

RESEARCH ARTICLE

Algebraic constraints and algorithms for common lines in cryo-EM

Tommi Muller¹ , Adriana L. Duncan², Eric J. Verbeke³ and Joe Kileel^{2,4}

¹Mathematical Institute, University of Oxford, Oxford, UK

²Department of Mathematics, University of Texas at Austin, Austin, TX, USA

³Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ, USA

⁴Oden Institute, University of Texas at Austin, Austin, TX, USA

Corresponding author: Tommi Muller; Email: tommi.muller@maths.ox.ac.uk

Received: 23 November 2023; **Revised:** 29 April 2024; **Accepted:** 03 May 2024

Keywords: ADMM; common lines; cryo-electron microscopy; discrete heterogeneity; group synchronization; low-rank matrix

Abstract

We revisit the topic of common lines between projection images in single-particle cryo-electron microscopy (cryo-EM). We derive a novel low-rank constraint on a certain $2n \times n$ matrix storing properly scaled basis vectors for the common lines between n projection images of one molecular conformation. Using this algebraic constraint and others, we give optimization algorithms to denoise common lines and recover the unknown 3D rotations associated with the images. As an application, we develop a clustering algorithm to partition a set of noisy images into homogeneous communities using common lines, in the case of discrete heterogeneity in cryo-EM. We demonstrate the methods on synthetic and experimental datasets.

Impact Statement

Single-particle cryo-electron microscopy (cryo-EM) is an imaging technique used to determine the 3D structure of biomolecules from noisy 2D projection images. This article revisits one of the first approaches to cryo-EM image processing, namely common lines between pairs of 2D class averages coming from the Fourier slice theorem. We present a novel mathematical approach for dealing with common lines: in contrast to some alternatives, it operates directly on the common lines themselves and avoids triplewise angular reconstitution completely. The article then derives novel algebraic constraints on sets of consistent common lines, including a straightforward low-rank matrix condition. The algebraic conditions are incorporated into optimization methods arising from the field of computer vision to produce new methods for computational tasks involving common lines. In particular, we achieve improved accuracy in common line denoising and rotation recovery at low signal-to-noise ratios. We also present a method to detect homogeneous communities of 2D class averages in the case of a cryo-EM dataset with multiple molecular conformations. Altogether this work clarifies a classic topic in cryo-EM, and opens the door to applying common lines techniques on more challenging datasets.

1. Introduction

Single-particle cryo-electron microscopy (cryo-EM) is an imaging technique capable of recovering the high-resolution 3D structure of molecules from many noisy tomographic projection images taken at unknown viewing angles. One of the first approaches for 3D reconstruction, known as angular reconstitution, is based on the common line property of projection images induced by the Fourier slice theorem.^(1,2) Due to the low signal-to-noise ratio (SNR) in cryo-EM data, detecting common lines is a

difficult task⁽³⁾: even today when applied to denoised averages of images, referred to as 2D class averages. Detecting common lines is subject to angular errors and incorrectly identified common lines. Although methods which seek to minimize global errors in the estimated viewing directions have increased the utility of common lines methods,⁽⁴⁾ additional constraints on common lines are needed to improve their accuracy and robustness.

In this article, we propose a novel approach for dealing with common lines. Specifically, we assemble the estimated common lines for a dataset of n images into a certain $2n \times n$ matrix, which stores properly scaled basis vectors for the common lines (Theorem 3.2). The matrix directly encodes common lines data, without requiring angular reconstitution on various subsets of images or needing voting procedures like some existing formulations.^(3,5) As such, it yields a direct and more global approach than prior constructions for common lines.

As a main contribution, we derive algebraic constraints on the matrix of common lines, which must be satisfied in order for a set of common lines to be consistent with a single asymmetric molecular conformation. The constraints include a straightforward low-rank condition on the matrix, as well as various sparse quadratic constraints. Importantly, the constraints enable new strategies for computational tasks involving common lines, in particular for denoising common lines; estimating 3D rotations; and clustering heterogeneous image sets into homogeneous subsets. We demonstrate this by adapting optimization algorithms from other domains to these tasks, using the algebraic constraints. We remark that our constraints seem better suited for numerical optimization than the semialgebraic constraints found in prior work.⁽⁶⁾

Notably, the clustering problem is a recent application of common lines.⁽⁷⁾ In more detail, the goal is to sort discretely heterogeneous image sets of multiple molecules into communities corresponding to homogeneous image subsets. This application is motivated by the increasing complexity of cryo-EM datasets, where samples may not be purified and thus the number of distinct molecules contained in a dataset is more than one.^(8–10) Our algebraic constraints and optimization algorithms enable consistency checks of subsets of images, to test whether the subset corresponds to a single molecule.

As a mathematical guarantee, we prove that computing the correct scales in the homogeneous case admits an essentially unique global optimum, see Theorem 5.1. We implement our algorithms and test them on simulated and real datasets in Section 7. The results demonstrate that our methods can be successful when applied to 2D class averages at noise levels comparable to experimental data, in both the homogeneous and discretely heterogeneous cases. We conclude with a discussion of potential future improvements.

1.1. Advantages

There are several advantages to our approach for dealing with common lines:

- The new formulation is directly in terms of the data, that is, in terms of the common lines themselves.
- It involves multiple common lines simultaneously and does not require triplewise angular reconstitution at all (in contrast to (5) for instance), making our approach fully global and potentially more robust to noise than alternatives.
- The algebraic constraints can be incorporated into existing optimization algorithms that have seen success in computer vision applications.⁽¹¹⁾
- The resulting algorithms outperform existing methods for denoising and rotation recovery on noisy simulated data, and perform comparably well for clustering heterogeneous image sets on real data, even though the optimization algorithms are off-the-shelf.

2. Background

First, we recall a standard simplified mathematical model for cryo-EM, in the homogeneous case of one molecular conformation. We assume there exists a 3D function $\varphi: \mathbb{R}^3 \rightarrow \mathbb{R}$ describing the electrostatic potential generated by the molecule. As data, we receive n two-dimensional tomographic projection

images, denoted $I_{R^{(i)}} : \mathbb{R}^2 \rightarrow \mathbb{R}$ for $i = 1, \dots, n$, where $R^{(i)} \in \text{SO}(3)$ are 3D rotations associated with each image. The goal of single-particle cryo-EM is to recover the underlying 3D structure φ from the set of 2D tomographic projection images which are observed at unknown rotations. The images, in their idealized and noiseless form, have the following Fourier transforms due to the Fourier slice theorem:

$$\widehat{I}_{R^{(i)}}(\widehat{x}, \widehat{y}) = \left(R^{(i)} \cdot \widehat{\varphi} \right)(\widehat{x}, \widehat{y}, 0). \quad (1)$$

Here $\widehat{\varphi}(\widehat{x}, \widehat{y}, \widehat{z}) := \int_{\mathbb{R}^3} \varphi(x, y, z) e^{\sqrt{-1}(\widehat{x}x + \widehat{y}y + \widehat{z}z)} dx dy dz$ denotes the Fourier transform of φ , and $R^{(i)} \cdot \widehat{\varphi}$ denotes the rotation of $\widehat{\varphi}$ by $R^{(i)}$. Writing $R^{(i)} = \begin{pmatrix} \mathbf{r}_1^{(i)} & \mathbf{r}_2^{(i)} & \mathbf{r}_3^{(i)} \end{pmatrix}^\top$, (1) reads

$$\widehat{I}_{R^{(i)}}(\widehat{x}, \widehat{y}) = \widehat{\varphi} \left(\left(R^{(i)} \right)^\top (\widehat{x} \widehat{y} 0)^\top \right) = \widehat{\varphi}(\widehat{x} \mathbf{r}_1^{(i)} + \widehat{y} \mathbf{r}_2^{(i)}). \quad (2)$$

Generically, for asymmetric molecules φ and distinct rotations $R^{(i)}$ and $R^{(j)}$, there exist unique lines through the origin in the domain of the Fourier-transformed images $\widehat{I}_{R^{(i)}}$ and $\widehat{I}_{R^{(j)}}$, respectively $\mathcal{L}_{ij} \subseteq \text{domain}(\widehat{I}_{R^{(i)}}) = \mathbb{R}^2$ and $\mathcal{L}_{ji} \subseteq \text{domain}(\widehat{I}_{R^{(j)}}) = \mathbb{R}^2$, such that the restrictions

$$\widehat{I}_{R^{(i)}}|_{\mathcal{L}_{ij}} = \widehat{I}_{R^{(j)}}|_{\mathcal{L}_{ji}} \quad (3)$$

are equal as functions on $\mathbb{R}^2$. In cryo-EM, one says that $\mathcal{L}_{ij}$ and $\mathcal{L}_{ji}$ are the *common lines* between the i th and j th image. In modest-noise settings, which are arrived at by working with 2D class averages instead of raw tomographic images,⁽¹²⁾ common lines can be estimated from real cryo-EM data. They give basic ways to do 3D reconstruction in cryo-EM; for example, see the angular reconstitution technique of van Heel⁽²⁾ or the works of Shkolnisky, Singer, and their collaborators^(3–5) for example.

From (2), the common lines $\mathcal{L}_{ij}$ and $\mathcal{L}_{ji}$ may be found mathematically by expressing the single line in 3D space:

$$\text{span}(\mathbf{r}_1^{(i)}, \mathbf{r}_2^{(i)}) \cap \text{span}(\mathbf{r}_1^{(j)}, \mathbf{r}_2^{(j)}) = \text{span}(\mathbf{r}_3^{(i)})^\perp \cap \text{span}(\mathbf{r}_3^{(j)})^\perp = \text{span}(\mathbf{r}_3^{(i)} \times \mathbf{r}_3^{(j)}) \subseteq \text{dom}(\widehat{\varphi}) = \mathbb{R}^3 \quad (4)$$

in the coordinate system of the i th and j th images, respectively. Here, $\times$ denotes the cross product in $\mathbb{R}^3$. Combining (2) and (4), (3) may be written as:

$$\begin{aligned} \widehat{I}_{R^{(i)}}(\lambda \widehat{x}_{ij}, \lambda \widehat{y}_{ij}) &:= \widehat{I}_{R^{(j)}}(\lambda \widehat{x}_{ji}, \lambda \widehat{y}_{ji}) \quad \text{for all } \lambda \in \mathbb{R}, \\ \text{where } \widehat{x}_{ij} &:= \langle \mathbf{r}_1^{(i)}, \mathbf{r}_3^{(i)} \times \mathbf{r}_3^{(j)} \rangle, \quad \widehat{y}_{ij} := \langle \mathbf{r}_2^{(i)}, \mathbf{r}_3^{(i)} \times \mathbf{r}_3^{(j)} \rangle, \\ \text{and } \widehat{x}_{ji} &:= -\langle \mathbf{r}_1^{(j)}, \mathbf{r}_3^{(j)} \times \mathbf{r}_3^{(i)} \rangle, \quad \widehat{y}_{ji} := -\langle \mathbf{r}_2^{(j)}, \mathbf{r}_3^{(j)} \times \mathbf{r}_3^{(i)} \rangle \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^3$. Common lines can therefore be encoded via:

Definition 2.1. Vectors $\mathbf{a}_{ij}, \mathbf{a}_{ji} \in \mathbb{R}^2$ are called *representatives* for the common lines $\mathcal{L}_{ij}$ and $\mathcal{L}_{ji}$ if there exists a nonzero scalar $\lambda_{ij} = \lambda_{ji} \in \mathbb{R}$ such that

$$\mathbf{a}_{ij} = \lambda_{ij} \begin{pmatrix} \langle \mathbf{r}_1^{(i)}, \mathbf{r}_3^{(i)} \times \mathbf{r}_3^{(j)} \rangle \\ \langle \mathbf{r}_2^{(i)}, \mathbf{r}_3^{(i)} \times \mathbf{r}_3^{(j)} \rangle \end{pmatrix}, \quad \mathbf{a}_{ji} = \lambda_{ji} \begin{pmatrix} -\langle \mathbf{r}_1^{(j)}, \mathbf{r}_3^{(j)} \times \mathbf{r}_3^{(i)} \rangle \\ -\langle \mathbf{r}_2^{(j)}, \mathbf{r}_3^{(j)} \times \mathbf{r}_3^{(i)} \rangle \end{pmatrix}. \quad (5)$$

Equivalently, representatives $\mathbf{a}_{ij}$ and $\mathbf{a}_{ji}$ are choices of basis vectors for the common lines $\mathcal{L}_{ij}$ and $\mathcal{L}_{ji}$ which satisfy $\widehat{I}_{R^{(i)}}(\lambda \mathbf{a}_{ij}) = \widehat{I}_{R^{(j)}}(\lambda \mathbf{a}_{ji})$ for all $\lambda \in \mathbb{R}$. Representatives for common lines can be estimated from 2D class averages in practice.

We stress that, although quite standard, the model (1) is greatly simplified. It neglects the effects of contrast transfer functions (CTFs), imperfect centering in particle picking, and blurring in class averaging. Further, we have restricted attention to the case of asymmetric molecules, as otherwise common lines are only unique up to the action of the relevant symmetry group (e.g., see (13)).

3. Constraints on sets of common lines

3.1. The common lines matrix

We introduce an object to keep track of all common lines in a dataset. It is the main object of this article.

Definition 3.1. A *common lines matrix* associated to rotations $R^{(1)}, \dots, R^{(n)} \in \text{SO}(3)$ is a matrix $A \in \mathbb{R}^{2n \times n}$, which when regarded as an $n \times n$ block matrices with 2×1 blocks $\mathbf{a}_{ij} \in \mathbb{R}^2$ is such that $\mathbf{a}_{ij}, \mathbf{a}_{ji}$ are representatives for the common lines ℓ_{ij}, ℓ_{ji} if i and j are distinct and $\mathbf{a}_{ii} = 0$ otherwise. If the scalars λ_{ij} in (5) are all equal to 1, then we call A the *pure common lines matrix*.

Thus a common lines matrix A associated to $R^{(1)}, \dots, R^{(n)}$ is uniquely defined up to $\binom{n}{2}$ nonzero real scalars λ_{ij} ($i < j$). In real data settings where the 2D class averages are sufficiently denoised, we can estimate A from data by estimating representatives for the common lines.

We now present constraints which a *pure* common lines matrix must satisfy. Firstly, there is the following low-rank condition. All of our computational methods take advantage of this.

Theorem 3.2. Let $A \in \mathbb{R}^{2n \times n}$ be the pure common lines matrix associated to Zariski-generic rotations $R^{(1)}, \dots, R^{(n)} \in \text{SO}(3)$ where $n \geq 3$. Then $\text{rank}(A) = 3$.

Proof. Since

$$\mathbf{a}_{ij} = \begin{pmatrix} \langle \mathbf{r}_1^{(i)}, \mathbf{r}_3^{(i)} \times \mathbf{r}_3^{(j)} \rangle \\ \langle \mathbf{r}_2^{(i)}, \mathbf{r}_3^{(i)} \times \mathbf{r}_3^{(j)} \rangle \end{pmatrix} = \begin{pmatrix} \langle \mathbf{r}_1^{(i)} \times \mathbf{r}_3^{(i)}, \mathbf{r}_3^{(j)} \rangle \\ \langle \mathbf{r}_2^{(i)} \times \mathbf{r}_3^{(i)}, \mathbf{r}_3^{(j)} \rangle \end{pmatrix} = \begin{pmatrix} -\langle \mathbf{r}_2^{(i)}, \mathbf{r}_3^{(j)} \rangle \\ \langle \mathbf{r}_1^{(i)}, \mathbf{r}_3^{(j)} \rangle \end{pmatrix} = \begin{pmatrix} -\mathbf{r}_2^{(i)\top} \\ \mathbf{r}_1^{(i)\top} \end{pmatrix}_{2 \times 3} \mathbf{r}_3^{(j)},$$

the pure common line matrix admits the following factorization:

$$A = \begin{pmatrix} -\mathbf{r}_2^{(1)\top} \\ \mathbf{r}_1^{(1)\top} \\ \vdots \\ -\mathbf{r}_2^{(n)\top} \\ \mathbf{r}_1^{(n)\top} \end{pmatrix}_{2n \times 3} \begin{pmatrix} \mathbf{r}_3^{(1)} & \dots & \mathbf{r}_3^{(n)} \end{pmatrix}_{3 \times n}. \quad (6)$$

Equation (6) witnesses $\text{rank}(A) \leq \min(3, n) = 3$. We have equality when $R^{(i)}$ are generic because the two matrices in the factorization are full rank.

There are also necessary quadratic constraints in the entries of a pure common lines matrix.

Proposition 3.3. Suppose $A \in \mathbb{R}^{2n \times n}$ is a pure common lines matrix where $n \geq 3$. Then for any $1 \leq i < j \leq n$, we have $\|\mathbf{a}_{ij}\|_2^2 = \|\mathbf{a}_{ji}\|_2^2$.

Proof. See Appendix A.

Proposition 3.4. Suppose $A \in \mathbb{R}^{2n \times n}$ is a pure common lines matrix where $n \geq 3$. Then for any $1 \leq i < j < k \leq n$, we have $\det(\mathbf{a}_{ij} \ \mathbf{a}_{ik}) = -\det(\mathbf{a}_{ji} \ \mathbf{a}_{jk}) = \det(\mathbf{a}_{ki} \ \mathbf{a}_{kj})$.

Proof. See Appendix A.

From Propositions 3.3 and 3.4, the number of quadratic constraints on a pure common lines matrix equals $\binom{n}{2} + 2\binom{n}{3}$.

Example 1. Consider $n=4$. Then a pure common lines matrix written in block form,

$$A = \begin{pmatrix} \mathbf{0} & \mathbf{a}_{12} & \mathbf{a}_{13} & \mathbf{a}_{14} \\ \mathbf{a}_{21} & \mathbf{0} & \mathbf{a}_{23} & \mathbf{a}_{24} \\ \mathbf{a}_{31} & \mathbf{a}_{32} & \mathbf{0} & \mathbf{a}_{34} \\ \mathbf{a}_{41} & \mathbf{a}_{42} & \mathbf{a}_{43} & \mathbf{0} \end{pmatrix}$$

has rank at most 3 and satisfies 14 quadratic equations, which are

$$\begin{aligned} \|\mathbf{a}_{12}\|_2^2 &= \|\mathbf{a}_{21}\|_2^2 & \|\mathbf{a}_{14}\|_2^2 &= \|\mathbf{a}_{41}\|_2^2 \\ \|\mathbf{a}_{13}\|_2^2 &= \|\mathbf{a}_{31}\|_2^2 & \|\mathbf{a}_{24}\|_2^2 &= \|\mathbf{a}_{42}\|_2^2 \\ \|\mathbf{a}_{23}\|_2^2 &= \|\mathbf{a}_{32}\|_2^2 & \|\mathbf{a}_{34}\|_2^2 &= \|\mathbf{a}_{43}\|_2^2 \end{aligned}$$

$$\begin{aligned} \det(\mathbf{a}_{12} \quad \mathbf{a}_{13}) &= -\det(\mathbf{a}_{21} \quad \mathbf{a}_{23}) = \det(\mathbf{a}_{31} \quad \mathbf{a}_{32}) \\ \det(\mathbf{a}_{12} \quad \mathbf{a}_{14}) &= -\det(\mathbf{a}_{21} \quad \mathbf{a}_{24}) = \det(\mathbf{a}_{41} \quad \mathbf{a}_{42}) \\ \det(\mathbf{a}_{13} \quad \mathbf{a}_{14}) &= -\det(\mathbf{a}_{31} \quad \mathbf{a}_{34}) = \det(\mathbf{a}_{41} \quad \mathbf{a}_{43}) \\ \det(\mathbf{a}_{23} \quad \mathbf{a}_{24}) &= -\det(\mathbf{a}_{32} \quad \mathbf{a}_{34}) = \det(\mathbf{a}_{42} \quad \mathbf{a}_{43}) \end{aligned}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. Note that $\text{rank}(A) \leq 3$ is equivalent to the vanishing of all 4×4 minors of A , giving a collection of homogeneous degree 4 polynomial constraints on the entries of A .

We note that (6) furnishes a polynomial map which sends an n -tuple of rotations to a pure common lines matrix:

$$\begin{aligned} \psi : \text{SO}(3)^n &\rightarrow \mathbb{R}^{2n \times n} \\ (R^{(1)}, \dots, R^{(n)}) &\mapsto A. \end{aligned} \tag{7}$$

Studying ψ will allow us understand additional important properties of pure common lines matrices. To do this, we will need to introduce some terminology and elementary concepts from algebraic geometry (see (14) for precise definitions.)

A subset $X \subseteq \mathbb{R}^d$ is called an *algebraic variety* if it is the set of points in $\mathbb{R}^d$ where a finite collection of polynomials all simultaneously equal 0. For example, $\text{SO}(3)$ is an algebraic variety since it is the set of matrices R in $\mathbb{R}^{3 \times 3} \cong \mathbb{R}^9$ satisfying the polynomial equations $R^\top R - I_{3 \times 3} = \mathbf{0}_{3 \times 3}$ and $\det(R) - 1 = 0$. Roughly speaking, an algebraic variety is similar to an embedded manifold, except possibly singular and always defined by polynomial equations. Due to the properties of polynomials, an algebraic variety X is a “thin” subset of $\mathbb{R}^d$ in which it lives: provided $X \neq \mathbb{R}^d$, the complement of X is always a dense subset filling up almost the entirety of the ambient space. More precisely, if one samples a random point from $\mathbb{R}^d$ according to any absolutely continuous probability distribution, then with probability 1 the point will lie in the complement of X . We say that some property P holds (*Zariski*) *generically* if it holds for all points in the complement of some algebraic variety $X \subsetneq \mathbb{R}^d$, and we call such points (*Zariski*) *generic*. Roughly speaking, this means that property P holds with probability 1 (even if, as usually the case, the variety $X \subsetneq \mathbb{R}^d$ is left unspecified).

Recall that the *fiber* of a map at a point p in its image is the set of points in its domain which map to p . Therefore to answer the question, “Does a pure common lines matrix uniquely determine the rotations which generated it?” we need to understand the fibers of the map ψ in (7). Our next result shows that the answer to the question is “Yes,” up to a global rotation, provided the pure common lines matrix is generic.

Theorem 3.5. For $n \geq 3$, generically, the fibers of the map ψ are isomorphic to $SO(3)$. More precisely, for generic rotations $R^{(1)}, \dots, R^{(n)}$ it holds that

$$\psi^{-1}\left(\psi\left(R^{(1)}, \dots, R^{(n)}\right)\right)=\left\{\left(R^{(1)} Q, \dots, R^{(n)} Q\right): Q \in SO(3)\right\}.$$

Proof. See [Appendix A](#).

Remark 3.6. **Theorem 3.5** does not contradict the chirality ambiguity in cryo-EM, which states that the 3D molecule and rotations can only ever be recovered up to a global rotation and global reflection given cryo-EM data. In [Theorem 5.1](#), we prove that there are two possible pure common lines matrices for a given non-pure common lines matrix. They differ by a global sign, and correspond to rotation tuples $(R^{(1)}, \dots, R^{(n)})$ and $(JR^{(1)}, \dots, JR^{(n)})$ where $J = \text{diag}(-1, -1, 1)$, respectively. The chirality or handedness ambiguity is well-known in the common lines literature^(2,5) and unavoidable.

3.2. The common lines variety

In general, the image of a polynomial map from an algebraic variety is not an algebraic variety, because polynomial *inequalities* (in addition to equations) are needed in the description of the image.⁽¹⁵⁾ This means that the set of all pure common lines matrices, that is, the image of ψ from $SO(3)$, on its own is not an algebraic variety. To resolve this, we consider the smallest algebraic variety in $\mathbb{R}^{2n \times n}$ containing $\psi(SO(3))$, that is, we add the smallest set of additional points (which are not pure common lines matrices) to $\psi(SO(3))$ until the union becomes an algebraic variety. The process is called taking the *Zariski closure* of $\psi(SO(3))$. We call the resulting algebraic variety the *common lines variety* and it lives in the ambient space $\mathbb{R}^{2n \times n}$. The common lines variety is defined by polynomial equations in the entries of a matrix $A \in \mathbb{R}^{2n \times n}$. Since the common lines variety includes all pure common lines matrices, the polynomials defining it, in particular, include the constraints we already identified in [Section 3.1](#).

Example 2. When $n = 3$, we used the computer algebra system Macaulay2⁽¹⁶⁾ to determine the collection of polynomial equations defining the common lines variety. Along with the 5 quadratic polynomials from [Propositions 3.3](#) and [3.4](#), our computation also found 1 polynomial of degree 6, 64 polynomials of degree 8, and 24 polynomials of degree 10, for a total of 94 polynomial equations. Notice that for $n = 3$, the rank 3 constraint of [Theorem 3.2](#) is vacuous. We find the 5 quadratics are the only homogeneous polynomials. The other 89 equations are highly complex and we refrain from explicitly writing them here. They are available at the GitHub repository (8.1).

In view of [Example 2](#), we believe that [Section 3.1](#) identifies all “simple-to-describe” algebraic constraints on pure common lines matrices. As such, it is important to understand to what extent these constraints are enough to characterize pure common lines matrices. This requires understanding the geometry of the common lines variety better, for which we will need to use a couple more basic concepts from algebraic geometry described in the next two paragraphs.

In general, every algebraic variety $X \subseteq \mathbb{R}^d$ admits a unique decomposition into a finite union of *irreducible components* $X = \bigcup_{i=1}^r X_i$, where $X_i \subseteq \mathbb{R}^d$ are algebraic varieties themselves and each cannot be decomposed as a union of two strictly smaller varieties. We think of X_i as “building blocks” of X . For example, the variety $\{(x, y) \in \mathbb{R}^2 : xy = 0\}$ is a union of the x - and y -axes, and these lines are its irreducible components. In general, each irreducible component X_i can be ascribed a *dimension*, which captures the number of degrees of freedom in X_i and coincides with manifold dimension when X_i is smooth. We note that the dimension of different irreducible components X_i of X may differ.

Given an algebraic variety $X \subseteq \mathbb{R}^d$, one can construct from it a larger algebraic variety $C(X) \subseteq \mathbb{R}^d$ called the *cone* over X by adding to X all points in $\mathbb{R}^d$ which lie on a line passing through the origin and a point on X . This constructs an algebraic variety that includes all scalar multiples of points of X .

Since the constraints we identified in [Section 3.1](#) are all polynomial equations, they define [Section 3.1](#) as an algebraic variety in $\mathbb{R}^{2n \times n}$. In the next proposition, we show that this algebraic variety contains the cone over the common lines variety as an irreducible component. This means that locally on this component, our constraints from [Section 3.1](#) are sufficient to characterize pure common lines matrices up to scale. The proof relies on computer algebra software,⁽¹⁶⁾ and checks that certain numerical matrices are full-rank, so we also report the range of n on which the proposition has been confirmed.

Proposition 3.7. *The algebraic variety defined by the low-rank constraint in [Theorem 3.2](#), along with the quadratic constraints in [Propositions 3.3](#) and [3.4](#), and the requirement that all diagonal blocks are 0, contains the cone over the common lines variety in $\mathbb{R}^{2n \times n}$ as an irreducible component for $n = 3, \dots, 50$.*

Proof. See [Appendix A](#).

4. Optimization problem

We encode the common lines from cryo-EM data by choosing representatives $\hat{\mathbf{a}}_{ij} \in \mathbb{R}^2$ ([Definition 2.1](#)) to form the 2×1 blocks in a common lines matrix $\hat{A} \in \mathbb{R}^{2n \times n}$. Suppose we have rescaled the 2×1 blocks of $\hat{A}$ so they all have norm 1. Then at least in clean situations, [Theorem 3.2](#) and [Propositions 3.3](#) and [3.4](#) imply we can scale the blocks by nonzero scalars λ_{ij} with $\lambda_{ij} = \lambda_{ji}$ so that the resulting matrix is a *pure* common lines matrix A , and thus has rank 3 and satisfies the set of norm and 2×2 determinant equations. [Proposition 3.7](#) states that these constraints are sufficient to determine the common lines variety locally. In [Section 5.4](#), we further prove that for purposes of recovering scales λ_{ij} to obtain a pure common lines matrix, the constraints are also sufficient.

Proper scales are not directly available from common lines data in cryo-EM. To find the scales we formulate an optimization problem, inspired by work for a mathematically similar problem in (11):

$$\min_{\substack{\{\mathbf{a}_{ij}\}, \{\lambda_{ij}\} \\ i, j = 1, \dots, n}} \sum_{i, j = 1}^n \|\hat{\mathbf{a}}_{ij} - \lambda_{ij} \mathbf{a}_{ij}\|_2 \quad (8)$$

$$\text{subject to } \begin{cases} \mathbf{a}_{ii} = 0 & \text{for all } 1 \leq i \leq n \\ \text{rank}(A) = 3 \\ \lambda_{ij} = \lambda_{ji} & \text{for all } 1 \leq i < j \leq n, \end{cases} \quad (9)$$

$$\begin{cases} \|\mathbf{a}_{ij}\|_2^2 = \|\mathbf{a}_{ji}\|_2^2 \\ \det(\mathbf{a}_{ij} \quad \mathbf{a}_{ik}) = -\det(\mathbf{a}_{ji} \quad \mathbf{a}_{jk}) = \det(\mathbf{a}_{ki} \quad \mathbf{a}_{kj}) \\ \text{for all } 1 \leq i < j < k \leq n. \end{cases} \quad (10)$$

The mixed L1/Frobenius norm $\|\cdot\|_2$ in the objective is chosen for its robustness to outliers.

Once we obtain a pure common lines matrix, we show in [Section 5.3](#) how to recover the rotations corresponding to the common lines (up to the ambiguity in [Remark 3.6](#)). Later in [Section 6](#), we solve the problem (8) to identify homogeneous clusters among images coming from a discrete number of distinct molecules.

5. Optimization algorithms

Our approach to solving (8) is first to solve the problem with constraint (9) only, and then to enforce the constraint (10) on the solution. These steps are in [Sections 5.1](#) and [5.2](#), respectively.

5.1. IRLS and ADMM for the rank constraint

To solve (8) with the rank constraint, we closely follow the approach of (11). We relax the mixed L1/Frobenius norm to a weighted least squares objective, where the weights and optimization variables are updated after each iteration of minimization via a procedure called Iterative Reweighted Least Squares (IRLS).⁽¹⁷⁾ Let t denote the IRLS iteration number. Then the objective (8) becomes:

$$\min_{A \in \mathbb{R}^{2n \times n}, \Lambda \in \mathbb{R}^{n \times n}} \|\hat{A} - (\Lambda \otimes \mathbf{1}_{2 \times 1}) \odot A\|_{WF}^2, \quad (11)$$

where $\otimes$ and $\odot$ is the Kronecker and Hadamard product of two matrices respectively, $\Lambda_{ij} = \lambda_{ij}$,

$$\|M\|_{WF}^2 := \sum_{i,j=1}^n w_{ij}^{(t)} \|\mathbf{m}_{ij}\|_2^2$$

is the squared weighted Frobenius norm of a block matrix $M \in \mathbb{R}^{2n \times n}$, the weights in (11) are

$$w_{ij}^{(t)} = \begin{cases} 1 / \left(\max \left\{ \delta, \|\hat{\mathbf{a}}_{ij} - \lambda_{ij}^{(t-1)} \mathbf{a}_{ij}^{(t-1)}\|_2 \right\} \right) & \text{if } i \neq j \\ 0 & \text{if } i = j, \end{cases} \quad (12)$$

and $0 < \delta \ll 1$ is a chosen regularization parameter.

Within each iteration of IRLS, we need to solve the problem (11) with the constraints (9). Since the objective is bilinear in A and Λ , we can do this using the Alternate Direction Method of Multiplier (ADMM).^(18,19) This gives an augmented Lagrangian optimization problem:

$$\begin{aligned} \max_{\Gamma \in \mathbb{R}^{2n \times n}} \min_{A, B \in \mathbb{R}^{2n \times n}, \Lambda \in \mathbb{R}^{n \times n}} & \frac{1}{2} \|\hat{A} - (\Lambda \otimes \mathbf{1}_{2 \times 1}) \odot A\|_{WF}^2 + \frac{\tau}{2} \|B - (A + \Gamma)\|_F^2 \\ \text{subject to} & \begin{cases} \mathbf{a}_{ii} = \mathbf{0} \text{ for all } 1 \leq i \leq n \\ \Lambda = \Lambda^\top \\ \text{rank}(B) = 3, \end{cases} \end{aligned} \quad (13)$$

where Γ is a matrix of Lagrange multipliers and $\tau = \sum_{i,j=1}^n w_{ij}^{(t)}$. We now describe the steps of the ADMM procedure. Since the problem (13) is non-convex, we alternately optimize for each variable. In the following, let k denote the ADMM iteration number and let $W^{(t)} \in \mathbb{R}^{n \times n}$ where $W_{ij}^{(t)} = w_{ij}^{(t)}$ be the matrix of weights within IRLS iteration t .

1. *Optimize A and Λ :* We alternately optimize for A and Λ until convergence. Let k' denote the iteration number for this step.

1a. First we solve the unconstrained problem for A :

$$\begin{aligned} \text{The solution is } \min_{A \in \mathbb{R}^{2n \times n}} & \frac{1}{2} \|\hat{A} - (\Lambda^{(k')} \otimes \mathbf{1}_{2 \times 1}) \odot A\|_{WF}^2 + \frac{\tau}{2} \|B^{(k)} - (A + \Gamma^{(k)})\|_F^2. \\ A^{(k'+1)} = & \left(\left(W^{(t)} \otimes \mathbf{1}_{2 \times 1} \right) \odot \left(\Lambda^{(k')} \otimes \mathbf{1}_{2 \times 1} \right) \odot \hat{A} + \frac{\tau}{4} \left(B^{(k)} + \Gamma^{(k)} \right) \right) \\ & \oslash \left(\left(W^{(t)} \otimes \mathbf{1}_{2 \times 1} \right) \odot \left(\Lambda^{(k')} \otimes \mathbf{1}_{2 \times 1} \right) \odot \left(\Lambda^{(k')} \otimes \mathbf{1}_{2 \times 1} \right) + \frac{\tau}{4} \mathbf{1}_{2n \times n} \right), \end{aligned} \quad (14)$$

where $\oslash$ is the element-wise division of two matrices. Then we project A onto the set of matrices whose 2×1 diagonals are 0:

$$\mathbf{a}_{ii}^{(k'+1)} = \mathbf{0}. \quad (15)$$

1b. Next we solve the unconstrained problem for Λ :

$$\begin{aligned} \min_{\Lambda \in \mathbb{R}^{n \times n}} & \frac{1}{2} \|\hat{A} - (\Lambda \otimes \mathbf{1}_{2 \times 1}) \odot A^{(k'+1)}\|_{WF}^2 \\ \text{subject to} & \Lambda = \Lambda^\top \end{aligned}$$

The solution is

$$\lambda_{ij}^{(k'+1)} = \begin{cases} \frac{w_{ij} \langle \hat{\mathbf{a}}_{ij}, \mathbf{a}_{ij}^{(k'+1)} \rangle + w_{ji} \langle \hat{\mathbf{a}}_{ji}, \mathbf{a}_{ji}^{(k'+1)} \rangle}{w_{ij} \|\mathbf{a}_{ij}^{(k'+1)}\|_2^2 + w_{ji} \|\mathbf{a}_{ji}^{(k'+1)}\|_2^2} & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases} \quad (16)$$

After repeating 1a. and 1b. until convergence, we obtain $A^{(k+1)}$ and $\Lambda^{(k+1)}$.

2. *Optimize B*: The constrained problem for B is

$$\begin{aligned} \min_{B \in \mathbb{R}^{2n \times n}} \quad & \frac{\tau}{2} \|B - (A^{(k+1)} + \Gamma^{(k)})\|_F^2 \\ \text{subject to} \quad & \text{rank}(B) = 3 \end{aligned}$$

This is solved by

$$B^{(k+1)} = \text{SVP}(A^{(k+1)} - \Gamma^{(k)}, 3), \quad (17)$$

where $\text{SVP}(M, 3)$ is the singular value projection of a matrix M onto the set of matrices of rank at most 3, which is computing by taking the highest three singular values of M and its corresponding left and right singular vectors.

3. *Update Γ* : In ADMM, there is a gradient ascent step for Γ , where the step is the solution to

$$\max_{\Gamma \in \mathbb{R}^{2n \times n}} \|B^{(k+1)} - (A^{(k+1)} - \Gamma)\|_F^2.$$

This gives the update

$$\Gamma^{(k+1)} = \Gamma^{(k)} + (B^{(k+1)} - A^{(k+1)}). \quad (18)$$

Steps 1, 2, and 3 are repeated until convergence in the optimization variables. This completes the ADMM procedure for IRLS iteration t . The IRLS weights $w_{ij}^{(t+1)}$ for iteration $t+1$ are updated using (12), and the ADMM procedure is repeated again. The whole pipeline is detailed in Algorithm 1.

Algorithm 1. IRLS and ADMM for rank constraint satisfaction

Input: $\hat{A} \in \mathbb{R}^{2n \times n}$, a common lines matrix

Output: $A \in \mathbb{R}^{2n \times n}$, a common lines matrix satisfying only the rank constraint

```

1: procedure IRLS-ADMM ( $\hat{A}$ )
2:   initialize  $A, \Lambda, W$ 
3:    $t \leftarrow 0$ 
4:   while not converged do
5:      $B \leftarrow A$ 
6:      $\Gamma \leftarrow 0_{2n \times n}$ 
7:      $\tau \leftarrow \sum_{i,j=1}^n w_{ij}^{(t)}$ 
8:      $k \leftarrow 0$ 
9:     while not converged do
10:       $k' \leftarrow 0$ 
11:      while not converged do
12:        update  $A$  using (14) and (15)
13:        update  $\Lambda$  using (16)
```

▷ 1. Update A and Λ

```

14:          $k' \leftarrow k' + 1$ 
15:     end while
16:     update  $B$  using (17) ▷ 2. Update  $B$ 
17:     update  $\Gamma$  using (18) ▷ 3. Update  $\Gamma$ 
18:      $k \leftarrow k + 1$ 
19: end while
20: update  $W$  using (12) ▷ Update IRLS weights
21:  $t \leftarrow t + 1$ 
22: end while
23: end procedure

```

5.2. Sinkhorn scaling for the quadratic constraints

When successful, IRLS-ADMM in Section 5.1 gives us a solution A to (8) satisfying the constraint (9). Next, we must enforce (10). As described below, our approach is to scale the rows and columns of A alternately until constraint (10) is satisfied, in a manner analogous to Sinkhorn's algorithm.⁽²⁰⁾ Note that nonzero row and column scales do not affect the rank of A , so constraint (9) will still be satisfied.

We find the row and column scales by solving least squares problems. First, we handle the norm constraints. Define $M \in \mathbb{R}^{n \times n}$ where

$$M_{ij} = \|\mathbf{a}_{ij}\|_2^2. \quad (19)$$

Then the norm constraints are satisfied if and only if $M = M^\top$, which leads us to the following constrained least squares problems:

$$\boldsymbol{\mu} = \arg \min_{\|\boldsymbol{\mu}\|_2 = 1} \|\text{diag}(\boldsymbol{\mu})M - (\text{diag}(\boldsymbol{\mu})M)^\top\|_F^2, \quad (20)$$

$$\boldsymbol{\tau} = \arg \min_{\|\boldsymbol{\tau}\|_2 = 1} \|M\text{diag}(\boldsymbol{\tau}) - (M\text{diag}(\boldsymbol{\tau}))^\top\|_F^2. \quad (21)$$

The solutions to problems (20) and (21) are

$$\min_{\|\boldsymbol{\mu}\|_2 = 1} \|N_L \boldsymbol{\mu}\|_2^2, \quad (22)$$

$$\min_{\|\boldsymbol{\tau}\|_2 = 1} \|N_R \boldsymbol{\tau}\|_2^2, \quad (23)$$

respectively, where $N_L, N_R \in \mathbb{R}^{n \times n}$ are the corresponding least squares matrices. See Appendix B for full details. The problems (22) and (23) are solved by taking $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$ to be the right singular vector corresponding to the smallest singular value of N_L and N_R , respectively.

Now we handle the determinant constraints. Scaling each $2 \times n$ row of A by $\mu_1, \dots, \mu_n \in \mathbb{R}$ and enforcing the constraints leads us to the equations

$$\mu_i^2 \det(\mathbf{a}_{ij} \ \mathbf{a}_{ik}) = -\mu_j^2 \det(\mathbf{a}_{ji} \ \mathbf{a}_{jk}) = \mu_k^2 \det(\mathbf{a}_{ki} \ \mathbf{a}_{kj}) \quad (24)$$

for all $1 \leq i < j < k \leq n$. Taking the signed root on each equation, we obtain

$$\begin{aligned} \mu_i \text{sgn}(\det(\mathbf{a}_{ij} \ \mathbf{a}_{ik})) \sqrt{|\det(\mathbf{a}_{ij} \ \mathbf{a}_{ik})|} &= -\mu_j \text{sgn}(\det(\mathbf{a}_{ji} \ \mathbf{a}_{jk})) \sqrt{|\det(\mathbf{a}_{ji} \ \mathbf{a}_{jk})|} \\ &= \mu_k \text{sgn}(\det(\mathbf{a}_{ki} \ \mathbf{a}_{kj})) \sqrt{|\det(\mathbf{a}_{ki} \ \mathbf{a}_{kj})|} \end{aligned} \quad (25)$$

Scaling the columns of A by $\tau_1, \dots, \tau_n \in \mathbb{R}$ and enforcing the constraints leads to the equations

$$\tau_j \tau_k \det(\mathbf{a}_{ij} \ \mathbf{a}_{ik}) = -\tau_i \tau_k \det(\mathbf{a}_{ji} \ \mathbf{a}_{jk}) = \tau_i \tau_j \det(\mathbf{a}_{ki} \ \mathbf{a}_{kj}) \quad (26)$$

for all $1 \leq i < j < k \leq n$. Dividing the first equation above by τ_k on both sides and the second equation above by τ_i on both sides, we obtain

$$\begin{aligned} \tau_j \det(\mathbf{a}_{ij} \quad \mathbf{a}_{ik}) &= -\tau_i \det(\mathbf{a}_{ji} \quad \mathbf{a}_{jk}) \\ -\tau_k \det(\mathbf{a}_{ji} \quad \mathbf{a}_{jk}) &= \tau_j \det(\mathbf{a}_{ki} \quad \mathbf{a}_{kj}) \end{aligned} \quad (27)$$

We observe that (25) and (27) are linear in μ_i and τ_i . We can enumerate all determinants into three vectors

$$\begin{aligned} \mathbf{v}_1 &= (\det(\mathbf{a}_{ij} \quad \mathbf{a}_{ik}))_{1 \leq i < j < k \leq n} \\ \mathbf{v}_2 &= (-\det(\mathbf{a}_{ji} \quad \mathbf{a}_{jk}))_{1 \leq i < j < k \leq n} \\ \mathbf{v}_3 &= (\det(\mathbf{a}_{ki} \quad \mathbf{a}_{kj}))_{1 \leq i < j < k \leq n}, \end{aligned} \quad (28)$$

each of length $\binom{n}{3}$. The determinant constraints are satisfied if and only if $\mathbf{v}_1 = \mathbf{v}_2 = \mathbf{v}_3$, which leads to the following constrained least squares problems:

$$\min_{\|\boldsymbol{\mu}\|_2=1} \|(\boldsymbol{\mu} \Delta \mathbf{v}_1) - (\boldsymbol{\mu} \Delta \mathbf{v}_2)\|_2^2 + \|(\boldsymbol{\mu} \Delta \mathbf{v}_2) - (\boldsymbol{\mu} \Delta \mathbf{v}_3)\|_2^2, \quad (29)$$

$$\min_{\|\boldsymbol{\tau}\|_2=1} \|(\boldsymbol{\tau} \Delta_1 \mathbf{v}_1) - (\boldsymbol{\tau} \Delta_1 \mathbf{v}_2)\|_2^2 + \|(\boldsymbol{\tau} \Delta_2 \mathbf{v}_2) - (\boldsymbol{\tau} \Delta_2 \mathbf{v}_3)\|_2^2, \quad (30)$$

where the quantities in brackets (defined in (B5) and (B6)) are the corresponding scalings (25) and (27) of $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ by $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$. The solutions to problems (29) and (30) are

$$\min_{\|\boldsymbol{\mu}\|_2=1} \|(D_{L,1} + D_{L,2})\boldsymbol{\mu}\|_2^2, \quad (31)$$

$$\min_{\|\boldsymbol{\tau}\|_2=1} \|(D_{R,1} + D_{R,2})\boldsymbol{\tau}\|_2^2, \quad (32)$$

respectively, where $D_{L,1}, D_{L,2}, D_{R,1}, D_{R,2} \in \mathbb{R}^{n \times n}$ are the corresponding least squares matrices. See [Appendix B](#) for full details. The problems (31) and (32) are again solved using SVD.

Now we describe the steps of the Sinkhorn scaling method. Let r denote the iteration number of the procedure.

1. *Scale rows*: Let $\boldsymbol{\mu} \in \mathbb{R}^n$ be the solution to

$$\min_{\|\boldsymbol{\mu}\|_2=1} \|(N_L + D_{L,1} + D_{L,2})\boldsymbol{\mu}\|_2^2. \quad (33)$$

Then we perform the update

$$A^{(r+\frac{1}{2})} = \frac{\|A^{(r)}\|_F^2}{\|(\text{diag}(\boldsymbol{\mu}) \otimes \mathbf{1}_{2 \times 1})A^{(r)}\|_F^2} (\text{diag}(\boldsymbol{\mu}) \otimes \mathbf{1}_{2 \times 1})A^{(r)}. \quad (34)$$

2. *Scale columns*: Let $\boldsymbol{\tau} \in \mathbb{R}^n$ be the solution to

$$\min_{\|\boldsymbol{\tau}\|_2=1} \|(N_R + D_{R,1} + D_{R,2})\boldsymbol{\tau}\|_2^2. \quad (35)$$

Then we perform the update

$$A^{(r+1)} = \frac{\|A^{(r+\frac{1}{2})}\|_F^2}{\|A^{(r+\frac{1}{2})} \text{diag}(\boldsymbol{\tau})\|_F^2} A^{(r+\frac{1}{2})} \text{diag}(\boldsymbol{\tau}). \quad (36)$$

The Sinkhorn scaling procedure is detailed in [Algorithm 2](#).

Algorithm 2. Sinkhorn scaling for quadratic constraint satisfaction

Input: $A \in \mathbb{R}^{2n \times n}$, the output of IRLS-ADMM

Output: $A' \in \mathbb{R}^{2n \times n}$, a pure common lines matrix

```

1: procedure SINKHORN ( $A$ )
2:    $r \leftarrow 0$ 
3:   while not converged do
4:     set  $N_L, D_{L,1}, D_{L,2}$  using (B1), (B7), (B8)
5:     set  $\mu$  to be the solution of (33)
6:     update  $A$  using (34) ▷ 1. Scale rows
7:     set  $N_R, D_{R,1}, D_{R,2}$  using (B2), (B9), (B10)
8:     set  $\tau$  to be the solution of (35)
9:     update  $A$  using (36) ▷ 2. Scale columns
10:  end while
11:   $r \leftarrow r + 1$ 
12:   $A' \leftarrow A$ 
13: end procedure

```

5.3. Rotation recovery

IRLS-ADMM and SINKHORN aim to output a pure common lines matrix $A \in \mathbb{R}^{2n \times n}$. Given a pure common lines matrix, we now show how to determine the underlying rotations $R^{(i)}$ in (3) which generated A (recall Theorem 3.5).

Given A , use singular value decomposition to compute a rank-3 factorization $A = BC^\top$ for $B \in \mathbb{R}^{2n \times 3}$ and $C \in \mathbb{R}^{n \times 3}$. We then seek an invertible matrix $Q \in \mathbb{R}^{3 \times 3}$ so that BQ and $CQ^{-\top}$ take the form of the factors in (6). In particular, it is required that the rows of BQ come in orthonormal pairs; more precisely, $(BQ)(BQ)^\top = BQQ^\top B^\top \in \mathbb{R}^{2n \times 2n}$ should have 2×2 identities along its diagonal. Setting $X = QQ^\top \in \mathbb{R}^{3 \times 3}$ and relaxing positive semidefiniteness, let us consider the following affine-linear least squares problem:

$$\min_{\substack{X \in \mathbb{R}^{3 \times 3} \\ X = X^\top}} \|\mathcal{L}(BXB^\top)\|_F^2, \quad (37)$$

where $\mathcal{L}: \mathbb{R}^{2n \times 2n} \rightarrow \mathbb{R}^{2n \times 2n}$ denotes the affine-linear operator which sets all entries of a $2n \times 2n$ matrix outside of the 2×2 diagonal blocks to 0, and subtracts the 2×2 identity matrices from the diagonal blocks. The normal equations for (37) may be written as

$$L\text{vec}(X) = \text{vec}(B^\top B), \quad (38)$$

where $L \in \mathbb{R}^{9 \times 9}$, whose rows and columns index the variables X_{ij} for $1 \leq i \leq j \leq 3$, and whose corresponding $(ij, k\ell)$ -th entry is

$$(L)_{ij,k\ell} = \langle (I_{n \times n} \otimes \mathbf{1}_{1 \times 2})(\mathbf{b}_i \odot \mathbf{b}_\ell), (I_{n \times n} \otimes \mathbf{1}_{1 \times 2})(\mathbf{b}_j \odot \mathbf{b}_k) \rangle, \quad (39)$$

where $\mathbf{b}_i$ is the i th column of B . Generically there is a unique symmetric matrix X solving (38).

Let $X = VDV^\top$ be an eigendecomposition for the solution to (37), where $V, D \in \mathbb{R}^{9 \times 9}$. In the clean case, the diagonal matrix D is already positive semidefinite. Then $VD^{1/2} \in \mathbb{R}^{3 \times 3}$ is a candidate for Q . Next the i th row block of BQ determines the first two rows of $R^{(i)}$, and the cross product of these two rows computes the third row of $R^{(i)}$. From this, we recover the n -tuple of rotations $(R^{(1)}, \dots, R^{(n)})$ up to global right multiplication by a rotation. The full procedure for rotation recovery is in Algorithm 3, which is formulated to apply to noisy inputs as well. We note that as explained in Remark 3.6, a pure common lines matrix A can only be recovered up to sign, because there are two distinct n -tuples of rotations that can be

recovered corresponding to the chiral ambiguity of common lines data. One can run ROTATIONS twice, on A and $-A$, to produce the two possible sets of rotations.

Algorithm 3. Rotation recovery

Input: $A \in \mathbb{R}^{2n \times n}$, an estimate for pure common lines matrix

Output: $(R_1, \dots, R_n) \in \text{SO}(3)^n$, rotations determining the pure common lines matrix

```

1: procedure ROTATIONS ( $A$ )
2:   set  $B$  using the SVD to get a rank-3 approximation  $BC^\top$  for  $A$ 
3:   set  $L$  using (39)
4:   set  $X$  to be least squares solution of (38)
5:   set  $V, D$  using the eigendecomposition  $X = VDV^\top$ 
6:    $D_{ii} \leftarrow \max(D_{ii}, 0)$ 
7:   for  $i = 1, \dots, n$  do
8:     set  $\mathbf{q}_i^\top$  to be the  $(2i - 2 + j)$ -th row of  $BVD^{1/2}$  for  $j = 1, 2$ 
9:     set  $\mathbf{q}_3^\top$  to be the  $i$ th row of  $C(VD^{1/2})^{-\top}$ 
10:    set  $R_i \in \mathbb{R}^{3 \times 3}$  to be  $\begin{pmatrix} \mathbf{q}_2^\top \\ -\mathbf{q}_1^\top \\ \mathbf{q}_3^\top \end{pmatrix}$ 
11:  end for
12:  if  $\det(R_i) < 0$  then
13:    for  $i = 1, \dots, n$  do
14:       $R_i \leftarrow -R_i$ 
15:    end for
16:  end if
17:  for  $i = 1, \dots, n$  do
18:    replace  $R_i$  by the nearest rotation matrix to  $R_i$  using the SVD of  $R_i$ 
19:  end for
20: end procedure

```

5.4. Justification of algorithms

Suppose the input to IRLS-ADMM is a noiseless common lines matrix $\hat{A} \in \mathbb{R}^{2n \times n}$, and its output $A \in \mathbb{R}^{2n \times n}$ is a global minimizer to the non-convex problem (13). In the next theorem, we show that we can recover the ground-truth pure common lines matrix, up to a global scale, by scaling the 2×1 blocks of A via IRLS-ADMM and enforcing the quadratic constraints via SINKHORN. The theorem justifies using IRLS-ADMM and SINKHORN.

Theorem 5.1. *Let $n \geq 4$. Let $A \in \mathbb{R}^{2n \times n}$ be a generic pure common lines matrix and $\lambda_{ij} \in \mathbb{R}$ be nonzero scales for $i, j = 1, \dots, n$ with $i \neq j$. Let $\Lambda \in \mathbb{R}^{n \times n}$ where $\Lambda_{ij} = \lambda_{ij}$ if $i \neq j$, $\Lambda_{ii} = 0$, and $\Lambda = \Lambda^\top$. Suppose*

$$B := (\Lambda \otimes \mathbf{1}_{2 \times 1}) \odot A = \begin{pmatrix} \mathbf{0} & \lambda_{12} \mathbf{a}_{12} & \dots & \lambda_{1,n-1} \mathbf{a}_{1,n-1} & \lambda_{1,n} \mathbf{a}_{1,n} \\ \lambda_{21} \mathbf{a}_{21} & \mathbf{0} & \dots & \lambda_{2,n-1} \mathbf{a}_{2,n-1} & \lambda_{2,n} \mathbf{a}_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_{n-1,1} \mathbf{a}_{n-1,1} & \lambda_{n-1,2} \mathbf{a}_{n-1,2} & \dots & \mathbf{0} & \lambda_{n-1,n} \mathbf{a}_{n-1,n} \\ \lambda_{n,1} \mathbf{a}_{n,1} & \lambda_{n,2} \mathbf{a}_{n,2} & \dots & \lambda_{n,n-1} \mathbf{a}_{n,n-1} & \mathbf{0} \end{pmatrix}$$

has rank 3. Then there exists $\mu_1, \dots, \mu_n, \tau_1, \dots, \tau_n \in \mathbb{R}$ such that

$$\begin{pmatrix} 0 & \lambda_{12} & \dots & \lambda_{1,n-1} & \lambda_{1,n} \\ \lambda_{21} & 0 & \dots & \lambda_{2,n-1} & \lambda_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_{n-1,1} & \lambda_{n-1,2} & \dots & 0 & \lambda_{n-1,n} \\ \lambda_{n,1} & \lambda_{n,2} & \dots & \lambda_{n,n-1} & 0 \end{pmatrix} = \begin{pmatrix} 0 & \mu_1 \tau_1 & \dots & \mu_1 \tau_{n-1} & \mu_1 \tau_n \\ \mu_2 \tau_1 & 0 & \dots & \mu_2 \tau_{n-1} & \mu_2 \tau_n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_{n-1} \tau_1 & \mu_{n-1} \tau_2 & \dots & 0 & \mu_{n-1} \tau_n \\ \mu_n \tau_1 & \mu_n \tau_2 & \dots & \mu_n \tau_{n-1} & 0 \end{pmatrix}. \quad (40)$$

If B additionally satisfies the quadratic constraints (10), then there exists $\tau \in \mathbb{R}$ such that for all $i, j = 1, \dots, n$ ($i \neq j$) it holds $\lambda_{ij} = \tau$.

Proof. See Appendix A.

Remark 5.2. Theorem 3.5 implies that given a pure common lines matrix, the rotation recovery problem in Section 5.3 is uniquely solvable up to a global rotation. Theorem 5.1 states that the ground-truth pure common lines matrix can only be determined up to a global scale; in particular, τ in Theorem 5.1 may be positive or negative. As in Remark 3.6, this sign flip corresponds to chiral ambiguity in cryo-EM. Apart from its sign, the global scale has no effect on rotation recovery in Algorithm 3.

6. Application: Clustering heterogeneous common lines

Here we present an application of our approach for common lines to a challenging problem in cryo-EM. We propose a clustering algorithm for detecting homogeneous communities of consistent common lines from discretely heterogeneous data, using our algebraic constraints. One can then use the clusters of common lines for rotation recovery and 3D reconstruction.

Several successful methods have been proposed for clustering heterogeneous cryo-EM data that consist of images of a *single* macromolecule with conformational landscapes or differences in subunits.^(21–27) Recent work of the third author⁽⁷⁾ proposed a method of solving a different heterogeneity challenge in cryo-EM, where the heterogeneity in the data comes from *multiple* distinct macromolecules rather than variations on one primary structure. Our proposed application focuses on the latter problem. For this setting of heterogeneity, the main prior work to compare against is (7).

The basic idea of our approach is illustrated in Figure 1. There the matrix in gray is a matrix of common lines from simulated heterogeneous data, corresponding to two distinct molecular conformations. The two homogeneous common lines matrices are diagonal blocks in green and purple. The 2×1 entries outside of these diagonal blocks do not correspond to any consistent lines and are just random lines in $\mathbb{R}^2$, encoded as representatives. Note that in general, a heterogeneous common lines matrix will not necessarily have consistent common lines matrices as diagonal blocks, but will be a $2 \times n$ row and $n \times 1$ column permutation of such a matrix. Gaussian white noise has also been added to the gray matrix to decrease the signal-to-noise ratio of the common lines. A scree plot in Figure 1 shows a noticeable spectral gap between the third and fourth singular values of the homogeneous common lines matrices (green and purple curve), which is not detectable for the entire heterogeneous common lines matrix (gray curve), thus demonstrating low-rank structure of submatrices.

Algorithm 4. Heterogeneous clustering on common line constraints

Input: $A \in \mathbb{R}^{2n \times n}$, a common lines matrix

Output: $C_1, \dots, C_r \subseteq \{1, \dots, n\}$, a partition of all common lines into consistent clusters

1: **procedure** CLUSTERS (A)

2: $L \leftarrow \emptyset$

▷ 1. Generate samples

```

3:  for  $i \geq 1$  until sufficient do
4:    set  $S \subseteq \{1, \dots, n\}$  to be a random sample such that  $|S| = 4$ 
5:    set  $A_S$  to be the submatrix of  $A$  associated to the common lines in  $S$ 
6:     $A_S \leftarrow \text{IRLS-ADMM}(A_S)$ 
7:     $A_S \leftarrow \text{SINKHORN}(A_S)$ 
8:    if converged then
9:      set  $M$  using (19) on  $A_S$ 
10:     set  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$  using (28) on  $A_S$ 
11:      $e \leftarrow \|M - M^\top\|_F^2 + \|\mathbf{v}_1 - \mathbf{v}_2\|_2^2 + \|\mathbf{v}_2 - \mathbf{v}_3\|_2^2$ 
12:      $L \leftarrow \text{append}(L, (S, e))$ 
13:    end if
14:  end for
15:  sort  $L$  by increasing values of  $e$  ▷ 2. Cluster samples
16:   $G \leftarrow \mathbf{0}_{n \times n}$ 
17:  while  $L \neq \emptyset$  do
18:     $(S, e) \leftarrow \text{pop}(L)$ 
19:     $G_{S,S} \leftarrow \max\{G_{S,S}, -\log(e) \cdot \mathbf{1}_{4 \times 4}\}$ 
20:  end while
21:   $G_{ii} \leftarrow 0$  for all  $i = 1, \dots, n$ 
22:   $C_1, \dots, C_r \leftarrow \text{COMMUNITYDETECTION}(G)$  ▷ Use method from (28)
23: end procedure

```

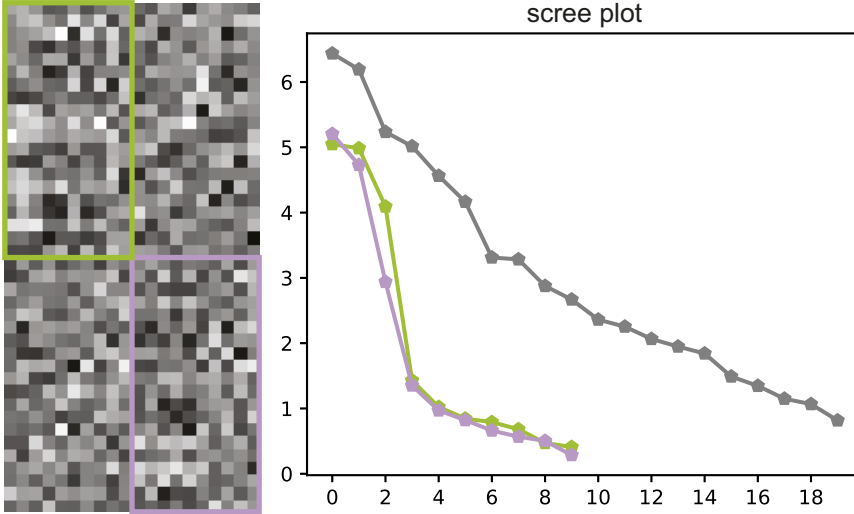


Figure 1. A heterogeneous common lines matrix and rank test for simulated rotations from two distinct molecules. Block-diagonals comparing rotations from the same molecule show rank-3 structure.

Our clustering algorithm consists of two main steps:

1. *Generate samples:* As input, we are given a single common lines matrix A , from which we identify small clusters of consistent common lines. We randomly sample a set of four common lines S and extract the corresponding 8×4 submatrix A_S from A (the choice of four common lines is explained in [Section 7.3](#) based on numerical experiments). We then run IRLS-ADMM and

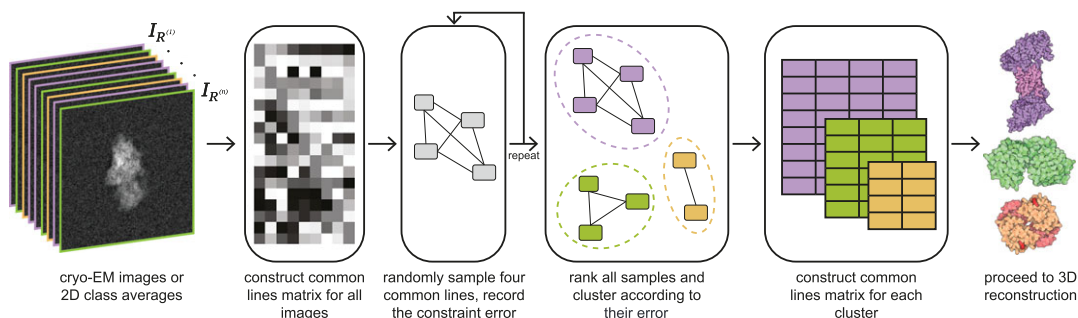


Figure 2. Algorithm for separating images of distinct molecules using algebraic constraints on common lines. The common lines matrix is first computed from an input set of images or class averages. We then apply Algorithm 4, our clustering algorithm. After clustering, images corresponding to the same molecule can then be used for 3D reconstruction.

SINKHORN on A_S . These methods may occasionally diverge due to numerical instability from noise in the data, as discussed in Section 7, in which case we discard the sample. We also discard the sample if the spectral gap between the third and fourth singular value of A_S is not sufficiently large (i.e., A_S does not have numerical rank 3). Otherwise, we obtain a quadratic constraint satisfaction error for the sample. We record both the sample and its error, and repeat this sampling sufficiently many times.

2. *Cluster samples:* We view the collection of samples and errors we obtain as a weighted hypergraph on n vertices whose hyperedges are of size 4 corresponding to the samples and whose hyperedge weights are given by their corresponding errors. We convert the weighted hypergraph into a weighted graph by constructing a weighted adjacency matrix whose (i, j) entry is the negative logarithm of the smallest error on a hyperedge containing both common lines i and j . We then use an unsupervised community detection algorithm on this adjacency matrix to find the clusters of consistent common lines. In our numerical experiments, we use the algorithm proposed by Lancichinetti, Fortunato, and Kertész,⁽²⁸⁾ which can identify overlapping communities and hierarchical structure, and depends only on a single hyperparameter controlling the scale of the hierarchies. In particular, we do not need to specify the number of clusters or their sizes.

The clustering algorithm is detailed in Algorithm 4 and illustrated in Figure 2.

7. Performance on data

We compare the performance of our methods to existing common lines based algorithms in the literature, namely functions in the software package ASPIRE⁽²⁹⁾ and the clustering algorithm of Verbeke et al.⁽⁷⁾ We study the problems of recovering rotations, denoising common lines, and partitioning discretely heterogeneous image sets into homogeneous subcommunities using common lines. The tests are done on simulated data (at various levels of noise) and real data.

Our simulated data consists of image data of the 40S, 60S, and 80S ribosome (available from the Electron Microscopy Data Bank⁽³⁰⁾ as entries EMD-4214, EMD-2811, and EMD-2858, respectively), generated by ASPIRE. The ribosomes and examples of their clean 2D projection images are displayed in Figure 3. Each image is 128×128 with a pixel size of 3 Å. White Gaussian noise is added to each image corresponding to a specified signal-to-noise ratio (SNR). We define the SNR by taking the signal to be the average squared intensity over each pixel in the clean image and setting the noise variance to achieve the appropriate ratio. The common lines between two images are detected by finding the line projections of the two images with the highest correlation, as computed in ASPIRE.

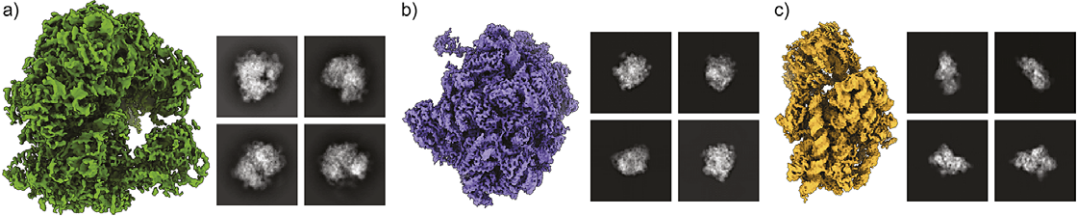


Figure 3. 3-D structures and example projection images for the three structures used for simulation. (a) 80S ribosome (EMD-2858) and example projection images. (b) 60S ribosome (EMD-2811) and example projection images. (c) 40S ribosome (EMD-4214) and example projection images.

In our numerical experiments, we have observed that the performance of IRLS-ADMM, and consequently SINKHORN, can depend on its initialization. In particular, we have occasionally observed divergence or vanishing of the entries of A in SINKHORN. This behavior appears to be due to numerical instabilities in SINKHORN arising from noise in the data or from using too many common lines. In these cases, we can either discard such runs and restart the algorithms with new initializations, or skip using SINKHORN. We address this issue in each of our tests.

7.1. Rotation recovery

Let $R^{(1)}, \dots, R^{(n)} \in \text{SO}(3)$ be the ground-truth rotations and $R_1, \dots, R_n \in \text{SO}(3)$ be the recovered rotations. Then Theorem 3.5 states that there exists a unique rotation $Q \in \text{SO}(3)$ such that $R^{(i)} = R_i Q$ for all $1 \leq i \leq n$. In other words, Q can be found by solving the orthogonal Procrustes problem

$$\min_{Q \in \text{SO}(3)} \frac{1}{n} \left\| \begin{pmatrix} R^{(1)} \\ \vdots \\ R^{(n)} \end{pmatrix} - \begin{pmatrix} R_1 \\ \vdots \\ R_n \end{pmatrix} Q \right\|_F^2. \quad (41)$$

The solution to this problem can be found using SVD.⁽³¹⁾ We note that for $n=1$, there is a simple relation between the Procrustes error and the angular error between two rotation matrices. For $R, S \in \text{SO}(3)$, it holds $\|R - S\|_F^2 = \langle R - S, R - S \rangle = \|R^\top R\|_F^2 - 2\langle R, S \rangle + \|S^\top S\|_F^2 = \|I_{3 \times 3}\|_F^2 - 2\langle R, S \rangle + \|I_{3 \times 3}\|_F^2 = 2 \cdot 3 - 2 \cdot \text{tr}(R^\top S)$. Hence the angular error between R and S is

$$\theta = \arccos\left(\frac{\text{tr}(R^\top S) - 1}{2}\right) = \arccos\left(\frac{3 - \frac{1}{2}\|R - S\|_F^2 - 1}{2}\right).$$

We compare our method to the procedure in ASPIRE on simulated data. Given common lines, the rotation recovery algorithm used in ASPIRE is the synchronization with voting procedure as described in (5). We use 30 images of the 60S, 80S, and 40S ribosomes at SNR = 0.125, 0.25, 0.5, 1. For each macromolecule and SNR, we generate 50 sets of 30 random ground-truth rotations and their corresponding noisy images. We report the average Procrustes error (41) per image (i.e., the Procrustes error divided by the number of images).

When running this test, we chose to only use our IRLS-ADMM algorithm followed by Rotations, and omitted the SINKHORN row and column scaling step due to observed numerical instabilities of SINKHORN with noise in the data or too many common lines. Also, Remark 5.2 states that our algorithm is only guaranteed to recover the ground-truth pure common lines matrix up to a global scale, and the sign of this scale produces two sets of rotations that differ by left-multiplication with $J = \text{diag}(-1, -1, 1)$. This sign flip corresponds to the chiral ambiguity in cryo-EM, as explained in Remark 3.6. Thus in our tests, we report the best rotational error amongst the two possible sets of rotations.

The results for rotation recovery are displayed in Table 1. Notably, at lower SNR our method is consistently more accurate than ASPIRE.

Table 1. The average rotation recovery error from 30 simulated images of macromolecules at various SNR, 50 runs each

Macromolecule	SNR	Average Procrustes rotation error (42)	
		Algorithm 1 and Algorithm 3	ASPIRE
EMD–2811 (60S ribosome)	0.125	3.6840	4.3426
	0.25	3.3215	3.6042
	0.5	1.5512	2.2704
	1	0.4215	0.4877
EMD–2858 (80S ribosome)	0.125	1.7903	2.5943
	0.25	0.8038	1.3175
	0.5	0.1576	0.2543
	1	0.0259	0.0240
EMD–4214 (40S ribosome)	0.125	4.0377	4.5281
	0.25	3.9079	4.0104
	0.5	3.6375	3.3994
	1	2.8271	2.1334

Bold values indicate the algorithm with lower error.

7.2. Denoising common lines

The problem we consider here is the following: given a *noisy* common lines matrix, how well can we recover the ground-truth clean pure common lines matrix? If $A, B \in \mathbb{R}^{2n \times n}$ are the ground truth and recovered pure common lines matrix respectively, then we measure this error to be

$$\min_{\lambda \in \mathbb{R}} \frac{1}{n} \|A - \lambda B\|_F^2 \quad (42)$$

since there is a global scale ambiguity in the recovered pure common lines matrix as discussed in Remark 5.2. The above problem is a least-squares problem in λ and hence has a closed-form solution.

We compare our methods to ASPIRE on simulated data as follows. We run the rotation recovery algorithm of ASPIRE based on common lines to obtain rotations $R^{(1)}, \dots, R^{(n)} \in SO(3)$. Then we construct a recovered pure common lines matrix B from these rotations by using the factorization (6). We then compare the denoising error (42) from the ground-truth clean common lines matrix A to the output of our IRLS-ADMM method and to the matrix B . As before, the simulated data consists of 30 images of the 60S, 80S, and 40S ribosomes at SNR = 0.125, 0.25, 0.5, 1, and we report the average denoising error (42) per image over 50 runs. We assume we have the ground-truth chiral information when recovering rotations with ASPIRE.

The results for common line denoising are displayed in Table 2. Again our IRLS-ADMM method outperforms ASPIRE for denoising common lines matrices, particularly at low SNR.

7.3. Clustering heterogeneous image sets

We test the performance of our algorithm CLUSTERS for clustering (see Section 6) on simulated and real data.

The success of clustering is measured by the adjusted Rand index⁽³²⁾ (ARI) between the ground-truth clusters and the recovered clusters. The range of this index is $-\infty < \text{ARI} \leq 1$, with $\text{ARI} = 1$ if the two partitions are identical. The ARI is a corrected-for-chance version of the Rand index, meaning that it is the expected Rand index for the cluster and is equal to 0 if every element is placed in a random cluster.

While any number of common lines $|S|$ can be sampled on line 4 of CLUSTERS, we found that sampling four common lines at a time was effective for a number of reasons: $|S| = 4$ enforces a non-trivial rank 3 constraint, and the small number of common lines allowed us to both generate many samples rapidly and

Table 2. The average denoising error of recovered pure common lines matrices from 30 simulated images of macromolecules at various SNR, 50 runs each

Macromolecule	SNR	Average denoising error (42)	
		Algorithm 1	ASPIRE
EMD-2811 (60S ribosome)	0.125	14.1982	18.2274
	0.25	12.7786	17.8805
	0.5	8.2349	16.8199
	1	4.1908	6.1365
EMD-2858 (80S ribosome)	0.125	11.2948	17.3292
	0.25	6.8850	13.3778
	0.5	2.3152	3.6690
	1	0.5680	0.3512
EMD-4214 (40S ribosome)	0.125	15.5635	18.0612
	0.25	15.1112	18.0862
	0.5	14.0420	18.0069
	1	11.6964	15.7237

Bold values indicate the algorithm with lower error.

improve the numerical stability of the SINKHORN scaling procedure. In addition, the COMMUNITYDETECTION algorithm we use is the one described in (28).

7.3.1. Simulated data

We generate a dataset containing three clusters, with $n = 5 + 30 + 15 = 50$ images from the 40S, 60S, and 80S ribosomes, respectively, from which we construct a common lines matrix.

CLUSTERS achieved perfect clustering ($\text{ARI} = 1$) at $\text{SNR} = 10$, $\text{ARI} = 0.8581$ at $\text{SNR} = 5$, and $\text{ARI} = 0.3286$ at $\text{SNR} = 1$. The clusters found at $\text{SNR} = 5$ are displayed in Figure 4, which shows that only one pair of images were placed in incorrect clusters.

7.3.2. Real data

Our real data consists of a subset of 2D class averages computed from the experimental data described in Verbeke et al.⁽⁷⁾ The subset we consider consists of two clusters with $n = 47 + 28 = 75$ images

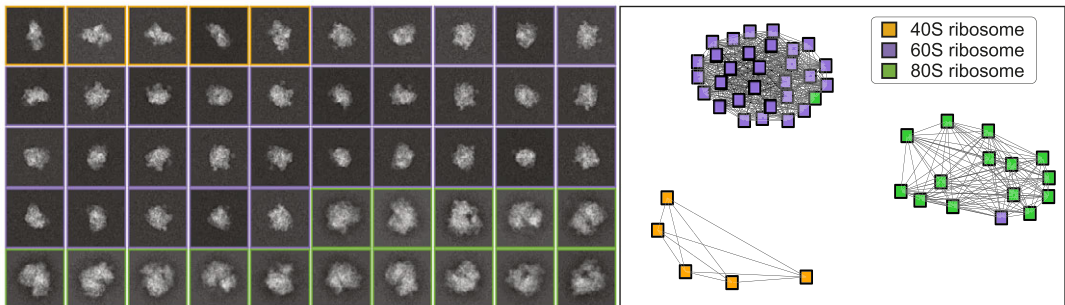


Figure 4. Clustering results for $n = 50$ simulated images with $\text{SNR} = 5$. Images are size 128×128 with pixel size of $3 \text{ \AA}$, and are colored according to the ground truth labels. Using CLUSTERS achieves $\text{ARI} = 0.8581$ and only one pair of images are incorrectly clustered.

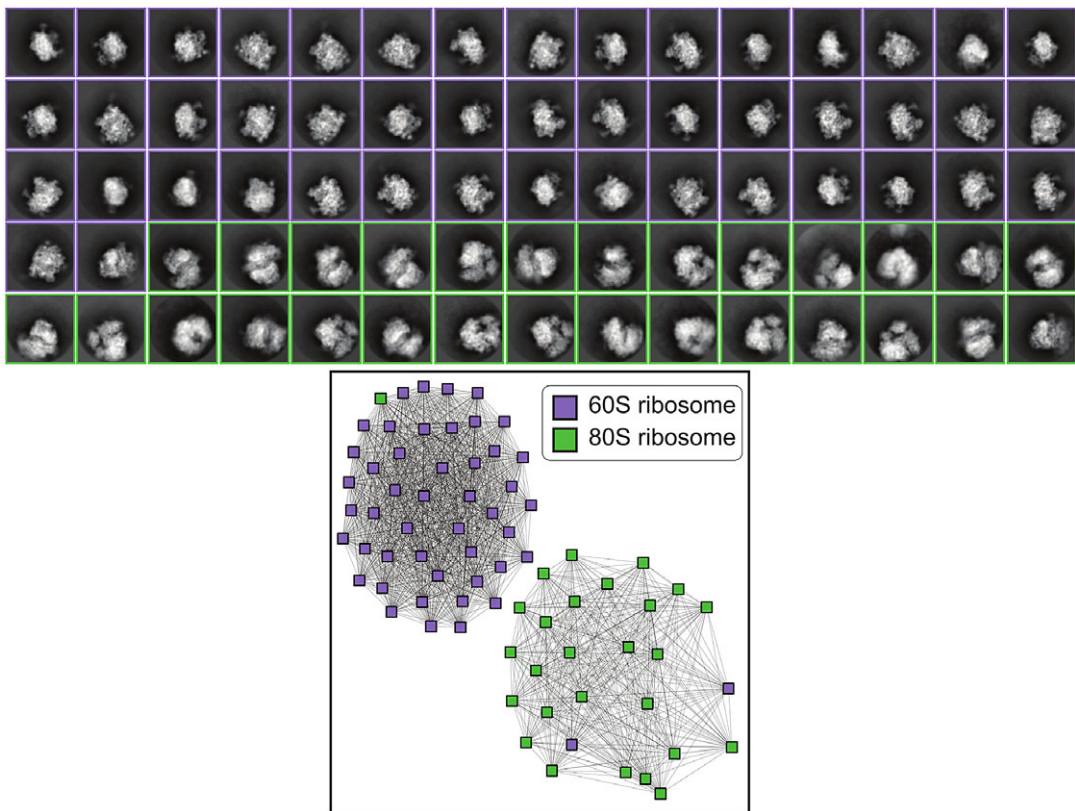


Figure 5. Clustering results for $n = 75$ 2D class averages from EMPIAR-10268 computed as described in (7). Images are size 96×96 with a pixel size of $4.4 \text{ \AA}$, and are colored according to the ground truth labels. Using *CLUSTERS* achieves $\text{ARI} = 0.8440$ and only three images are incorrectly clustered.

corresponding to the 60S and 80S ribosomes respectively. Each class average is 96×96 with a pixel size of $4.4 \text{ \AA}$. We use the labels from (7) as ground truth for clustering.

Figure 5 shows the clusters found by our algorithm *CLUSTERS*, achieving $\text{ARI} = 0.8440$ and misclassifying only three images.

The clustering algorithm used in (7) is based on performing community detection on a nearest-neighbors graph constructed using *Euclidean distances* between the best-matching line projections between every pair of images. We stress that our clustering algorithm uses a completely distinct aspect of common lines data: the *positions* of the common lines. As a proof of concept for our constraints, we do not make use of the correlations between the common lines at all, unlike (7). The test for *CLUSTERS* only uses a subset of the dataset in (7), which has $n = 100$ images and includes images with unknown labels. If we compare only the 60S and 80S images that were clustered, then *CLUSTERS* achieves a similar performance, where one additional image is misclassified by *CLUSTERS* compared to (7).

8. Conclusion

This article revisited the fundamental topic of common lines in cryo-EM image processing. We discussed a novel approach for dealing with common lines, based on a certain $2n \times n$ matrix encoding the common lines between n projection images. We proved that if the 2×1 blocks of the matrix are properly scaled, then the matrix satisfies nice algebraic constraints: a low-rank condition and several sparse quadratic constraints. The new formulation operates directly on common lines data, and is fully global in that it does

not require angular reconstitution or voting procedures at all. It opens the door to different and potentially more robust approaches to computational tasks involving common lines. Using the algebraic constraints, we adapted optimization algorithms from other domains to give new methods to denoise common lines data, and recover the 3D rotations underlying noisy images. Numerical experiments show that these methods have increased accuracy at low SNR, compared to existing methods based on common lines. We also explored a setting where traditional common lines methods fail to apply – cryo-EM datasets with discrete heterogeneity – by proposing a sampling-based process to cluster the images in homogeneous subcommunities based on our algebraic constraints. Experiments with simulated and real data show the method performs well when applied to images with high noise.

Although there is clear promise, several future directions could be pursued for further improvements. Firstly, in [Section 5](#) the optimization algorithm building on (11) is quite complex. Matrix scaling problems as in (11) and our work are an interesting variation on the problem of matrix completion; would other optimization approaches perform better? Secondly, extensions to molecules with nontrivial point group symmetries would be useful (and currently are a focus in other common lines research). Perhaps our formulation can suggest another way to incorporate symmetries into common lines methods. Lastly, in the application to discrete heterogeneity, we neglected correlation scores between the common lines, on which (7) relied. It is likely better to use both the scores and the algebraic constraint errors.

Data availability statement. Data and replication code are available at <https://github.com/ozitommi/algebraic-common-lines>.

Author contribution. All authors conceived the project, designed the algorithms, and developed the mathematical theory. T.M., A.L.D., and E.J.V. prepared the data. T.M. and A.L.D. wrote the software and performed the numerical experiments. All authors wrote the manuscript and approved its final submission.

Funding statement. T.M. is supported by a Mathematical Institute Scholarship at the University of Oxford. A.L.D. is supported in part by NSF DMS 1937215. E.J.V. is supported by the Simons Foundation Math+X Investigator award to Amit Singer. J.K. is supported in part by NSF DMS 2309782, NSF CISE-IIS 2312746, and start-up grants from the College of Natural Science and Oden Institute at the University of Texas at Austin.

Competing interest. The authors declare no competing interests exist.

References

1. Vainshtein BK and Goncharov AB (1986) Determination of the spatial orientation of arbitrarily arranged identical particles of unknown structure from their projections. *Soviet Physics Doklady* **31**, 278.
2. Van Heel M (1987) Angular reconstitution: A posteriori assignment of projection directions for 3D reconstruction. *Ultra-microscopy* **21**, 111–123.
3. Singer A, Coifman RR, Sigworth FJ, Chester DW and Shkolnisky Y (2010) Detecting consistent common lines in cryo-EM by voting. *Journal of Structural Biology* **169**, 312–322.
4. Singer A and Shkolnisky Y (2011) Three-dimensional structure determination from common lines in cryo-EM by eigenvectors and semidefinite programming. *SIAM Journal on Imaging Sciences* **4**, 543–572.
5. Shkolnisky Y and Singer A (2012) Viewing direction estimation in cryo-EM using synchronization. *SIAM Journal on Imaging Sciences* **5**, 1088–1110.
6. Dynerman D (2014) Semi-algebraic geometry of common lines. *Research in the Mathematical Sciences* **1**, 1–20.
7. Verbeke EJ, Zhou Y, Horton AP, Mallam AL, Taylor DW and Marcotte EM (2020) Separating distinct structures of multiple macromolecular assemblies from cryo-EM projections. *Journal of Structural Biology* **209**, 107416.
8. Verbeke EJ, Mallam AL, Drew K, Marcotte EM and Taylor DW (2018) Classification of single particles from human cell extract reveals distinct structures. *Cell Reports* **24**, 259, e3–268.
9. Yi X, Verbeke EJ, Chang Y, Dickinson DJ and Taylor DW (2019) Electron microscopy snapshots of single particles from single cells. *Journal of Biological Chemistry* **294**, 1602–1608.
10. Sae-Lee W, McCafferty CL, Verbeke EJ, Havugimana PC, Papoulas O, McWhite CD, Houser JR, Vanuytsel K, Murphy GJ, Drew K, Emili A, Taylor DW and Marcotte EM (2022) The protein organization of a red blood cell. *Cell Reports* **40**, 111103.
11. Sengupta S, Amir T, Galun M, Goldstein T, Jacobs DW, Singer A and Basri R (2017) A new rank constraint on multi-view fundamental matrices, and its application to camera location recovery. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, pp. 2413–2421.
12. Frank J (2006) *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in their Native State*. Oxford: Oxford University Press.

13. Geva AS and Shkolnisky Y (2023) A common lines approach for ab initio modeling of molecules with tetrahedral and octahedral symmetry. *SIAM Journal on Imaging Sciences* **16**, 1978–2014.
14. Harris J (1992) *Algebraic Geometry: A First Course*, vol. 133. Cham: Springer Science & Business Media.
15. Bochnak J, Coste M and Roy M-F (2013) *Real Algebraic Geometry*, Vol. **36**. Cham: Springer Science & Business Media.
16. Grayson DR and Stillman ME. Macaulay2, a software system for research in algebraic geometry. Available at <http://www2.macaulay2.com>.
17. Holland PW and Welsch RE (1977) Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods* **6**, 813–827.
18. Boyd S, Parikh N, Chu E, Peleato B and Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* **3**, 1–122.
19. Goldstein T, O'Donoghue B, Setzer S and Baraniuk R (2014) Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences* **7**, 1588–1623.
20. Sinkhorn R and Knopp P (1967) Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics* **21**, 343–348.
21. Aizenbud Y and Shkolnisky Y (2019) A max-cut approach to heterogeneity in cryo-electron microscopy. *Journal of Mathematical Analysis and Applications* **479**, 1004–1029.
22. Herman G and Kalinowski M (2008) Classification of heterogeneous electron microscopic projections into homogeneous subsets. *Ultramicroscopy* **108**, 327–338.
23. Shatsky M, Hall RJ, Nogales E, Malik J and Brenner SE (2010) Automated multi-model reconstruction from single-particle electron microscopy data. *Journal of Structural Biology* **170**, 98–108.
24. Penczek PA, Frank J and Spahn CM (2006) A method of focused classification, based on the bootstrap 3D variance analysis, and its application to EF-G-dependent translocation. *Journal of Structural Biology* **154**, 184–194.
25. Liao HY, Hashem Y and Frank J (2015) Efficient estimation of three-dimensional covariance and its application in the analysis of heterogeneous samples in cryo-electron microscopy. *Structure* **23**, 1129–1137.
26. Scheres SH (2012) RELION: implementation of a Bayesian approach to cryo-EM structure determination. *Journal of Structural Biology* **180**, 519–530.
27. Katsevich E, Katsevich A and Singer A (2015) Covariance matrix estimation for the cryo-EM heterogeneity problem. *SIAM Journal on Imaging Sciences* **8**, 126–185.
28. Lancichinetti A, Fortunato S and Kertész J (2009) Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* **11**, 033015.
29. Wright G, Andén J, Bansal V, Xia J, Langfield C, Carmichael J, Sowattanangkul K, Brook R, Shi Y, Heimowitz A, Pragier G, Sason I, Moscovich A, Shkolnisky Y and Singer A (2023) ComputationalCryoEM/ASPIRE-Python: v0.12.0 <https://doi.org/10.5281/zenodo.5657281>.
30. Lawson CL, Patwardhan A, Baker ML, Hryc C, Garcia ES, Hudson BP, Lagerstedt I, Ludtke SJ, Pintilie G, Sala R, Westbrook JD, Berman HM, Kleywegt GJ and Chiu W (2015) EMDDataBank unified data resource for 3DEM. *Nucleic Acids Research* **44**, D396–D403.
31. Schönemann PH (1966) A generalized solution of the orthogonal procrustes problem. *Psychometrika* **31**, 1–10.
32. Hubert L and Arabie P (1985) Comparing partitions. *Journal of Classification* **2**, 193–218.
33. Breiding P and Timme S (2018) HomotopyContinuation.jl: A package for homotopy continuation in Julia. In *Mathematical Software – ICMS 2018*. Cham: Springer, pp. 458–465.
34. Levin E, Kileel J and Boumal N (2023) Finding stationary points on bounded-rank matrices: A geometric hurdle and a smooth remedy. *Mathematical Programming* **199**, 831–864.

A. Appendix. Additional proofs

Proof of Proposition 3.3. Since orthogonal matrices preserve norms, we have

$$\begin{aligned}
 1 = \|\mathbf{r}_3^{(i)}\|_2^2 &= \left\| R^{(j)} \mathbf{r}_3^{(i)} \right\|_2^2 = \left\| \begin{pmatrix} 1 & & \\ & -1 & \\ & & 1 \end{pmatrix} \begin{pmatrix} & & 1 \\ & 1 & \\ 1 & & \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^{(j)\top} \\ \mathbf{r}_2^{(j)\top} \\ \mathbf{r}_3^{(j)\top} \end{pmatrix} \mathbf{r}_3^{(i)} \right\|_2^2 = \left\| \begin{pmatrix} \mathbf{r}_3^{(j)\top} \\ -\mathbf{r}_2^{(j)\top} \\ \mathbf{r}_1^{(j)\top} \end{pmatrix} \mathbf{r}_3^{(i)} \right\|_2^2 \\
 &= \langle \mathbf{r}_3^{(j)}, \mathbf{r}_3^{(i)} \rangle^2 + \left\| \begin{pmatrix} -\langle \mathbf{r}_2^{(j)}, \mathbf{r}_3^{(i)} \rangle \\ \langle \mathbf{r}_1^{(j)}, \mathbf{r}_3^{(i)} \rangle \end{pmatrix} \right\|_2^2 = \langle \mathbf{r}_3^{(j)}, \mathbf{r}_3^{(i)} \rangle^2 + \|\mathbf{a}_{ji}\|_2^2.
 \end{aligned} \tag{A1}$$

Similarly, we have

$$1 = \langle \mathbf{r}_3^{(i)}, \mathbf{r}_3^{(j)} \rangle^2 + \|\mathbf{a}_{ij}\|_2^2. \quad (\text{A2})$$

Thus, by equating (A1) with (A2) and simplifying, we obtain $\|\mathbf{a}_{ij}\|_2^2 = \|\mathbf{a}_{ji}\|_2^2$.

Proof of Proposition 3.4. Let $D = \det \begin{pmatrix} \mathbf{r}_3^{(i)} & \mathbf{r}_3^{(j)} & \mathbf{r}_3^{(k)} \end{pmatrix}$. Then for each i we have

$$\begin{aligned} D &= \det \left(\begin{pmatrix} 1 & & \\ & -1 & \\ & & 1 \end{pmatrix} \begin{pmatrix} & & 1 \\ 1 & & \\ & 1 & \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^{(i)\top} \\ \mathbf{r}_2^{(i)\top} \\ \mathbf{r}_3^{(i)\top} \end{pmatrix} \begin{pmatrix} \mathbf{r}_3^{(i)} & \mathbf{r}_3^{(j)} & \mathbf{r}_3^{(k)} \end{pmatrix} \right) = \det \left(\begin{pmatrix} \mathbf{r}_3^{(i)\top} \\ -\mathbf{r}_2^{(i)\top} \\ \mathbf{r}_1^{(i)\top} \end{pmatrix} \begin{pmatrix} \mathbf{r}_3^{(i)} & \mathbf{r}_3^{(j)} & \mathbf{r}_3^{(k)} \end{pmatrix} \right) \\ &= \det \begin{pmatrix} 1 & \langle \mathbf{r}_3^{(i)}, \mathbf{r}_3^{(j)} \rangle & \langle \mathbf{r}_3^{(i)}, \mathbf{r}_3^{(k)} \rangle \\ 0 & -\langle \mathbf{r}_2^{(i)}, \mathbf{r}_3^{(j)} \rangle & -\langle \mathbf{r}_2^{(i)}, \mathbf{r}_3^{(k)} \rangle \\ 0 & \langle \mathbf{r}_1^{(i)}, \mathbf{r}_3^{(j)} \rangle & \langle \mathbf{r}_1^{(i)}, \mathbf{r}_3^{(k)} \rangle \end{pmatrix} = \det \begin{pmatrix} -\langle \mathbf{r}_2^{(i)}, \mathbf{r}_3^{(j)} \rangle & -\langle \mathbf{r}_2^{(i)}, \mathbf{r}_3^{(k)} \rangle \\ \langle \mathbf{r}_1^{(i)}, \mathbf{r}_3^{(j)} \rangle & \langle \mathbf{r}_1^{(i)}, \mathbf{r}_3^{(k)} \rangle \end{pmatrix} = \det(\mathbf{a}_{ij} \quad \mathbf{a}_{ik}), \end{aligned} \quad (\text{A3})$$

by expanding the determinant along the first column. Similarly for each j and k we have

$$D = \det \begin{pmatrix} \langle \mathbf{r}_3^{(j)}, \mathbf{r}_3^{(i)} \rangle & 1 & \langle \mathbf{r}_3^{(j)}, \mathbf{r}_3^{(k)} \rangle \\ -\langle \mathbf{r}_2^{(j)}, \mathbf{r}_3^{(i)} \rangle & 0 & -\langle \mathbf{r}_2^{(j)}, \mathbf{r}_3^{(k)} \rangle \\ \langle \mathbf{r}_1^{(j)}, \mathbf{r}_3^{(i)} \rangle & 0 & \langle \mathbf{r}_1^{(j)}, \mathbf{r}_3^{(k)} \rangle \end{pmatrix} = -\det \begin{pmatrix} -\langle \mathbf{r}_2^{(j)}, \mathbf{r}_3^{(i)} \rangle & -\langle \mathbf{r}_2^{(j)}, \mathbf{r}_3^{(k)} \rangle \\ \langle \mathbf{r}_1^{(j)}, \mathbf{r}_3^{(i)} \rangle & \langle \mathbf{r}_1^{(j)}, \mathbf{r}_3^{(k)} \rangle \end{pmatrix} = -\det(\mathbf{a}_{ji} \quad \mathbf{a}_{jk}). \quad (\text{A4})$$

$$D = \det \begin{pmatrix} \langle \mathbf{r}_3^{(k)}, \mathbf{r}_3^{(i)} \rangle & \langle \mathbf{r}_3^{(k)}, \mathbf{r}_3^{(j)} \rangle & 1 \\ -\langle \mathbf{r}_2^{(k)}, \mathbf{r}_3^{(i)} \rangle & -\langle \mathbf{r}_2^{(k)}, \mathbf{r}_3^{(j)} \rangle & 0 \\ \langle \mathbf{r}_1^{(k)}, \mathbf{r}_3^{(i)} \rangle & \langle \mathbf{r}_1^{(k)}, \mathbf{r}_3^{(j)} \rangle & 0 \end{pmatrix} = \det \begin{pmatrix} -\langle \mathbf{r}_2^{(k)}, \mathbf{r}_3^{(i)} \rangle & -\langle \mathbf{r}_2^{(k)}, \mathbf{r}_3^{(j)} \rangle \\ \langle \mathbf{r}_1^{(k)}, \mathbf{r}_3^{(i)} \rangle & \langle \mathbf{r}_1^{(k)}, \mathbf{r}_3^{(j)} \rangle \end{pmatrix} = \det(\mathbf{a}_{ki} \quad \mathbf{a}_{kj}). \quad (\text{A5})$$

Thus, equating (A3)–(A5), we obtain $\det(\mathbf{a}_{ij} \quad \mathbf{a}_{ik}) = -\det(\mathbf{a}_{ji} \quad \mathbf{a}_{jk}) = \det(\mathbf{a}_{ki} \quad \mathbf{a}_{kj})$.

Proof of Theorem 3.5. We first prove the theorem in the case $n=3$. Given $R_1, R_2, R_3 \in SO(3)$, we have $A = \psi(R_1, R_2, R_3) = \psi(R_1 Q, R_2 Q, R_3 Q)$ for any $Q \in SO(3)$ since

$$\begin{pmatrix} -\mathbf{r}_2^{(1)\top} \\ \mathbf{r}_1^{(1)\top} \\ \vdots \\ -\mathbf{r}_2^{(3)\top} \\ \mathbf{r}_1^{(3)\top} \end{pmatrix}_{6 \times 3} \begin{pmatrix} \mathbf{r}_3^{(1)} & \mathbf{r}_3^{(2)} & \mathbf{r}_3^{(3)} \end{pmatrix}_{3 \times n} = \begin{pmatrix} \begin{pmatrix} -\mathbf{r}_2^{(1)\top} \\ \mathbf{r}_1^{(1)\top} \\ \vdots \\ -\mathbf{r}_2^{(3)\top} \\ \mathbf{r}_1^{(3)\top} \end{pmatrix} Q \\ Q^\top \mathbf{r}_3^{(1)} \quad Q^\top \mathbf{r}_3^{(2)} \quad Q^\top \mathbf{r}_3^{(3)} \end{pmatrix}_{6 \times 3}.$$

Hence, the fibers of ψ contain at least a copy of $SO(3)$. By fixing $Q = R_1^\top$, we may assume that $R_1 = I$, so that $A = \psi(R_1, R_2, R_3) = \psi(I, R_2 R_1^\top, R_3 R_1^\top)$. To prove the statement, we therefore need to show that fibers of the map ψ with $R_1 = I$ fixed,

$$\begin{aligned} \overline{\psi} : SO(3) \times SO(3) &\rightarrow \mathbb{R}^{6 \times 3} \\ (R^{(1)}, R^{(2)}) &\mapsto \psi(I, R^{(1)}, R^{(2)}), \end{aligned} \quad (\text{A6})$$

generically consist of only one point. Our strategy to show this is to set up a corresponding system of polynomial equations for a random instance of two rotations $R^{(1)}$ and $R^{(2)}$, and then solve the system numerically, showing that it has only one solution. It is sufficient to solve the polynomial system over the complex numbers $\mathbb{C}$, and exhibit that there is a unique solution even over $\mathbb{C}$. We parameterize $R_1, R_2 \in SO(3)$ using the Euler-Rodriguez formula:

$$R_i = \begin{pmatrix} a_i^2 + b_i^2 - c_i^2 - d_i^2 & 2(b_i c_i - a_i d_i) & 2(b_i d_i + a_i c_i) \\ 2(b_i c_i + a_i d_i) & a_i^2 + c_i^2 - b_i^2 - d_i^2 & 2(c_i d_i - a_i b_i) \\ 2(b_i d_i - a_i c_i) & 2(c_i d_i + a_i b_i) & a_i^2 + d_i^2 - b_i^2 - c_i^2 \end{pmatrix},$$

where $a_i, b_i, c_i, d_i \in \mathbb{R}$ such that $a_i^2 + b_i^2 + c_i^2 + d_i^2 = 1$. We construct one pure common lines matrix $A = \psi(I, R_1, R_2)$ whose entries are polynomial in a_i, b_i, c_i, d_i , and another pure common lines matrix $\bar{A} = \psi(I, \bar{R}_1, \bar{R}_2)$, from random rotation matrices $\bar{R}_1, \bar{R}_2 \in SO(3)$, whose entries are real. Setting $A = \bar{A}$ gives us a system of polynomial equations, which we solve over $\mathbb{C}$ using the package HomotopyContinuation.jl in Julia.⁽³³⁾ Every rotation under the Euler-Rodriguez parametrization has two parametrizations, namely (a, b, c, d) and $(-a, -b, -c, -d)$ representing the same matrix in $SO(3)$. By parameterizing R_1 and R_2 , we therefore expect to find 4 real solutions to the system and therefore one point in the fiber, which is indeed what we find using homotopy continuation. The computation is carried out in the file fiber.jl in our GitHub repository (8.1).

Now with the theorem true for $n=3$, it in fact follows that the theorem is true for all $n > 3$: if $A_1 = \psi(R_1, \dots, R_n)$ and $A_2 = \psi(S_1, \dots, S_n)$, where $A_1 = A_2$ is a generic point in the image of ψ , then we need to show that there exists a unique $Q \in SO(3)$ such that $S_i = R_i Q$ for all $i = 1, \dots, n$. Choosing two 6×3 pure common lines submatrices of $A_1 = A_2$ corresponding to indices $\{i, j, k\}$ and $\{i, \ell, m\}$, the theorem in the case $n=3$ implies that there exist $Q_{ijk}, Q_{i\ell m} \in SO(3)$ such that $(S_i, S_j, S_k) = (R_i Q_{ijk}, R_j Q_{ijk}, R_k Q_{ijk})$ and $(S_i, S_j, S_\ell) = (R_i Q_{i\ell m}, R_\ell Q_{i\ell m}, R_m Q_{i\ell m})$. In particular, $R_i Q_{ijk} = R_i Q_{i\ell m}$, so $Q_{ijk} = Q_{i\ell m}$. Thus, any triplet of rotations are related by the same matrix Q .

Proof of Proposition 3.7. Let $\mathcal{V}_n \subseteq \mathbb{R}^{2n \times n}$ denote the cone over the common lines variety (i.e., $\mathcal{V}_n$ is the smallest algebraic variety containing all scalar multiples of all pure common lines matrices). Let $\mathcal{W}_n \subseteq \mathbb{R}^{2n \times n}$ denote the variety defined by the polynomial constraints in Proposition 3.7. Note that since the constraints are invariant to global scaling, we have $\mathcal{V}_n \subseteq \mathcal{W}_n$. By (15), in order to show that $\mathcal{V}_n$ is an irreducible component of $\mathcal{W}_n$ we need to show that

$$\dim(\mathcal{V}_n) = \dim(\mathcal{W}_n, A) \quad (\text{A7})$$

holds a generic point $A \in \mathcal{V}_n$, where $\dim(\cdot)$ and $\dim(\cdot, \cdot)$ respectively denote the dimension of a variety and the local dimension of a variety at a point. This is analogous to computing the dimension of a manifold by computing the dimension of its tangent space at a point.

Recall $\mathcal{V}_n$ is the cone over the common lines variety, which is the smallest algebraic variety containing the image of the map ψ in (7). Thus using the fiber dimension theorem (the algebraic geometric analog of the rank-nullity theorem from linear algebra), Theorem 3.5 implies

$$\dim(\mathcal{V}_n) = 3n - 2. \quad (\text{A8})$$

Next, we compute $\dim(\mathcal{W}_n, A)$. We first note that $\mathcal{W}_n \subseteq \mathcal{X}_n$, where $\mathcal{X}_n$ is the algebraic variety of all rank ≤ 3 matrices in $\mathbb{R}^{2n \times n}$ defined by the vanishing of 4×4 minors. Note the map $\rho: \mathbb{R}^{2n \times 3} \times \mathbb{R}^{n \times 3} \rightarrow \mathcal{V}_n$ given by $\rho(B, C) = BC^T$ parameterizes $\mathcal{V}_n$ with 9-dimensional fibers over each rank-3 point.⁽³⁴⁾ This holds because for each 3×3 invertible matrix M , $\rho(B, C) = \rho(BM^{-1}, CM^T)$. Hence

$$\dim(\mathcal{W}_n, A) + 9 = \dim(\rho^{-1}(\mathcal{W}_n), (B, C)) \quad (\text{A9})$$

for $(B, C) \in \mathbb{R}^{2n \times 3} \times \mathbb{R}^{n \times n}$ such that $\rho(B, C) = A$. We now compute the right-hand side of (A9) in the computer algebra system Macaulay2.⁽¹⁶⁾ Specifically, we differentiate the defining constraints of $\mathcal{W}_n$ with respect to B and C , noting that the rank-3 constraint is enforced automatically by ρ and the other constraints in Proposition 3.7 are biquadratic and bilinear in (B, C) . We evaluate the Jacobian matrix at a point (B, C) in the fiber of ρ over A , where $A \in \mathcal{V}_n$ is generated as a random scaling of $\psi(R^{(1)}, \dots, R^{(n)})$ where $R^{(i)}$ are random. Generically, the nullity of the Jacobian matrix equals $\dim(\rho^{-1}(\mathcal{W}_n), (B, C))$. Performing numerical computations for $n = 3, \dots, 50$ in double-precision arithmetic yields numerical nullities of $3n + 7$. Comparing (A9) with (A8) implies (A7) as desired.

Proof of Theorem 5.1. Since nonzero row and column scaling preserves rank, we may assume without loss of generality that $\lambda_{i1} = \lambda_{1j} = 1$ for all $i, j = 2, \dots, n$: this is achieved by choosing $\mu_1 = \tau_1 = 1$, $\mu_i = \lambda_{i1}^{-1}$, and $\tau_j = \lambda_{1j}^{-1}$. We will show that $\lambda_{ij} = \lambda_{k\ell}$ for all $i, j, k, \ell = 2, \dots, n$, $i \neq j$, $k \neq \ell$. In other words, we will prove that the matrix Λ must have the form in (A12).

First, we prove that $\lambda_{ij} = \lambda_{ik}$ for all $i, j, k = 2, \dots, n$, $i \neq j$, $i \neq k$. There are three cases to consider: $i < j < k$, $j < i < k$, and $j < k < i$. Suppose we are in the first case. Since $\text{rank}(B) = 3$, every 4×4 minor of B vanishes. Thus, choosing row indices $1, 2, 2i - 1$, and $2i$, and column indices $1, i, j$, and k , we have

$$\det \begin{pmatrix} 0 & \mathbf{a}_{1i} & \mathbf{a}_{1j} & \mathbf{a}_{1k} \\ \mathbf{a}_{i1} & 0 & \lambda_{ij}\mathbf{a}_{ij} & \lambda_{ik}\mathbf{a}_{ik} \end{pmatrix} = \lambda_{ik} \det(\mathbf{a}_{1i} \ \mathbf{a}_{1j}) \det(\mathbf{a}_{i1} \ \mathbf{a}_{ik}) - \lambda_{ij} \det(\mathbf{a}_{1i} \ \mathbf{a}_{1k}) \det(\mathbf{a}_{i1} \ \mathbf{a}_{ij}) = 0. \quad (\text{A10})$$

But since A is a pure common lines matrix, $\det(\mathbf{a}_{1i} \ \mathbf{a}_{1j}) = -\det(\mathbf{a}_{i1} \ \mathbf{a}_{ij})$ and $\det(\mathbf{a}_{i1} \ \mathbf{a}_{ik}) = -\det(\mathbf{a}_{1i} \ \mathbf{a}_{1k})$ by Proposition 3.4, all of which are nonzero since A is generic. Thus $\lambda_{ij} = \lambda_{ik}$.

Expanding the 4×4 minor with the same choice of row and column indices for the remaining two cases gives us

$$\det \begin{pmatrix} \mathbf{0} & \mathbf{a}_{1j} & \mathbf{a}_{1i} & \mathbf{a}_{1k} \\ \mathbf{a}_{i1} & \lambda_{ij}\mathbf{a}_{ij} & \mathbf{0} & \lambda_{ik}\mathbf{a}_{ik} \end{pmatrix} = -\det \begin{pmatrix} \mathbf{0} & \mathbf{a}_{1i} & \mathbf{a}_{1j} & \mathbf{a}_{1k} \\ \mathbf{a}_{i1} & \mathbf{0} & \lambda_{ij}\mathbf{a}_{ij} & \lambda_{ik}\mathbf{a}_{ik} \end{pmatrix} = 0$$

and

$$\det \begin{pmatrix} \mathbf{0} & \mathbf{a}_{1j} & \mathbf{a}_{1k} & \mathbf{a}_{1i} \\ \mathbf{a}_{i1} & \lambda_{ij}\mathbf{a}_{ij} & \lambda_{ik}\mathbf{a}_{ik} & \mathbf{0} \end{pmatrix} = \det \begin{pmatrix} \mathbf{0} & \mathbf{a}_{1i} & \mathbf{a}_{1j} & \mathbf{a}_{1k} \\ \mathbf{a}_{i1} & \mathbf{0} & \lambda_{ij}\mathbf{a}_{ij} & \lambda_{ik}\mathbf{a}_{ik} \end{pmatrix} = 0,$$

respectively, which brings us back to (A10).

Suppose that $3 \leq i \leq n-1$. The 4×4 minor whose row indices are $2i-1, 2i, 2(i+1)-1$, and $2(i+1)$, and whose column indices are $1, i-1, i$, and $i+1$ is

$$\det \begin{pmatrix} \mathbf{a}_{i1} & \lambda_{i,i-1}\mathbf{a}_{i,i-1} & \mathbf{0} & \lambda_{i,i+1}\mathbf{a}_{i,i+1} \\ \mathbf{a}_{i+1,1} & \lambda_{i+1,i-1}\mathbf{a}_{i+1,i-1} & \lambda_{i+1,i}\mathbf{a}_{i+1,i} & \mathbf{0} \end{pmatrix} = \det \begin{pmatrix} \mathbf{0} & \lambda_{i,i+1}\mathbf{a}_{i,i+1} & \mathbf{a}_{i1} & \lambda_{i,i-1}\mathbf{a}_{i,i-1} \\ \lambda_{i+1,i}\mathbf{a}_{i+1,i} & \mathbf{0} & \mathbf{a}_{i+1,1} & \lambda_{i+1,i-1}\mathbf{a}_{i+1,i-1} \end{pmatrix}.$$

We showed earlier that $\lambda' := \lambda_{i,i+1} = \lambda_{i,i-1}$ and $\lambda'' := \lambda_{i+1,i} = \lambda_{i+1,i-1}$. Thus expanding, we obtain

$$\det \begin{pmatrix} \mathbf{0} & \lambda'\mathbf{a}_{i,i+1} & \mathbf{a}_{i1} & \lambda'\mathbf{a}_{i,i-1} \\ \lambda''\mathbf{a}_{i+1,i} & \mathbf{0} & \mathbf{a}_{i+1,1} & \lambda''\mathbf{a}_{i+1,i-1} \end{pmatrix} = \lambda'\lambda''^2 \det(\mathbf{a}_{i1} \ \mathbf{a}_{i,i+1}) \det(\mathbf{a}_{i+1,i-1} \ \mathbf{a}_{i+1,i}) \quad (\text{A11}) \\ - \lambda'^2 \lambda'' \det(\mathbf{a}_{i,i-1} \ \mathbf{a}_{i,i+1}) \det(\mathbf{a}_{i+1,1} \ \mathbf{a}_{i+1,i}) = 0.$$

After dividing both sides of the last equation by $\lambda'\lambda''$, we obtain $\lambda' = \lambda''$. Lastly, the 4×4 minor with the same choice of row and column indices when $i=2$ is

$$\det \begin{pmatrix} \mathbf{a}_{21} & \mathbf{0} & \lambda_{23}\mathbf{a}_{23} & \lambda_{24}\mathbf{a}_{24} \\ \mathbf{a}_{31} & \lambda_{32}\mathbf{a}_{32} & \mathbf{0} & \lambda_{34}\mathbf{a}_{34} \end{pmatrix} = \det \begin{pmatrix} \mathbf{0} & \lambda'\mathbf{a}_{23} & \mathbf{a}_{21} & \lambda'\mathbf{a}_{24} \\ \lambda''\mathbf{a}_{32} & \mathbf{0} & \mathbf{a}_{31} & \lambda''\mathbf{a}_{34} \end{pmatrix} = 0$$

since $\lambda' := \lambda_{23} = \lambda_{24}$ and $\lambda'' := \lambda_{32} = \lambda_{34}$, which brings us back to (A11). Thus we have proven that $\lambda_{ij} = \lambda_{k\ell}$ for all $i, j, k, \ell = 2, \dots, n, i \neq j, k \neq \ell$. Let λ be their common value. Then

$$\Lambda = \begin{pmatrix} 0 & 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & \lambda & \dots & \lambda & \lambda \\ 1 & \lambda & 0 & \dots & \lambda & \lambda \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \lambda & \lambda & \dots & 0 & \lambda \\ 1 & \lambda & \lambda & \dots & \lambda & 0 \end{pmatrix} \quad (\text{A12})$$

and we may obtain (40) by choosing $\mu_1 = \frac{1}{\lambda}, \mu_2 = \dots = \mu_n = 1$ and $\tau_1 = 1, \tau_2 = \dots = \tau_n = \lambda$.

Now we prove the last statement of the theorem. What we have shown so far is that $\text{rank}(B) = 3$ implies that there exists scales μ_i and τ_j such that $\lambda_{ij} = \mu_i \tau_j$. Since $\lambda_{ij} = \lambda_{ji}$, we have

$$\mu_i \tau_j = \mu_j \tau_i$$

for all $i, j = 1, \dots, n$. If $\boldsymbol{\mu} = (\mu_i)_{i=1}^n$ and $\boldsymbol{\tau} = (\tau_j)_{j=1}^n$, then this means that the matrix $\boldsymbol{\mu} \boldsymbol{\tau}^\top \in \mathbb{R}^{n \times n}$ is symmetric. This is true if and only if $\boldsymbol{\mu} = \boldsymbol{\tau}$. In particular,

$$\mu_i = \tau_i. \quad (\text{A13})$$

If B furthermore satisfies the determinant constraints, then we have

$$\mu_i^2 \tau_j \tau_k \det(\mathbf{a}_{ij} \ \mathbf{a}_{ik}) = -\mu_j^2 \tau_i \tau_k \det(\mathbf{a}_{ji} \ \mathbf{a}_{jk}) = \mu_k^2 \tau_i \tau_j \det(\mathbf{a}_{ki} \ \mathbf{a}_{kj})$$

for all $1 \leq i < j < k \leq n$, which implies that

$$\mu_i^2 \tau_j \tau_k = \mu_j^2 \tau_i \tau_k = \mu_k^2 \tau_i \tau_j.$$

Substituting (A13), we obtain

$$\tau_i^2 \tau_j \tau_k = \tau_j^2 \tau_i \tau_k = \tau_k^2 \tau_i \tau_j = \tau_j = \tau_k$$

after dividing by $\tau_i \tau_j \tau_k$. Denoting the common value of the above equation on the right by τ , we find that $\mu_i = \tau$. Hence, $\lambda_{ij} = \tau$ for all $i, j = 1, \dots, n, i \neq j$, which gives the desired result.

B. Appendix. Least squares problems

In Section 5.2, we consider the least squares problems (20) and (21), which are

$$\boldsymbol{\mu} = \arg \min_{\|\boldsymbol{\mu}\|_2 = 1} \|\text{diag}(\boldsymbol{\mu})M - (\text{diag}(\boldsymbol{\mu})M)^\top\|_F^2,$$

$$\boldsymbol{\tau} = \arg \min_{\|\boldsymbol{\tau}\|_2 = 1} \|M \text{diag}(\boldsymbol{\tau}) - (M \text{diag}(\boldsymbol{\tau}))^\top\|_F^2.$$

These have solutions

$$\min_{\|\boldsymbol{\mu}\|_2 = 1} \|N_L \cdot \boldsymbol{\mu}\|_2^2,$$

$$\min_{\|\boldsymbol{\tau}\|_2 = 1} \|N_R \cdot \boldsymbol{\tau}\|_2^2,$$

respectively, where $N_L \in \mathbb{R}^{n \times n}$ with

$$(N_L)_{ij} = \begin{cases} \sum_{k=1, \dots, n} M_{ik}^2 & \text{if } i=j \\ -M_{ij}M_{ji} & \text{if } i \neq j \end{cases} \quad (\text{B1})$$

and $N_R \in \mathbb{R}^{n \times n}$ with

$$(N_R)_{ij} = \begin{cases} \sum_{k=1, \dots, n} M_{ki}^2 & \text{if } i=j \\ -M_{ij}M_{ji} & \text{if } i \neq j \end{cases}. \quad (\text{B2})$$

We also consider the least squares problems (29) and (30), which are

$$\min_{\|\boldsymbol{\mu}\|_2 = 1} \|(\boldsymbol{\mu} \Delta \mathbf{v}_1) - (\boldsymbol{\mu} \Delta \mathbf{v}_2)\|_2^2 + \|(\boldsymbol{\mu} \Delta \mathbf{v}_2) - (\boldsymbol{\mu} \Delta \mathbf{v}_3)\|_2^2, \quad (\text{B3})$$

$$\min_{\|\boldsymbol{\tau}\|_2 = 1} \|(\boldsymbol{\tau} \Delta_1 \mathbf{v}_1) - (\boldsymbol{\tau} \Delta_1 \mathbf{v}_2)\|_2^2 + \|(\boldsymbol{\tau} \Delta_2 \mathbf{v}_2) - (\boldsymbol{\tau} \Delta_2 \mathbf{v}_3)\|_2^2, \quad (\text{B4})$$

where

$$\begin{aligned} \boldsymbol{\mu} \Delta \mathbf{v}_1 &:= (\mu_i \text{sgn}(\det(\mathbf{a}_{ij} \ \mathbf{a}_{ik}))) \sqrt{|\det(\mathbf{a}_{ij} \ \mathbf{a}_{ik})|} \Big|_{1 \leq i < j < k \leq n} \\ \boldsymbol{\mu} \Delta \mathbf{v}_2 &:= (\mu_j \text{sgn}(\det(\mathbf{a}_{ji} \ \mathbf{a}_{jk}))) \sqrt{|\det(\mathbf{a}_{ji} \ \mathbf{a}_{jk})|} \Big|_{1 \leq i < j < k \leq n} \\ \boldsymbol{\mu} \Delta \mathbf{v}_3 &:= (\mu_k \text{sgn}(\det(\mathbf{a}_{ki} \ \mathbf{a}_{kj}))) \sqrt{|\det(\mathbf{a}_{ki} \ \mathbf{a}_{kj})|} \Big|_{1 \leq i < j < k \leq n} \end{aligned} \quad (\text{B5})$$

and

$$\begin{aligned} \boldsymbol{\tau} \Delta_1 \mathbf{v}_1 &:= (\tau_j \det(\mathbf{a}_{ij} \ \mathbf{a}_{ik})) \Big|_{1 \leq i < j < k \leq n} & \boldsymbol{\tau} \Delta_1 \mathbf{v}_2 &:= (-\tau_i \det(\mathbf{a}_{ji} \ \mathbf{a}_{jk})) \Big|_{1 \leq i < j < k \leq n} \\ \boldsymbol{\tau} \Delta_2 \mathbf{v}_2 &:= (-\tau_k \det(\mathbf{a}_{ji} \ \mathbf{a}_{jk})) \Big|_{1 \leq i < j < k \leq n} & \boldsymbol{\tau} \Delta_2 \mathbf{v}_3 &:= (\tau_j \det(\mathbf{a}_{ki} \ \mathbf{a}_{kj})) \Big|_{1 \leq i < j < k \leq n} \end{aligned} \quad (\text{B6})$$

are all vectors of length $\binom{n}{3}$. The solution to problem (B3) is

$$\min_{\|\boldsymbol{\mu}\|_2 = 1} \|(D_{L,1} + D_{L,2}) \cdot \boldsymbol{\mu}\|_2^2,$$

where $D_{L,1}, D_{L,2} \in \mathbb{R}^{n \times n}$ with

$$(D_{L,1})_{pq} = \begin{cases} \sum_{1 \leq j < k \leq n} \text{sgn}(v_{pj,pk}) |v_{pj,pk}| & \text{if } p=q \\ -\sum_{\substack{j \neq p, k > p \\ \max\{p,q\} < k \leq n}} \text{sgn}(v_{pq,pk} v_{qp,qk}) \sqrt{|v_{pq,pk}|} \sqrt{|v_{qp,qk}|} & \text{if } p \neq q \end{cases} \quad (\text{B7})$$

$$(D_{L,2})_{pq} = \begin{cases} \sum_{1 \leq j < k \leq n} \text{sgn}(v_{pk,pj}) |v_{pj,pk}| & \text{if } p = q \\ j \neq p, k > p & \\ - \sum_{\max\{p,q\} < k \leq n} \text{sgn}(v_{pq,pk} v_{qp,qk}) \sqrt{|v_{pq,pk}|} \sqrt{|v_{qp,qk}|} & \text{if } p \neq q \end{cases} \quad (\text{B8})$$

and the solution to problem (B4) is

$$\min_{\|\mathbf{r}\|_2 = 1} \| (D_{R,1} + D_{R,2}) \cdot \mathbf{r} \|_2^2$$

where $D_{R,1}, D_{R,2} \in \mathbb{R}^{n \times n}$ with

$$(D_{R,1})_{pq} = \begin{cases} \sum_{1 \leq j < k \leq n} v_{jp,jk}^2 & \text{if } p = q \\ j \neq p, k > p & \\ - \sum_{\max\{p,q\} < k \leq n} v_{pq,qk} v_{qp,qk} & \text{if } p \neq q \end{cases}, \quad (\text{B9})$$

$$(D_{R,2})_{pq} = \begin{cases} \sum_{1 \leq k < j \leq n} v_{jk,jp}^2 & \text{if } p = q \\ j \neq p, k < p & \\ - \sum_{1 \leq k < \max\{p,q\}} v_{pk,pq} v_{qk,qp} & \text{if } p \neq q \end{cases}, \quad (\text{B10})$$

where $v_{ij,ik} := \det(\mathbf{a}_{ij} \ \mathbf{a}_{ik})$.