}'^T \mathbf{Q}' \quad (4)$$

Step 4: Singular Value Decomposition (SVD)

$$\mathbf{H} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad (5)$$

Step 5: Compute optimal rotation

$$\mathbf{R} = \mathbf{V} \mathbf{U}^T \quad (6)$$

If $\det(\mathbf{R}) < 0$ (reflection), correct by flipping the sign of the last column of \mathbf{V} :

$$\mathbf{V}[:, -1] \leftarrow -\mathbf{V}[:, -1], \quad \mathbf{R} = \mathbf{V}\mathbf{U}^T \quad (7)$$

Step 6: Apply transformation

The aligned coordinates are:

$$\mathbf{Q}_{\text{aligned}} = (\mathbf{Q} - \bar{\mathbf{q}})\mathbf{R} + \bar{\mathbf{p}} \quad (8)$$

2.2 Weighted Kabsch Algorithm

For molecular alignment, different atoms should contribute differently based on their atomic mass or type.

2.2.1 Weight Definition

Define weights w_i for each atom i . In SeamStress, three weight schemes are available:

1. **Mass weighting** (default): $w_i = m_i \cdot f_i$ where

$$m_i = \text{atomic mass}, \quad f_i = \begin{cases} h & \text{if atom } i \text{ is heavy} \\ 1 & \text{if atom } i \text{ is hydrogen} \end{cases} \quad (9)$$

Here h is the `heavy_atom_factor` (default: $h = 1.0$).

2. **Uniform weighting**: $w_i = 1$ for all atoms

3. **Heavy-only weighting**: $w_i = \begin{cases} 1 & \text{if atom } i \text{ is not H} \\ 0 & \text{if atom } i \text{ is H} \end{cases}$

2.2.2 Weighted Algorithm

Step 1: Normalize weights

$$w'_i = \frac{w_i}{\sum_{j=1}^N w_j} \quad (10)$$

Step 2: Weighted centroids

$$\bar{\mathbf{p}} = \sum_{i=1}^N w'_i \mathbf{p}_i, \quad \bar{\mathbf{q}} = \sum_{i=1}^N w'_i \mathbf{q}_i \quad (11)$$

Step 3: Center coordinates

$$\mathbf{P}' = \mathbf{P} - \bar{\mathbf{p}}, \quad \mathbf{Q}' = \mathbf{Q} - \bar{\mathbf{q}} \quad (12)$$

Step 4: Weighted covariance matrix

Let $\mathbf{W} = \text{diag}(w'_1, w'_2, \dots, w'_N)$. The covariance matrix is:

$$\mathbf{H} = \mathbf{P}'^T \mathbf{W} \mathbf{Q}' \quad (13)$$

This can be computed efficiently as:

$$\mathbf{H} = (\sqrt{\mathbf{W}} \mathbf{P}')^T (\sqrt{\mathbf{W}} \mathbf{Q}') \quad (14)$$

where $\sqrt{\mathbf{W}} = \text{diag}(\sqrt{w'_1}, \sqrt{w'_2}, \dots, \sqrt{w'_N})$.

Steps 5-6: Proceed with SVD and rotation computation as in standard Kabsch algorithm.

3 RMSD Calculation

The Root Mean Square Deviation (RMSD) quantifies the structural difference after optimal alignment.

3.1 Definition

After aligning \mathbf{Q} to \mathbf{P} , the RMSD is:

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_i - \mathbf{q}_i^{\text{aligned}}\|^2} \quad (15)$$

Expanding the Euclidean norm:

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N [(p_{i,x} - q_{i,x}^{\text{aligned}})^2 + (p_{i,y} - q_{i,y}^{\text{aligned}})^2 + (p_{i,z} - q_{i,z}^{\text{aligned}})^2]} \quad (16)$$

3.2 Implementation

In matrix form:

$$\mathbf{D} = \mathbf{P} - \mathbf{Q}_{\text{aligned}} \quad (17)$$

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^3 D_{ij}^2} = \sqrt{\frac{1}{N} \|\mathbf{D}\|_F^2} \quad (18)$$

where $\|\cdot\|_F$ is the Frobenius norm.

4 Permutation Search

For symmetric molecules, finding the optimal atom correspondence is crucial.

4.1 Problem Statement

Given molecules with identical connectivity, find permutation $\pi : \{1, \dots, N\} \rightarrow \{1, \dots, N\}$ that minimizes:

$$\text{RMSD}_\pi = \text{RMSD}(\mathbf{P}, \mathbf{Q}_\pi) \quad (19)$$

where \mathbf{Q}_π applies permutation π to rows of \mathbf{Q} :

$$\mathbf{Q}_\pi = \begin{bmatrix} \mathbf{q}_{\pi(1)} \\ \mathbf{q}_{\pi(2)} \\ \vdots \\ \mathbf{q}_{\pi(N)} \end{bmatrix} \quad (20)$$

4.2 Brute Force Search Algorithm

SeamStress uses a factored brute force search by atom type.

4.2.1 Factorization by Atom Type

Separate heavy atoms (C, N, O, etc.) from hydrogens:

- Heavy atom indices: $\mathcal{H} = \{i : \text{atom}_i \neq \text{H}\}$, $|\mathcal{H}| = n_h$
- Hydrogen indices: $\mathcal{I} = \{i : \text{atom}_i = \text{H}\}$, $|\mathcal{I}| = n_H$

Total permutations to test: $n_h! \times n_H!$

For ethylene (C_2H_4): $2! \times 4! = 2 \times 24 = 48$ permutations

Algorithm 1 Standard Permutation Search

```

1:  $\text{RMSD}_{\min} \leftarrow \infty$ 
2:  $\pi_{\text{best}} \leftarrow \text{identity}$ 
3: for each heavy atom permutation  $\pi_h \in S_{n_h}$  do
4:   for each hydrogen permutation  $\pi_H \in S_{n_H}$  do
5:     Combine:  $\pi \leftarrow \pi_h \cup \pi_H$ 
6:      $\mathbf{Q}' \leftarrow \mathbf{Q}_\pi$  ▷ Apply permutation
7:      $\mathbf{Q}_{\text{aligned}} \leftarrow \text{KabschAlign}(\mathbf{P}, \mathbf{Q}')$ 
8:      $r \leftarrow \text{RMSD}(\mathbf{P}, \mathbf{Q}_{\text{aligned}})$ 
9:     if  $r < \text{RMSD}_{\min}$  then
10:        $\text{RMSD}_{\min} \leftarrow r$ 
11:        $\pi_{\text{best}} \leftarrow \pi$ 
12:     end if
13:   end for
14: end for
15: return  $(\pi_{\text{best}}, \text{RMSD}_{\min})$ 

```

5 Fragment-Based Permutation Optimization

For molecules where each heavy atom has exactly one bonded hydrogen (e.g., benzene), we can treat heavy atom-hydrogen pairs as rigid fragments.

5.1 Applicability Condition

Fragment mode applies when:

$$\forall i \in \mathcal{H} : |\{j \in \mathcal{I} : \text{bonded}(i, j)\}| = 1 \quad (21)$$

This means each heavy atom has exactly one bonded hydrogen.

5.2 Fragment Definition

Define fragments F_k as heavy atom + bonded hydrogen pairs:

$$F_k = \{h_k, H_k\} \quad (22)$$

where h_k is a heavy atom and H_k is its bonded hydrogen.

5.3 Complexity Reduction

Standard mode (benzene with 6 carbons, 6 hydrogens):

$$\text{Permutations} = 6! \times 6! = 720 \times 720 = 518,400 \quad (23)$$

Fragment mode (benzene with 6 C-H fragments):

$$\text{Permutations} = 6! = 720 \quad (24)$$

Speedup: $\frac{518,400}{720} = 720\times$ faster!

5.4 Fragment Permutation Algorithm

Algorithm 2 Fragment-Based Permutation Search

```
1: Build fragment map:  $F = \{F_1, F_2, \dots, F_{n_h}\}$ 
2:  $\text{RMSD}_{\min} \leftarrow \infty$ 
3:  $\pi_{\text{best}} \leftarrow \text{identity}$ 
4: for each fragment permutation  $\sigma \in S_{n_h}$  do
5:   Initialize  $\pi \leftarrow [0, 0, \dots, 0]$  of length  $N$ 
6:   for  $k = 1$  to  $n_h$  do
7:      $F_{\text{ref}} \leftarrow F_k$  ▷ Reference fragment
8:      $F_{\text{tgt}} \leftarrow F_{\sigma(k)}$  ▷ Target fragment
9:     for atom  $a$  in  $F_{\text{ref}}$ , atom  $b$  in  $F_{\text{tgt}}$  do
10:       $\pi[a] \leftarrow b$  ▷ Map atoms in fragments
11:    end for
12:  end for
13:   $\mathbf{Q}' \leftarrow \mathbf{Q}_{\pi}$ 
14:   $\mathbf{Q}_{\text{aligned}} \leftarrow \text{KabschAlign}(\mathbf{P}, \mathbf{Q}')$ 
15:   $r \leftarrow \text{RMSD}(\mathbf{P}, \mathbf{Q}_{\text{aligned}})$ 
16:  if  $r < \text{RMSD}_{\min}$  then
17:     $\text{RMSD}_{\min} \leftarrow r$ 
18:     $\pi_{\text{best}} \leftarrow \pi$ 
19:  end if
20: end for
21: return  $(\pi_{\text{best}}, \text{RMSD}_{\min})$ 
```

6 Two-Stage Alignment Process

SeamStress uses a two-stage alignment to separate permutation search from heavy atom weighting.

6.1 Rationale

1. **Stage 1 (Permutation search):** Find optimal atom correspondence using mass-weighted alignment
2. **Stage 2 (Heavy atom refinement):** Re-align with increased heavy atom weighting using the permutation from Stage 1

This separates the combinatorial optimization (permutation) from the geometric optimization (alignment).

6.2 Mathematical Formulation

Stage 1: Find optimal permutation

Use mass-weighted Kabsch with $h = 1.0$:

$$\pi^* = \arg \min_{\pi \in S_N} \text{RMSD}_{\text{mass}}(\mathbf{P}, \mathbf{Q}_{\pi}) \quad (25)$$

Stage 2: Refine alignment

Apply heavy atom weighting with $h > 1.0$ (e.g., $h = 10.0$):

$$\mathbf{Q}_{\text{final}} = \text{WeightedKabsch}(\mathbf{P}, \mathbf{Q}_{\pi^*}, h) \quad (26)$$

6.3 Algorithm

Algorithm 3 Two-Stage Alignment

```
1: Input: Reference  $\mathbf{P}$ , Target  $\mathbf{Q}$ , heavy factor  $h$ 
2:
3: // Stage 1: Permutation Search
4:  $\pi^* \leftarrow \text{FindBestPermutation}(\mathbf{P}, \mathbf{Q}, h = 1.0)$ 
5:  $\mathbf{Q}' \leftarrow \mathbf{Q}_{\pi^*}$  ▷ Apply best permutation
6:
7: // Stage 2: Heavy Atom Refinement
8: if  $h > 1.0$  then
9:    $\mathbf{Q}_{\text{aligned}} \leftarrow \text{WeightedKabsch}(\mathbf{P}, \mathbf{Q}', h)$ 
10:   $\text{RMSD} \leftarrow \text{ComputeRMSD}(\mathbf{P}, \mathbf{Q}_{\text{aligned}})$ 
11: else
12:   $\mathbf{Q}_{\text{aligned}} \leftarrow \mathbf{Q}'$ 
13:   $\text{RMSD} \leftarrow \text{RMSD from Stage 1}$ 
14: end if
15:
16: return  $(\pi^*, \mathbf{Q}_{\text{aligned}}, \text{RMSD})$ 
```

7 Complete Alignment Workflows

7.1 Mode 1: Multi-Family Alignment

This mode groups molecules by connectivity and aligns each family independently.

7.1.1 Workflow

1. **Read geometries:** Load all XYZ files
2. **Connectivity analysis:** Compute SMILES hash for each molecule
3. **Family grouping:** Group molecules by SMILES
4. **Inter-family alignment:** Align family centroids to master reference

$$\mathbf{C}_i^{\text{aligned}} = \text{WeightedKabsch}(\mathbf{C}_{\text{master}}, \mathbf{C}_i, h_{\text{inter}}) \quad (27)$$

where h_{inter} is `inter_family_heavy_atom_factor`

5. **Intra-family alignment:** For each family i and molecule j :

$$\pi_{ij}^*, \mathbf{M}_{ij}^{\text{aligned}} = \text{TwoStageAlign}(\mathbf{C}_i^{\text{aligned}}, \mathbf{M}_{ij}, h_{\text{intra}}) \quad (28)$$

where h_{intra} is `intra_family_heavy_atom_factor`

7.2 Mode 2: Align-All-to-Centroid

This mode treats all molecules as one family and aligns to a single reference.

7.2.1 Workflow

1. **Load reference centroid:** Read specified centroid file \mathbf{C}_{ref}

2. **Align all spawning points:** For each molecule j :

$$\pi_j^*, \mathbf{M}_j^{\text{aligned}} = \text{TwoStageAlign}(\mathbf{C}_{\text{ref}}, \mathbf{M}_j, h_{\text{intra}}) \quad (29)$$

3. **Align all centroids:** For visualization, align all centroids to reference:

$$\mathbf{C}_k^{\text{aligned}} = \text{KabschAlign}(\mathbf{C}_{\text{ref}}, \mathbf{C}_k) \quad (30)$$

No permutation search for centroid alignment (identity permutation only)

4. **Save for analysis:**

- Aligned spawns \rightarrow `family_1/*.xyz`
- All aligned centroids \rightarrow `family_1/centroids.xyz` (multi-frame)

7.3 Visualization in Dimensionality Reduction

Mode 1: Each family centroid plotted as one star (\star)

$$\text{Stars} = \{\mathbf{C}_1^{\text{aligned}}, \mathbf{C}_2^{\text{aligned}}, \dots, \mathbf{C}_{n_{\text{families}}}^{\text{aligned}}\} \quad (31)$$

Mode 2: All aligned centroids plotted as stars (\star)

$$\text{Stars} = \{\mathbf{C}_1^{\text{aligned}}, \mathbf{C}_2^{\text{aligned}}, \dots, \mathbf{C}_{n_{\text{centroids}}}^{\text{aligned}}\} \quad (32)$$

In both modes, individual spawning points plotted as dots (\bullet).

8 Computational Complexity

8.1 Kabsch Algorithm

- Centroid computation: $O(N)$
- Covariance matrix: $O(N)$
- SVD of 3×3 matrix: $O(1)$
- Apply transformation: $O(N)$
- **Total:** $O(N)$ where N is number of atoms

8.2 Permutation Search

Standard mode:

$$\text{Complexity} = n_h! \times n_H! \times O(N) \quad (33)$$

Fragment mode:

$$\text{Complexity} = n_h! \times O(N) \quad (34)$$

For benzene ($n_h = 6$, $n_H = 6$, $N = 12$):

- Standard: $720 \times 720 \times O(12) \approx 6.2 \times 10^6$ operations
- Fragment: $720 \times O(12) \approx 8.6 \times 10^3$ operations
- Speedup: $720 \times$

8.3 Complete Workflow

For M molecules with F families:

Mode 1 (Multi-family):

$$O(F \cdot (\text{permutation search}) + M \cdot (\text{permutation search})) \quad (35)$$

Mode 2 (Align-all-to-centroid):

$$O(M \cdot (\text{permutation search}) + C \cdot O(N)) \quad (36)$$

where C is number of centroids (no permutation search for centroid alignment).

9 Numerical Stability

9.1 Weight Normalization

Weights are always normalized to sum to 1:

$$\sum_{i=1}^N w'_i = 1 \quad (37)$$

This prevents numerical overflow/underflow issues.

9.2 SVD Stability

The SVD is numerically stable and works correctly even for:

- Nearly degenerate configurations (collinear points)
- Large variations in coordinate magnitudes
- Ill-conditioned covariance matrices

9.3 Reflection Detection

Checking $\det(\mathbf{R}) < 0$ prevents reflections:

- If $\det(\mathbf{R}) = +1$: proper rotation
- If $\det(\mathbf{R}) = -1$: reflection detected, corrected by flipping last singular vector

10 Implementation Notes

10.1 Heavy Atom Factor Selection

Default ($h = 1.0$): Mass-weighted only

- C (mass 12) has $12\times$ influence of H (mass 1)
- Balanced for most molecules

Moderate ($h = 5.0$ to $h = 10.0$): Enhanced heavy atom weighting

- C has $60\times$ to $120\times$ influence of H
- Useful when hydrogens cause alignment issues

- Recommended for inter-family centroid alignment

Extreme ($h = 100.0$): Near heavy-only alignment

- C has $1200\times$ influence of H
- Essentially ignores hydrogens
- Use with caution

10.2 RMSD Warning Thresholds

Mode 2 (Align-all-to-centroid):

- Mean RMSD > 1.0 Å: Warning that molecules may have different connectivity
- Individual RMSD > 0.5 Å: Flagged as high deviation

These thresholds indicate potential issues with the alignment assumption.

11 References

1. Kabsch, W. (1976). "A solution for the best rotation to relate two sets of vectors." *Acta Crystallographica Section A* 32(5): 922-923.
2. Kabsch, W. (1978). "A discussion of the solution for the best rotation to relate two sets of vectors." *Acta Crystallographica Section A* 34(5): 827-828.
3. Coutsiias, E.A., Seok, C., & Dill, K.A. (2004). "Using quaternions to calculate RMSD." *Journal of Computational Chemistry* 25(15): 1849-1857.
4. Theobald, D.L. (2005). "Rapid calculation of RMSDs using a quaternion-based characteristic polynomial." *Acta Crystallographica Section A* 61(4): 478-480.