



SCIENCE
Mathematical Sciences

An Application of Mapper

Analyzing trajectories of macrophages

Jae S. Bang

Outline for This Presentation

1. Related Prior Works
 - 1.1 Backgrounds
 - 1.2 Data Structure
 - 1.3 Methodology
 - 1.4 Potential Extensions
2. Mapper Algorithm
 - 2.1 Motivation of Mapper
 - 2.2 Pipeline of Mapper Algorithm
 - 2.3 Ball Mapper
 - 2.4 Multidimensional Scaling
3. Application of Mapper

Backgrounds and Motivation

Backgrounds[Park et al., 2026]

- Traditionally, macrophages in wound healing have been described in two main phases: M1 state¹, and M2 state.²
- It has been observed that their state correlates with distinct morphological and migratory features across different model systems.

Motivation[Park et al., 2026]

Classifying their states provides valuable insights into macrophage functions during tissue repair and helps guide therapeutic strategies for inflammatory and wound healing disorders.

¹initial phase; inflammatory

²subsequent phase; anti-inflammatory and tissue repairing

Problems and Goals

Problem Statement[Park et al., 2026]

1. It has been reported that macrophages activation occurs along a continuum, **not strictly conforming** to the M1/M2 framework.
2. Cells exhibit heterogeneous morphologies and diverse motility patterns over time. Quantitative analysis *in vivo* settings remains limited.

⇒ Comprehensive analysis of macrophages is required.

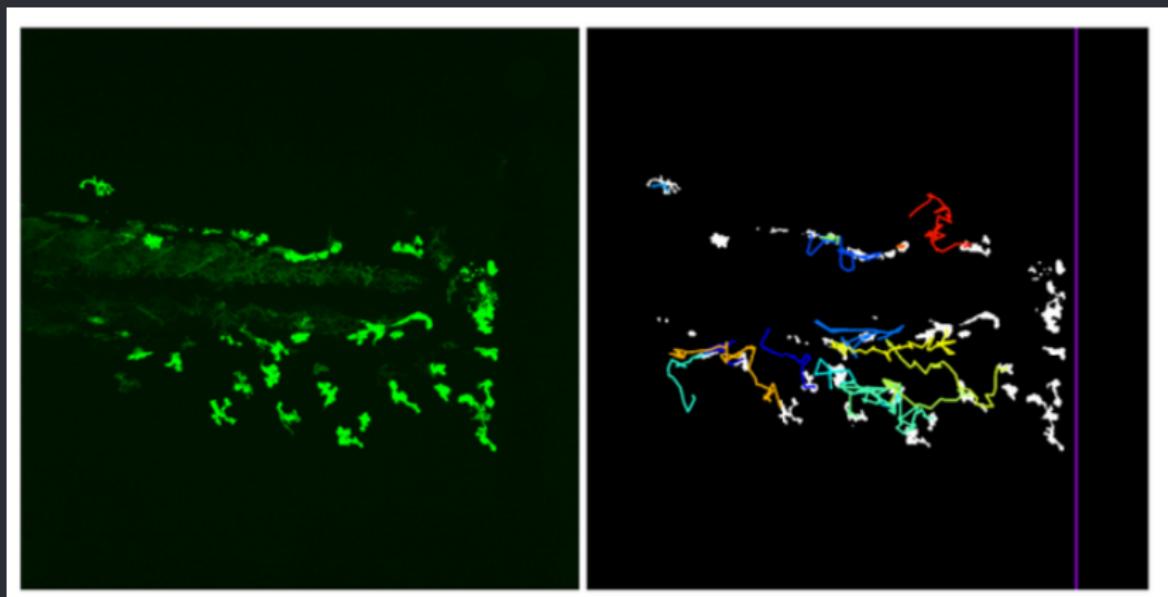
Research Goal[Park et al., 2026]

1. Analyze morpho-kinetic features, i.e. both morphological and dynamic features.
2. Classify the states of macrophages in transition period, and estimate the timing of the phenotypic switch.

Data Structure

	Video No.	Pixel Width (μm)	Pixel Height (μm)	Time Step (min)	Time Range
M1-like	1	0.32	0.32	2.5	0.5–6.0 hpa
	2	0.32	0.32	2	0.6–6.0 hpa
	3	0.32	0.32	2.5	1.0–6.0 hpa
	4	0.32	0.32	1	2.0–6.0 hpa
	5	0.32	0.32	2.5	1.5–6.0 hpa
M2-like	6	0.32	0.32	2.5	10.5–16.0 hpa
	7	0.32	0.32	2.5	10.5–16.0 hpa
TP	8	0.32	0.32	2.5	6.0–10.5 hpa
	9	0.32	0.32	2.5	6.0–10.5 hpa
	10	0.32	0.32	2.5	6.0–10.5 hpa
	11	0.32	0.32	2.5	6.0–10.5 hpa
	12	0.32	0.32	2.5	6.0–10.5 hpa
NonM1-like	13	0.32	0.32	2.5	0.5–6.0 hpa
	14	0.32	0.32	2	0.6–6.0 hpa
	15	0.32	0.32	2.5	1.0–6.0 hpa
	16	0.32	0.32	1	2.0–6.0 hpa
	17	0.32	0.32	2.5	1.5–6.0 hpa
unM	18	0.31	0.31	1.5	0–5.5 h
	19	0.31	0.31	1.5	0–5.5 h
	20	0.31	0.31	1.5	0–5.5 h

Data Structure



Feature	Description	Statistics	Trajectory type
Area (area)	The area of a macrophage from a trajectory at a time frame	mean_-, std_-, med_-, max_-, min_-	None
Perimeter (peri)	The perimeter of a macrophage from a trajectory at a time frame	mean_-, std_-, med_-, max_-, min_-	None
Circularity (cir)	The circularity of a macrophage from a trajectory at a time frame	mean_-, std_-, med_-, max_-, min_-	None
Random Length Ratio (r_{len})	Ratio of the average length of random parts to the total length of the trajectory	None	None
Ratio of Random Segments (r_{seg})	Ratio of line segments belonging to random parts relative to the total number of time frames	None	None
Meander Ratio (M)	Ratio of the total trajectory length to the displacement between the initial and final points	None	_ori, _smo
Tangent Projection Metrics (TPM)	Projection of trajectory tangent vectors onto the outward wound normal vector	mean_-, std_-, med_-, max_-, min_-	_ori, _smo
Positive TPM Ratio (posT)	Ratio of points with positive TPM values	None	_ori, _smo
Negative TPM Ratio (negT)	Ratio of points with negative TPM values	None	_ori, _smo
Velocity Projection Metrics (VPM)	Projection of velocity vectors onto the outward wound normal vector	mean_-, std_-, med_-, max_-, min_-	_ori, _smo
Distance to Wound (distW)	Distances from the wound across all points of trajectories	mean_-, std_-, med_-, max_-, min_-	_ori, _smo
Speed (sp)	Speeds across all points of trajectories	mean_-, std_-, med_-, max_-, min_-	_ori, _smo

Data Structure

time (min)	area (micrometer^2)	perimeter (micrometer)	Dist_to_W (micrometer)
235.0	183.193907	74.901234	452.58059
237.5	208.300147	73.468343	449.652299
240.0	164.007024	65.646153	450.08049
242.5	132.573195	77.876637	450.218577
245.0	158.087667	70.00904	450.600738
247.5	182.989791	69.899417	449.698328
250.0	196.461432	99.078741	452.313501
252.5	211.565999	87.131718	452.718677
255.0	204.626062	82.452160	441.620661

... (more features)

Methodology

1. Trajectory-to-feature representation.

Extract 63 features from each trajectory, which becomes a point in \mathbb{R}^{63} . Therefore we get points in \mathbb{R}^{63} as many as the number of trajectories we have.

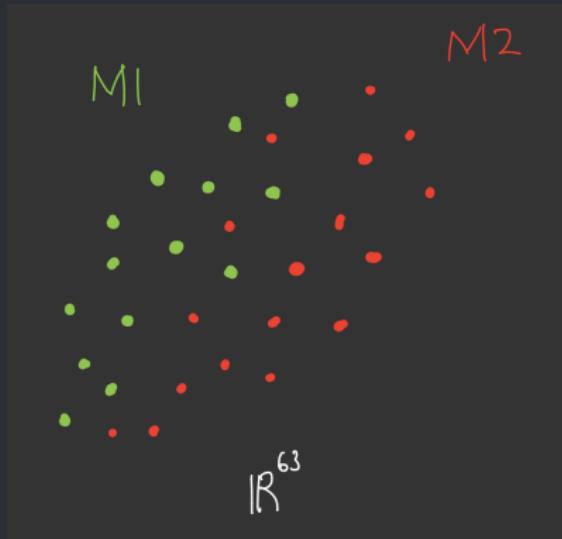


Trajectory

Methodology

2. Reference phenotypes for training.

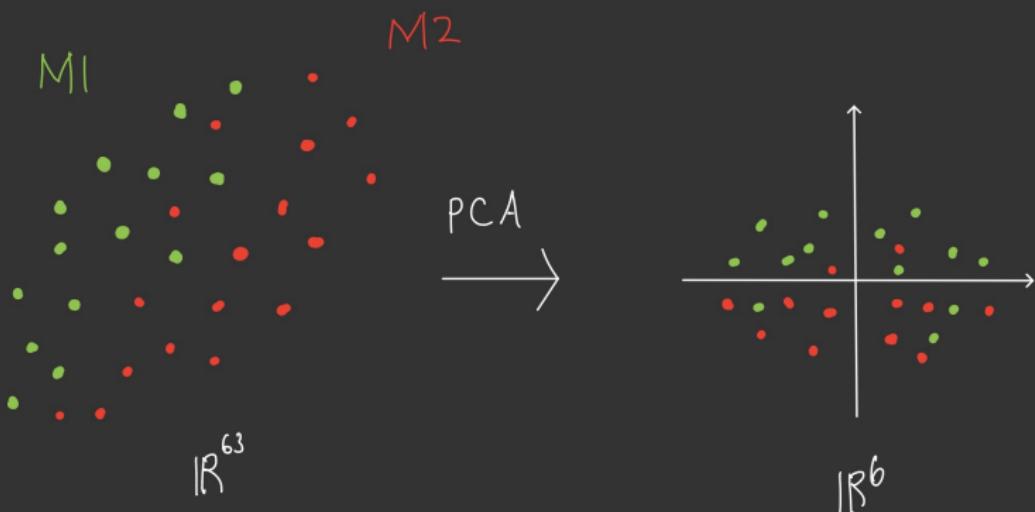
Trajectories from the early wound phase (M1-like) and the late wound/resolution phase (M2-like) are used as labeled reference data.



Methodology

3. Dimensionality reduction with PCA.

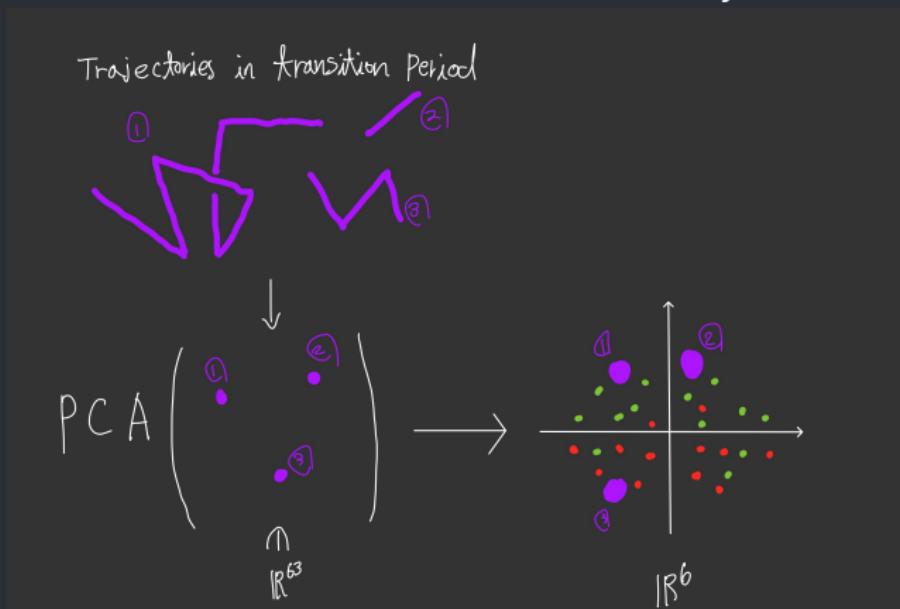
PCA(Principal Component Analysis) is fitted on the labeled reference feature matrix and the feature space is reduced to six major principal components, i.e. each trajectory is represented by its coordinates in this 6D PCA space.



Methodology

4. Classification of transition-period trajectories.

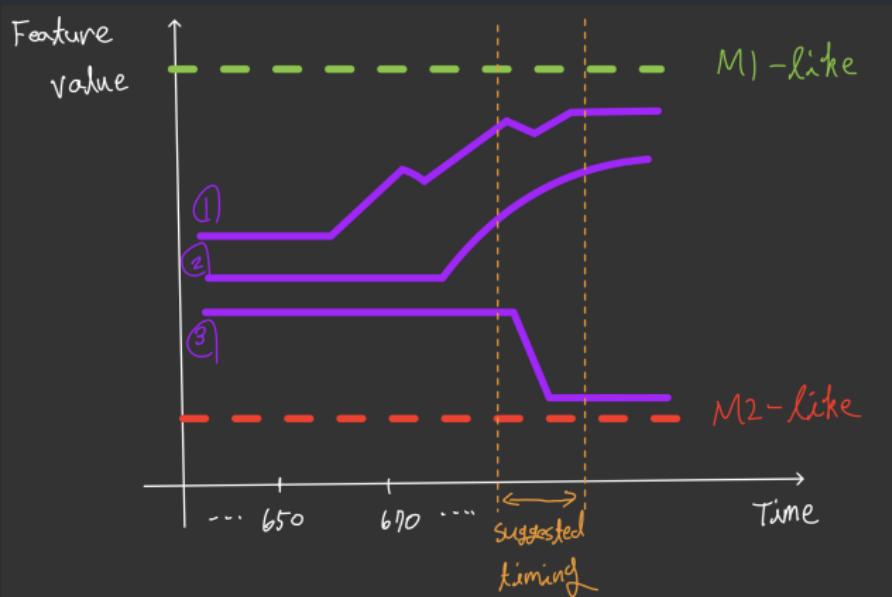
Trajectories from the transition period are projected into the same PCA space and classified as M1-like or M2-like using a classifier trained on the labeled reference trajectories.



Methodology

5. Estimating switch timing.

We estimate the M1→M2 transition period by comparing time-binned (20-min) trends of key features in the classified TP trajectories.



Potential Extensions

- **Improved interpretability.**

Since classification is performed in a 6D PCA embedding, reporting feature-attribution could clarify which of the 63 morpho-kinetic features primarily drive the decisions.

- **Richer multivariate integration.**

While the study presents clear univariate and pairwise summaries, complementary multivariate interpretability tools could help reveal higher-order feature interactions.

- **Incorporating within-trajectory dynamics.**

Because features are computed as summary statistics over an entire trajectory, the current representation does not explicitly capture within-trajectory temporal dynamics.

Outline for the section 2

1. Related Prior Works

1.1 Backgrounds

1.2 Data Structure

1.3 Methodology

1.4 Potential Extensions

2. Mapper Algorithm

2.1 Motivation of Mapper

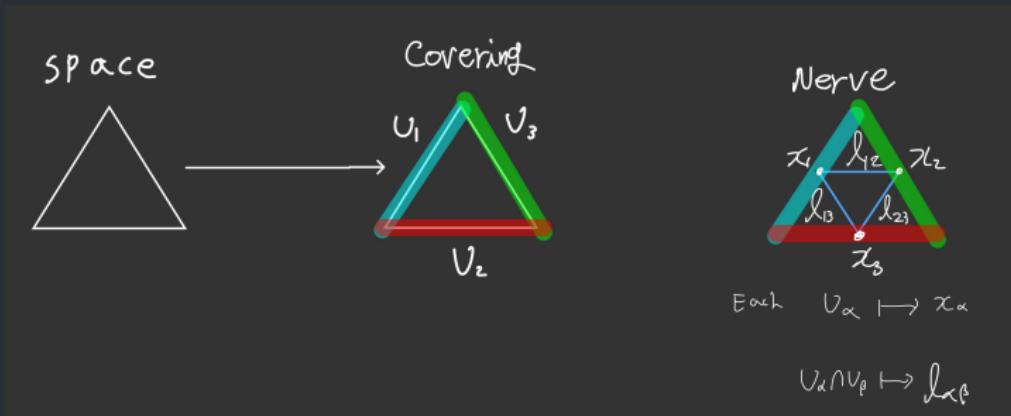
2.2 Pipeline of Mapper Algorithm

2.3 Ball Mapper

2.4 Multidimensional Scaling

3. Application of Mapper

Motivation of Mapper: Nerve



Definition (Nerve)

Let X be a topological space. Given a finite covering $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of the space X , the **nerve** of the covering \mathcal{U} is a simplicial complex $N(\mathcal{U})$ whose vertex set is the indexing set A , and where a family $\{\alpha_0, \alpha_1, \dots, \alpha_k\}$ spans a k -simplex in $N(\mathcal{U})$ if and only if $U_{\alpha_0} \cap U_{\alpha_1} \cap \dots \cap U_{\alpha_k} \neq \emptyset$.

Motivation of Mapper: Nerve

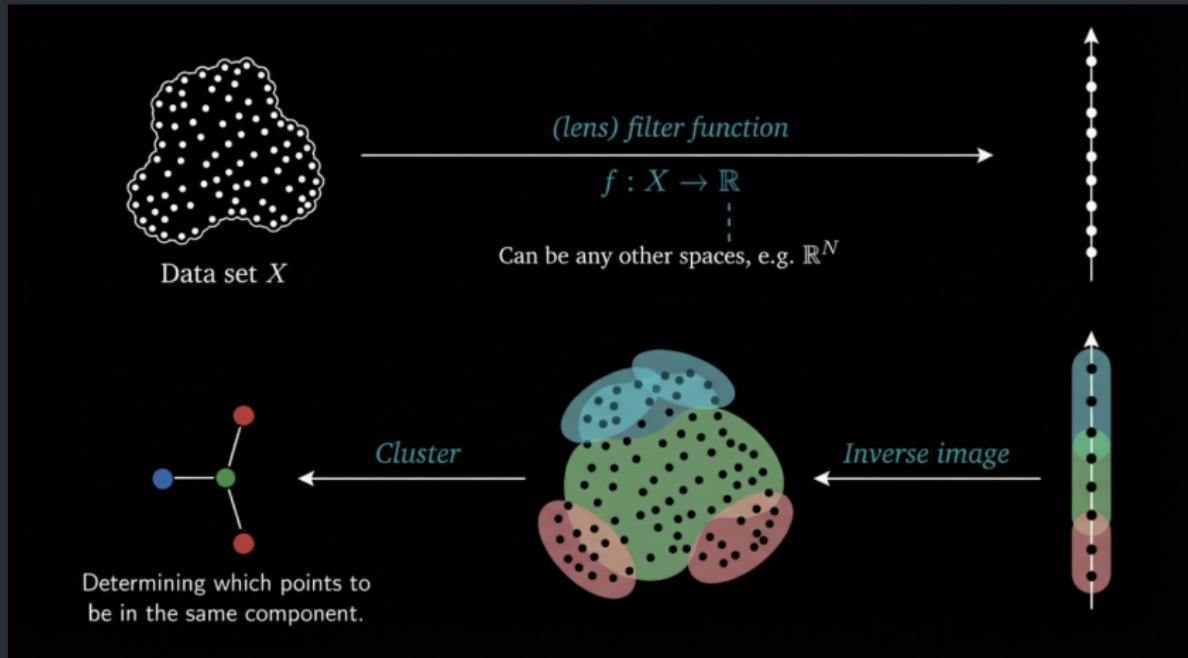
Theorem (Nerve Theorem)

If \mathcal{U} is a good cover of X , i.e. $\bigcap_{\alpha \in A'} U_\alpha$ is either empty or contractible for all $A' \subseteq A$, then $N(\mathcal{U})$ is homotopy equivalent to X .

Therefore, our task is nothing more or less than to construct a **good cover**.

Pipeline of Mapper Algorithm

- How can we construct a covering on discrete data set X ?



There are too much parameters!

Construction of a Mapper Graph

Let a data set X be given.

Parameter 1: Distance matrix given by the chosen metric

1. Choose a filter function f .

→ Parameter 2: Choice of filter function

2. Cover the image of f .

→ Parameter 3,4: The number of coverings, and the overlapping percentage

3. Cluster each inverse image.

→ Parameter 5: Clustering algorithm

There are too much parameters!

The result may be different dramatically:

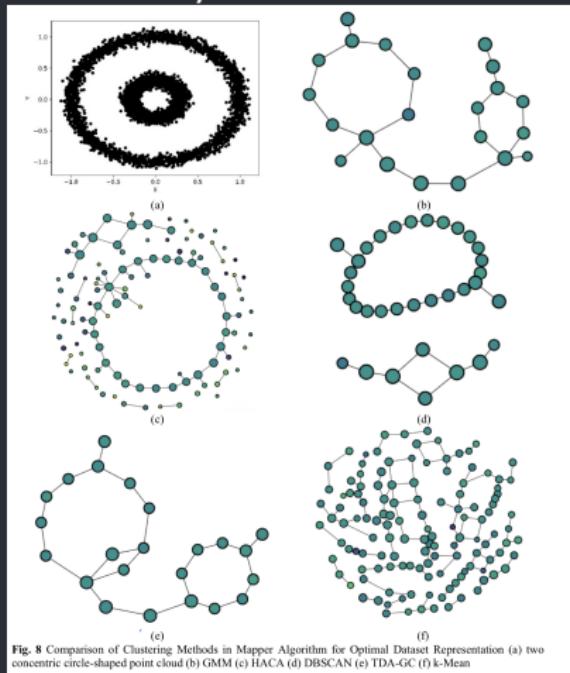


Fig. 8 Comparison of Clustering Methods in Mapper Algorithm for Optimal Dataset Representation (a) two concentric circle-shaped point cloud (b) GMM (c) HACA (d) DBSCAN (e) TDA-GC (f) k-Mean

1. Lens: a projection onto y-axis
2. 15 cover partitions, a 20% overlap between covers.
3. Only used different clustering algorithms.

Ball Mapper

An alternative way of obtaining an overlapping cover of X which only requires two parameters: metric, and radius r .

Rough Algorithm of the Ball Mapper

1. Choose $X' \subset X$ (at least one arbitrary choice³) such that $X \subseteq \bigcup_{b \in X'} B(b, r)$. Each $B(b, r)$ becomes a node.
2. Add an edge between nodes $B(b_1, r)$ and $B(b_2, r)$ if and only if $B(b_1, r) \cap B(b_2, r) \neq \emptyset$. Do this for all pairs. One can extend this idea to higher dimensions.

It is known that the ball mapper summarizes the overall structure of data well.

³Therefore, the stability issue exists.

Ball Mapper

For example, one can show the following theorem:

Theorem

Let X be a point cloud and $B \subset X$ be a collection of ball centers forming an ϵ -net. Then, the following sequence of inclusions holds:

$$\bigcup_{b \in B} B(b, \epsilon) \subset \bigcup_{x \in X} B(x, \epsilon) \subset \bigcup_{b \in B} B(b, 2\epsilon)$$

Furthermore, this can be extended to an infinite sequence of inclusions:

$$\bigcup_{b \in B} B(b, \epsilon) \subset \bigcup_{x \in X} B(x, \epsilon) \subset \bigcup_{b \in B} B(b, 2\epsilon) \subset \bigcup_{x \in X} B(x, 2\epsilon) \subset \dots$$

Ball Mapper

Corollary (Approximation of Persistent Homology)

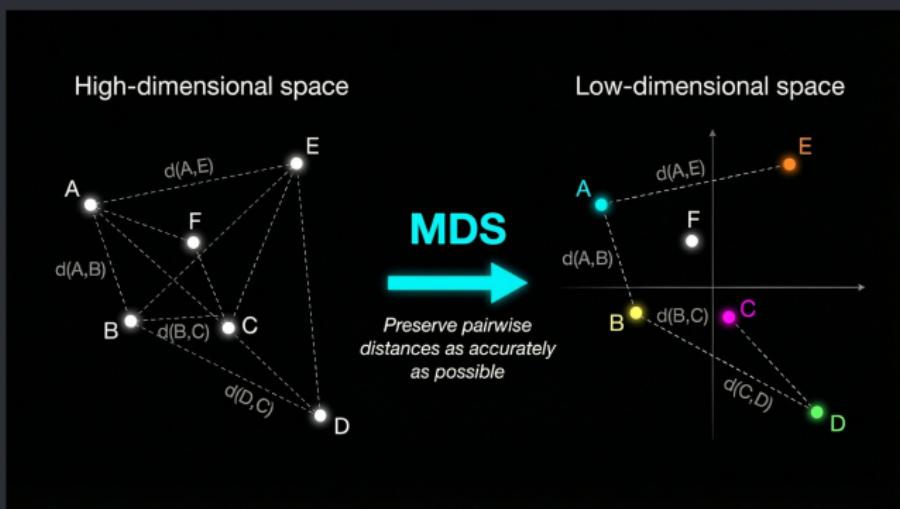
The extended sequence of spatial inclusions induces the following interleaved sequence of homology groups:

$$\begin{array}{ccc} H\left(\bigcup_{x \in X} B(x, i\epsilon)\right) & \longrightarrow & H\left(\bigcup_{x \in X} B(x, (i+1)\epsilon)\right) \\ \uparrow & \searrow & \uparrow \\ H\left(\bigcup_{b \in B} B(b, i\epsilon)\right) & \longrightarrow & H\left(\bigcup_{b \in B} B(b, (i+1)\epsilon)\right) \end{array}$$

Consequently, the persistent homology of the original space X can be approximated by computing the nerve complex of the subsampled points in B .

Multidimensional Scaling

- Multidimensional Scaling (MDS) is a dimension reduction technique.
- **Objective:** To find a low-dimensional representation of data that preserves the original pairwise distances between data points as accurately as possible.



Outline for the section 3

1. Related Prior Works

1.1 Backgrounds

1.2 Data Structure

1.3 Methodology

1.4 Potential Extensions

2. Mapper Algorithm

2.1 Motivation of Mapper

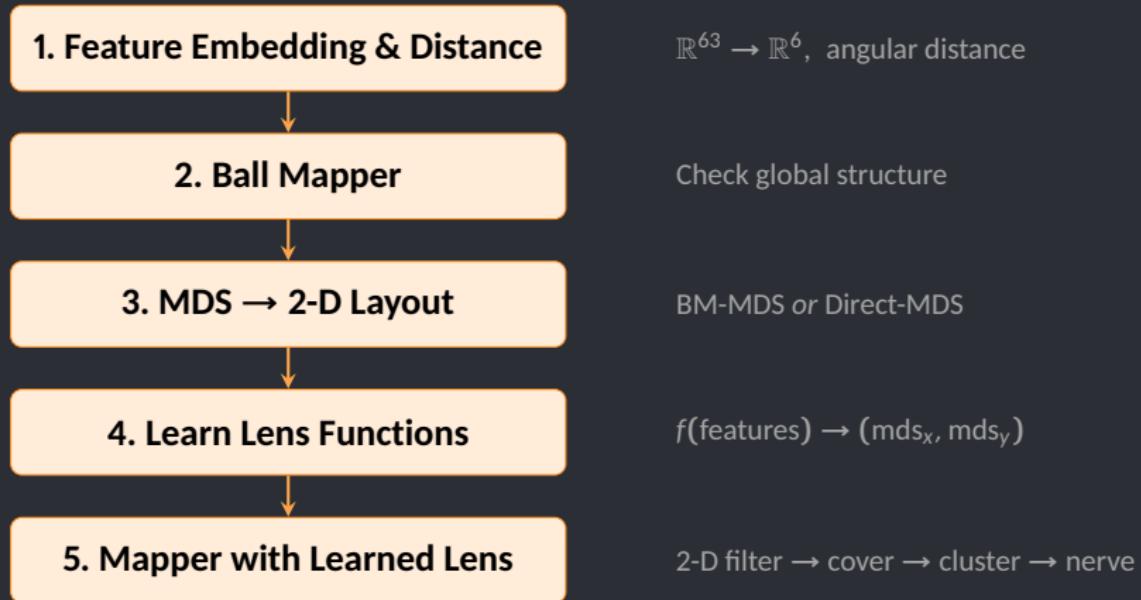
2.2 Pipeline of Mapper Algorithm

2.3 Ball Mapper

2.4 Multidimensional Scaling

3. Application of Mapper

Our Pipeline: Overview



What I Expected to See in the Mapper Graph

1. M1-like and M2-like trajectories form **distinct regions**.
2. Transition-period trajectories **interpolate** between them.
3. The **time gradient** aligns with the M1→M2 axis, providing an independent estimate of the switch timing.

Step 1: Feature Embedding & Distance Matrix

1. Select embedding features.

From the full 63-dimensional feature vector, choose a subset of interpretable features:

$$(\text{Area}, \text{Perimeter}, \text{Dist_W}, \text{TPM}, \text{Speed}, \text{Circularity}) \in \mathbb{R}^6.$$

2. Normalize and compute distance.

Each feature column is **standardized**, then the **angular (cosine)** distance matrix is computed:

$$d_{\text{ang}}(x_i, x_j) = \sqrt{2 \left(1 - \frac{\langle x_i, x_j \rangle}{\|x_i\| \|x_j\|} \right)}$$

Step 2: Ball Mapper — Structural Overview

- Apply Ball Mapper with the precomputed distance matrix D .
- Radius ε is chosen from the **k -NN distance quantile**:

$$\varepsilon = Q_{0.95}(\{d_k(x)\}_{x \in X}), \quad k = 10.$$

Result

I was able to observe that there are "some" structures between phase switching timing and features. I was able to get an immediate visual insight by coloring nodes by *time*, *M1-membership*, *M2-membership*, or individual features.

Step 3: MDS

- Motivation.
Ball mapper graph showed some structures between “time” and the locations of nodes, so it was worth trying to press the graph onto \mathbb{R}^2 plane.
- Goal.
Find functions that represent the MDS coordinates well, and by doing so I expected that I can get lens functions that can show the “switching” time.

Step 3: MDS — Two Approaches

Ball mapper based MDS

1. Aggregate each node's member points into a **representative** (median).
2. Compute inter-node angular distances.
3. Apply MDS to the *node* distance matrix.
4. Assign each data point the average MDS coordinate of the nodes it belongs to.

Direct MDS

1. Apply MDS **directly** to the full $N \times N$ distance matrix D .
2. Each data point receives its own MDS coordinate immediately.

Step 4: Learning Lens Functions

Key idea. MDS coordinates themselves are *not* functions of features.
We **learn** an explicit map:

$$\ell: \mathbb{R}^5 \longrightarrow \mathbb{R}^2, \quad \ell(\mathbf{x}) = (\ell_x(\mathbf{x}), \ell_y(\mathbf{x})),$$

such that $\ell(\mathbf{x}_i) \approx (\text{mds}_x^{(i)}, \text{mds}_y^{(i)}).$

Regression Models

Linear (deg = 1): $\ell_x(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b.$

Polynomial (deg = 2): Includes all cross-terms $x_j x_k$ and $x_j^2.$

Custom transforms: Hand-crafted nonlinear basis functions

Step 5: Mapper with the Learned Lens

Once the lens $\ell: \mathbb{R}^5 \rightarrow \mathbb{R}^2$ is learned:

1. **Filter.** Apply ℓ to every trajectory segment \rightarrow 2-D filter values $(\ell_x^{(i)}, \ell_y^{(i)})$.
2. **Cover.** Partition $\ell(\mathbb{R}^5) \subset \mathbb{R}^2$ with a rectangular grid (20×20 , overlap 30%).
3. **Cluster.** Within each cover element, cluster using DBSCAN on the precomputed distance matrix.
4. **Nerve.** Connect clusters that share data points \rightarrow the **Mapper graph**.

Advantages of Mapper

- The lens is an **explicit, interpretable formula** of morpho-kinetic features — not a black-box projection.
- Comparing lenses of different complexity reveals which features drive the data topology.
- By choosing features well, I was able to consider dynamic aspects of trajectories, not just overall static trajectories.

Summary & Next Steps

Summary

1. Ball Mapper provides a **parameter-light structural overview** of the morpho-kinetic feature space.
2. MDS + regression yields an **explicit, interpretable 2-D lens** from biological features.
3. The learned lens, fed into Mapper, produces a graph that **separates phenotypes** and captures the $M1 \rightarrow M2$ transition.

Ongoing & future work

- Interpret the linear lens function biologically.
- Divide data into train($M1 \& M2$) and test sets(Transition), and check whether the lens functions detect the switching period.