TOMATOMP: Robust and Multi-Parameter Topological Clustering

Ludo Andrianirina **Mathieu Carrière**[1]*
[1]DataShape, Centre Inria d'Université Côte d'Azur, Sophia-Antipolis, France

Abstract

Topological clustering, and its main algorithm TOMATO, is a method from Topological Data Analysis (TDA) that has been applied successfully in several applications during the last few years. This is due to its high versatility, as clusters are detected from the persistent components in the superlevel sets of any user-defined function (gene expression, pixel values, etc), and to its high efficiency, as topological clustering enjoys robustness guarantees. However, TOMATO is also limited in several ways. First, a graph on the data points needs to be provided as a hyperparameter of the method (whose fine-tuning is left to the user). Second, TOMATO is known to be very sensitive to outlier values in the function range. Finally, and most importantly, TOMATO can only handle one function at a time, whereas it is critical to use several functions in various applications. **In this article, we introduce TOMATOMP: the first topological clustering method able to handle several functions at the same time with theoretical guarantees.** More specifically, we leverage a recent tool from multi-parameter persistent homology, called MMA decompositions, to design our clustering algorithm and prove that it enjoys robustness properties. As corollaries, we show that it can be used to make TOMATO independent of graph tuning and robust to outliers. Finally, we provide a set of numerical experiments showcasing the efficiency and quality of the clusterings produced by TOMATOMP, by showing strong improvement over non-topological and topological baselines for various datasets.

1 Introduction

In the field of unsupervised learning, the *topological clustering* method, originally introduced in [8], has recently become increasingly relevant in various application scenarios. Intuitively, this method belongs to the family of hierarchical clustering methods, where the hierarchy is built on top of the outputs of *mode-seeking* algorithms, which aim at finding the *modes*, or *basins of attraction*, of a density estimator f (such as, e.g., Mean Shift [9] or [18]).

However, contrary to traditional mode-seeking algorithms, which suffer from several caveats including, e.g., sensitivity to initialization or noise, the modes obtained with topological clustering are captured using the toolset of *persistent homology* (PH), which not only allows to build a hierarchy between modes, but also ensures theoretical robustness of the computed clusters, which usual hierarchical clustering algorithms typically lack (as illustrated, for instance, by the empirical instability of dendrograms [5]).

Moreover, while topological clustering was introduced with density estimators to mimic the framework of mode-seeking algorithms, the modes of many other functions have been introduced and used in the last few years, each related to the tasks and datasets at hand: Heat Kernel Signature for segmenting 3D shapes [27], smoothed gene expression for quantifying spatial variations [4],

*Corresponding author: mathieu.carriere@inria.fr

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

gene expression gradient for segmenting tissues [28], pixel values for capturing morphological features [10], etc. This illustrates the high versatility of topological clustering, as well as its efficiency, in very different applications.

The main topological clustering algorithm is called ToMATo [8] (though several variants exist in the literature, such as AuToMATo [16] or STopover [2]), and builds a hierarchy by processing the data points in decreasing order of the function values, which can be thought of as a scan of the dataset; each time a local maxima is encountered, a corresponding mode has been found. Moreover, these modes are progressively filled in by looking at the connected components that *persist* as the scanning of the data points goes on: after its detection with its local maxima, the mode will progressively grow until the scan reaches a saddle point, which connects the mode to another one. This interval (from the mode detection to its merge) is called the *lifetime*, or *prominence*, of the mode, and quantifies its importance. The merging also allows to build a hierarchy (by recording which modes get merged to which), that, in turn, allows to only keep the modes that pass a given prominence threshold (similarly to cutting a dendrogram at a given height). See Figure 1

Limitations of ToMATo. However, ToMATo suffers from a few weaknesses. First, deciding whether a data point is a regular point, a saddle, a local minimum or a local maximum requires to build a graph G on top of the data points (point types are then assessed based on the neighbor values). Such graph is a hyper-parameter whose tuning is left to the user, despite being critical for the clustering quality. Second, while the PH framework allows to guarantee that ToMATo clusters are robust to small function perturbations, ToMATo is sensitive to *outlier* values: only a few points with aberrant function values are sufficient to make ToMATo clusters completely uninformative. Finally, there are many applications where the modes of *several* functions are relevant (for instance, pairs or triplets of gene expressions), but ToMATo can only handle one function at a time.

Contributions. In this article, we propose ToMAToMP: the first (to our knowledge) multi-parameter topological clustering algorithm with theoretical guarantees. In particular:

- (i) We leverage the recent framework of *multi-parameter persistent homology*, and, in particular, *MMA decompositions* [20], to design a new multi-parameter topological clustering algorithm ToMAToMP (Algorithm 1). Our code can be found at: `https://github.com/MathieuCarriere/tomatomp`.
- (ii) We prove that ToMAToMP enjoys a robustness guarantee (Theorem 2) that is similar to the one of ToMATo in the single-parameter case.
- (iii) As direct corollaries, we prove that ToMAToMP allows to (a) use ToMATo without any graph parameter tuning, and (b) make ToMATo robust against outliers (Corollaries 1 and 2), hence fixing two critical limitations of ToMATo.
- (iv) We demonstrate the relevance of our method on several applications, ranging from 3D shape segmentation to gene rankings in spatial transcriptomics datasets, with favorable results against standard topological and non-topological baselines.

Related work. While topological clustering is still a recent field, extending standard topological clustering (with or without PH) has been studied in various contexts. For instance, Huber et al. [16] have proposed bootstrap methods for automatically tuning the prominence threshold in order to capture only those statistically significant modes. Kim and Mémoli [17] have used the framework of multi-parameter PH to handle and quantify the evolution of dynamic hierarchical clusterings when datasets change over time. More closely related to our work, Rolle and Scoccola [25, 26] have proposed to study the stability of *slices* (i.e., reductions of several functions into one using, e.g., linear combinations) of multi-parameter PH in order to design robust one-parameter topological clusterings. While this approach provides nice stability guarantees, it also requires the user to pick a slice, which can be seen as a new hyper-parameter. On the other hand, we emphasize that our proposed ToMAToMP method has essentially the same hyperparameters than ToMATo and thus does not require to pick a specific slice; instead, it leverages the whole multi-parameter PH structures.

Plan. In Section 2, we recall the basics of ToMATo and (multi-parameter) PH, with a focus on MMA decompositions. Then, in Section 3.1, we introduce our new algorithm ToMAToMP, and present its theoretical properties in Section 3.2. Finally, numerical experiments are showcased in Section 4, and concluding remarks are provided in Section 5.

2 Background

In this section, we recall the basics of ToMATo and (multi-parameter) PH. See Appendix A.1 and B and Algorithm 2 for more details, and [14, 23, 13] for thorough presentations. We first provide an (informal) description of ToMATo, and then connect it to (multi-parameter) PH.

Persistence-Based Clustering: ToMATo. The main idea of ToMATo is to compute clusters as the modes, or basins of attraction, of a continuous function f, and to use *persistence theory* to build a hierarchy between clusters. Roughly speaking, the algorithm works by taking as input a graph G whose vertex set $V(G)$ is the dataset $\hat{X}_n$. Then, points in $\hat{X}_n$ are sorted in decreasing function order, and processed in different ways depending on the point type (which is assessed with the point neighbors in G): local maxima induce the creation of new modes that start persisting, while saddles and local minima induce the merging of two modes and the end of one of the two modes' persistences; as the modes with lowest maxima are always merged to the modes with highest ones, this induces a hierarchy between modes. Each mode is represented with a point $p \in \mathbb{R}^2$, using the function values of the corresponding local maxima and saddle/local minima, called the *birth* and *death* times respectively; their difference is called the *prominence* $\mathrm{prom}(p)$ of the mode. Such points are stored in a so-called *persistence diagram* $\mathrm{Dgm}_0(f)$. See Figure 1.

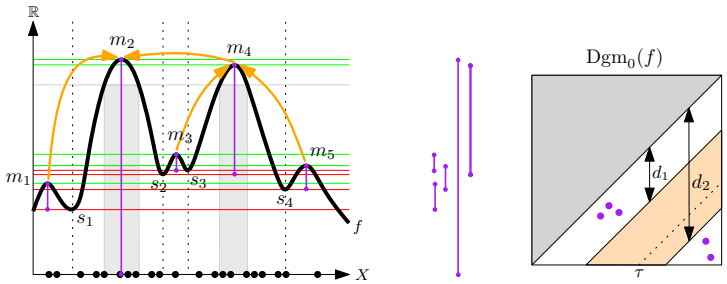

Figure 1: Example of ToMATo clustering induced by a density estimator f defined on a space $\hat{X}_n \subset \mathbb{R}$. The different local maxima m_i are connected together (with orange arrows) at the saddles s_i, and prominences are displayed as purple bars. The corresponding PD is displayed on the right: every mode is associated to a PD point, and the PD is (d_1, d_2)-separated. Choosing a threshold $d_1 \leq \tau \leq d_2$ induces a final clustering with two clusters.

Notations. Given a dataset $\hat{X}_n$, a function $f : \hat{X}_n \to \mathbb{R}$, a graph G, and a prominence threshold $\tau > 0$, we let $\mathcal{C}_{\text{ToMATo}} : \hat{X}_n \to [\![1, N]\!]$ (where N is the number of clusters) be the corresponding clustering $\mathcal{C}_{\text{ToMATo}} = \text{ToMATo}(\hat{X}_n, f, G, \tau)$.

Single-Parameter PH. The persistence diagram (PD) introduced in the ToMATo algorithm is the main descriptor of *persistent homology* (PH), which studies the persistence of topological features (encoded in homology groups) in filtrations of topological spaces. Several robustness properties have been established for PDs, and directly apply to ToMATo. They are stated using *diagram distances* (including the so-called *bottleneck distance* d_b), which are similar to optimal transport distances, as they are obtained by computing the cost of matching PD points together (and to the diagonal if PDs have different numbers of points). The exact definition of such distances is not necessary for the main exposition, and we provide it in Definition 10 in Appendix A.1.

The robustness property of ToMATo requires that the PD points are sufficiently separated:

Definition 1 ([8, Definition 4.1]). *Let $0 \leq d_1 < d_2$. The PD $\mathrm{Dgm}_0(f)$ is called (d_1, d_2)-*separated *if its points have prominences either less than d_1 or more than d_2. See Figure1.*

Theorem 1 ([8, Theorem 10.1], [25, Theorem 84]). *Let $0 \leq d_1 < d_2$. Let $\hat{X}_n$ be a dataset, G a graph whose node set is $\hat{X}_n$, and $f, g : \hat{X}_n \to \mathbb{R}$ be two functions s.t. $\mathrm{Dgm}_0(f)$ is (d_1, d_2)-separated. Assume $\varepsilon := \|f - g\|_\infty \leq \frac{d_2 - d_1}{16}$. Then, $d_b(\mathrm{Dgm}_0(f), \mathrm{Dgm}_0(g)) \leq \varepsilon$.*

*Moreover, let $\mathcal{C}_f := \text{ToMATo}(\hat{X}_n, f, G, \tau)$ and $\mathcal{C}_g := \text{ToMATo}(\hat{X}_n, g, G, \tau')$, with $d_1 + 4\varepsilon < \tau < d_2 - 4\varepsilon$ and $|\tau - \tau'| \leq 2\varepsilon$. Then, $\mathcal{C}_f$ and $\mathcal{C}_g$ are mutually 4ε-*related*: there exists a bijection*

$b : \mathrm{im}(\mathcal{C}_f) \to \mathrm{im}(\mathcal{C}_g)$ *s.t., for every cluster* $C_i = \mathcal{C}_f^{-1}(i)$, $i \in \mathrm{im}(\mathcal{C}_f)$, *with associated PD point* $p_i = (b_i, d_i) \in \mathrm{Dgm}_0(f)$, *one has* $C_i \cap f^{-1}([d_i + 4\varepsilon, +\infty[) \subseteq \mathcal{C}_g^{-1}(b(i))$ *(and vice-versa).*

See Appendix C for a proof.

Note that while TOMATO was described for a (finite) dataset $\hat{X}_n$, it can be applied straightforwardly (and PDs as well) to a continuous space X and function $f : X \to \mathbb{R}$ (in this case, there is no need to provide a graph G) and Theorem 1 still holds. We let $\text{TOMATO}(X, f, \tau)$ denote the corresponding clustering.

Multi-parameter PH. It is known that generalizing single-parameter PH (i.e., PH with a single function) to the setting where *several* functions $f_1, \ldots, f_p$, $p > 1$, are provided is very difficult, as there is no analogue of PDs for multi-parameter PH. Hence, several approaches have turned to *slicing*, i.e., to returning to single-parameter PH using *diagonal lines*.

Definition 2. *Let X be a topological space. Let $f_1, \ldots, f_p \to \mathbb{R}$ be continuous functions. Let ℓ be a diagonal line in $\mathbb{R}^p$, i.e., a line with direction $[1, \ldots, 1]$, and let $\varphi : \mathbb{R} \to \ell$ be a linear parametrization of ℓ. The function $F_\ell(f_1, \ldots, f_p) : X \to \mathbb{R}$ is defined as the function satisfying: $F_\ell(f_1, \ldots, f_p)^{-1}([t, +\infty[) = \bigcap_{i=1}^p f_i^{-1}([[\varphi(t)]_i, +\infty[)$[2]. Moreover, given $p_i = (b_i, d_i) \in \mathrm{Dgm}_0(F_\ell(f_1, \ldots, f_p))$, we let the* bar $\mathcal{B}(p_i)$ *denote the segment in $\mathbb{R}^p$ with endpoints $\varphi(b_i), \varphi(d_i)$.*

Note that the previous definition also holds in the discrete setting (dataset $\hat{X}_n$, graph G).

Using bars from functions obtained from diagonal lines allows to build the so-called *MMA decompositions*. These decompositions aim at mimicking PDs for multi-parameter PH: they are comprised of shapes in $\mathbb{R}^p$, each shape being interpreted as the lifetime of a connected component.

Definition 3. *Let X be a topological space. Let $f_1, \ldots, f_p : X \to \mathbb{R}$ be continuous functions. Let $\mathcal{L}$ be an ordered family of diagonal lines $\mathcal{L} = \{\ell_1, \ldots, \ell_{|\mathcal{L}|}\}$. Moreover, let m_{ℓ_i} be a* matching *function, i.e., an injective function between a subset of $\mathrm{Dgm}_0(F_{\ell_i}(f_1, \ldots, f_p))$ and $\mathrm{Dgm}_0(F_{\ell_{i+1}}(f_1, \ldots, f_p))$. The* MMA decomposition *is defined as:*

$$\mathrm{MMA}_0(f_1, \ldots, f_p; \mathcal{L}, \{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}}) := \{\mathcal{I}(S)\}_{S \in \mathcal{D}(\{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}})}, \tag{1}$$

where $\mathcal{D}(\{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}})$ denotes the set of sets of consecutive bars matched together under the matching functions, that is, each S is of the form $S := \{b_{i_1}, \ldots, b_{i_{|S|}}\}$ where $b_{i_j} = \mathcal{B}(p)$ for some $p \in \mathrm{Dgm}_0(F_{\ell_{i_j}}(f_1, \ldots, f_p))$ and $b_{i_{j+1}} = \mathcal{B}(\mathrm{m}_{\ell_{i_j}}(p))$ for all $1 \leq j \leq |S| - 1$, and where $\mathcal{I}(S)$ denotes an interval[3] with smallest volume that one can form using the endpoints of the bars in S.

Note that MMA decompositions can also be computed in the discrete setting (dataset $\hat{X}_n$, graph G). See Figure 2. Note also that, given some diagonal line ℓ that does not necessarily belongs to the family $\mathcal{L}$ used to compute an MMA decomposition $\mathrm{MMA} = \{\mathcal{I}_i\}_{1 \leq i \leq N}$, there is a corresponding *induced PD* $\mathrm{MMA}|_\ell := \{\mathcal{I}_i \cap \ell\}_{1 \leq i \leq N}$. By construction, on lines belonging to $\mathcal{L}$, induced PDs coincide with sliced PDs: $\mathrm{MMA}|_\ell = \{\mathcal{B}(p)\}_{p \in \mathrm{Dgm}(F_\ell(f_1, \ldots, f_p))}, \forall \ell \in \mathcal{L}$.

3 TOMATOMP: Algorithm and Guarantees

In this section, we introduce **TOMATOMP**, the first multi-parameter topological clustering method based on MMA decompositions. We first provide our algorithm in Section 3.1, and then state our theoretical guarantees in Section 3.2.

3.1 The TOMATOMP algorithm

The main idea is quite simple: start with a family of diagonal lines $\mathcal{L}$, compute the TOMATO clusterings on every line, relate the TOMATO clusters of different lines together using the intervals appearing in the MMA decomposition (so that all TOMATO clusters are now indexed by the MMA intervals), and, for every data point x, assign it to the most frequent TOMATO cluster that it belongs to across the lines. See Algorithm 1. Its complexity is driven by the one of computing MMA decompositions, which is cubic in the number of graph nodes and edges.

[2] Recall that $[\cdot]_i$ denotes the i-th coordinate in $\mathbb{R}^p$.

[3] Recall that an interval in $\mathbb{R}^p$ is a connected union of hyperrectangles; see Definition 8 in Appendix A.1.

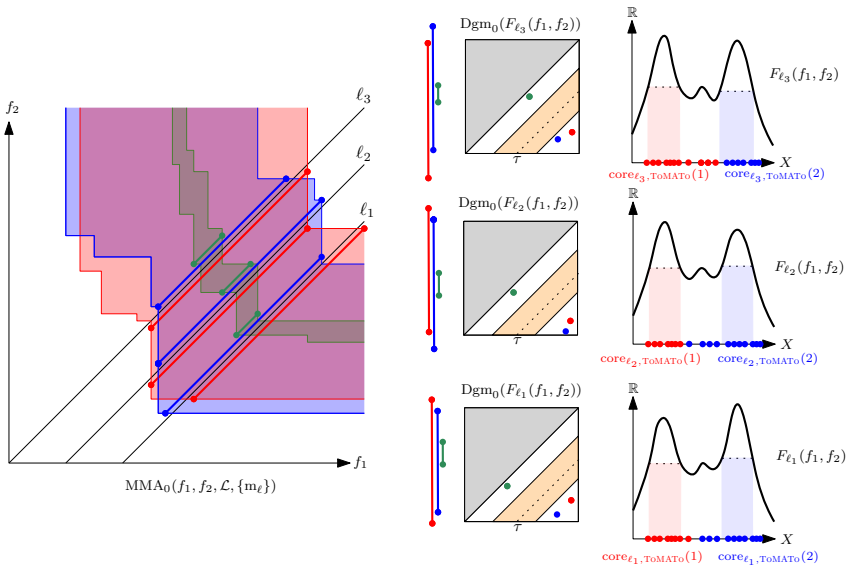

Figure 2: **(Left)** MMA decomposition with induced PDs of three diagonal lines, displayed both with points in $\mathbb{R}^2$ and bars displayed along the lines. **(Right)** Each PD (obtained by running ToMATo with $F_\ell(f_1, f_2)$) leads to a clustering of the dataset. The cores of the clusters (computed as the subparts of the clusters that are sufficiently above the saddles) remain stable across lines.

Data: $\hat{X}_n, f_1, \ldots, f_p, G, \tau, \mathcal{L}, \{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}}$
Result: $\mathcal{C}_{\text{ToMAToMP}}$
$\{\mathcal{I}_i\}_{1 \leq i \leq N} \leftarrow \mathrm{MMA}_0(f_1, \ldots, f_p; \mathcal{L}, \{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}})$;
for $\ell \in \mathcal{L}$ **do**
  $\mathcal{C}_{\ell,\text{ToMATo}} \leftarrow \text{ToMATo}(\hat{X}_n, F_\ell(f_1, \ldots, f_p), G, \tau)$;
  $\pi_\ell \leftarrow$ permutation s.t. $b_i \in \mathrm{MMA}|_\ell = \mathcal{B}(p_{\pi_\ell(i)}), p_{\pi_\ell(i)}) \in \mathrm{Dgm}_0(F_\ell(f_1, \ldots, f_p))$;
  for $x \in \hat{X}_n$ **do**
    $\mathcal{C}_{\ell,\text{MMA}}(x) \leftarrow \pi_\ell(\mathcal{C}_{\ell,\text{ToMATo}}(x))$;
  end
end
for $x \in \hat{X}_n$ **do**
  $\mathcal{C}_{\text{ToMAToMP}}(x) \leftarrow \mathrm{argmax}_{1 \leq i \leq N} |\{\ell \in \mathcal{L} : \mathcal{C}_{\ell,\text{MMA}}(x) = i\}|$;
end
return $\mathcal{C}_{\text{ToMAToMP}}$

Algorithm 1: The ToMAToMP algorithm

3.2 Theoretical Robustness of ToMAToMP

We now state and prove the main guarantee associated to ToMAToMP. In short, our main result states that, if the input has well-separated induced PDs (and thus clear distinction between noisy and relevant clusters for every line), then the clusters produced by ToMAToMP will be in bijection with the corresponding PD points of the induced PDs with large prominences. See Figure 2.

Theorem 2 (Robustness of ToMAToMP)**.** *Let X be a topological space. Let $f_1, \ldots, f_p : X \to \mathbb{R}$ be continuous functions, and let G be some graph defined on the data points. Let $\mathcal{L} = \{\ell_1, \ldots, \ell_{|\mathcal{L}|}\}$ be an ordered family of diagonal lines at distance $\eta > 0$, and let $\{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}}$ be a family of compatible matching functions*[4] *achieving the q-th diagram distances between consecutive PDs.*

(A1) Assume that $\exists 0 \leq d_1 < d_2$ s.t. $\forall \ell \in \mathcal{L}$, $\mathrm{Dgm}_0(F_\ell(f_1, \ldots, f_p))$ is (d_1, d_2)-separated.

[4]That is, the pairs of bars matched together by each m_ℓ have no strictly comparable endpoints: the start points s_1, s_2 corresponding to bars b_1, b_2 s.t. $b_2 = \mathrm{m}_\ell(b_1)$ do not satisfy $s_1 <_p s_2$ or $s_2 <_p s_1$ (and similarly for endpoints), where $x \leq_p x' \in \mathbb{R}^p$ iif $[x]_i \leq [x']_i$ for all $1 \leq i \leq p$.

(A2) *Let* $d^* := \frac{1}{2}\min\{\|p-p'\|_\infty : \exists \ell \in \mathcal{L} \text{ s.t. } p,p' \in \mathrm{Dgm}_0(F_\ell(f_1,\dots,f_p)), \mathrm{prom}(p) \geq d_2, \mathrm{prom}(p') \geq d_2\}$. *Assume* $\eta \leq \min\{\frac{d_2-d_1}{16}, d^*\}$.

Then:

- (i) *the matching functions* $\{\mathrm{m}_\ell\}_{\ell\in\mathcal{L}}$ *are unique and restrict to bijections between points of consecutive PDs with prominences at least* d_2,
- (ii) *for any* $d_1 + 4\eta < \tau < d_2 - 4\eta$, *the number of clusters* N *computed by* $\mathcal{C}_{\text{ToMAToMP}} := \text{ToMAToMP}(\hat{X}_n, f_1, \dots, f_p, G, \tau, \mathcal{L}, \{\mathrm{m}_\ell\}_{\ell\in\mathcal{L}})$ *is equal to the number* N *of points of* $\mathrm{Dgm}_0(F_\ell(f_1,\dots,f_p))$ *with prominences at least* d_2[5], *and*
- (iii) *Given a cluster index* $1 \leq i \leq N$, *let* $S_i := \{b_i^\ell\}_{\ell\in\mathcal{L}}$ *denote the set of consecutive bars matched together by the matching functions* m_ℓ. *Then, letting* $\mathcal{C}_{\ell,\text{ToMATo}} := \text{ToMATo}(\hat{X}_n, F_\ell(f_1,\dots,f_p), G, \tau)$, *the family of clusterings* $\{\mathcal{C}_{\ell,\text{ToMATo}}\}_{\ell\in\mathcal{L}}$ *is* 4η*-related to* $\mathcal{C}_{\text{ToMAToMP}}$, *i.e., there exists a cluster* $C_i = \mathcal{C}^{-1}_{\text{ToMAToMP}}(i)$ *s.t.* $\bigcap_{\ell\in\mathcal{L}} \mathrm{core}_{\ell,\text{ToMATo}}(i) \subseteq C_i$, *where the* core *is defined as* $\mathrm{core}_{\ell,\text{ToMATo}}(i) := \mathcal{C}^{-1}_{\ell,\text{ToMATo}}(\pi_\ell(i)) \cap F_\ell(f_1,\dots,f_p)^{-1}([\varphi_\ell^{-1}(b_i^\ell) + 4\eta, +\infty[)$, *where each* φ_ℓ *is a linear parametrization of* ℓ. *The set* $\bigcap_{\ell\in\mathcal{L}} \mathrm{core}_{\ell,\text{ToMATo}}(i)$ *is called the* 4η-core *of* C_i.

See Figure 2 for an illustration of the notations and statements, and Appendix D for the proof and a few extensions. One may wonder whether $\bigcap_{\ell\in\mathcal{L}} \mathrm{core}_{\ell,\text{ToMATo}}(i) = \emptyset$, making (iii) trivial. Note that choosing η sufficiently small while keeping $|\mathcal{L}|$ upper bounded guarantees (using Theorem 8 in Appendix C) that the intersection is not empty, as PD points can move at most η from a line to another. The local maxima of modes must thus stay in the cores if there are not too many lines, as their function values can decrease by at most $|\mathcal{L}| \cdot \eta$ across the whole sweep.

3.3 Applications to ToMATo

In this section, we use our main result Theorem 2 to break the two aforementioned limitations of ToMATo, i.e., the need for the user to provide and/or fine-tune an input graph G, and the high sensitivity to outliers in function values. See Appendix D for more details.

Neighborhood graph tuning. The most frequent graphs used by ToMATo are δ-neighborhood graphs. As proved in a recent result [1, Theorem 4.6 (iii)], it turns out that letting δ increase across a specific range of values, combined with an input function f, allows to recover the ToMATo clusters associated to f without the need to pick a specific δ using ToMAToMP.

Corollary 1. *Let* $\hat{X}_n$ *be a geodesic* ε*-sample of a compact geodesic space* X *with convexity radius* $\rho(X)$ *s.t.* $\varepsilon \leq \rho(X)/4$, *let* $\delta_{\max} \in [4\varepsilon, \rho(X)[$, *and let* $G := G_{\delta_{\max}}$ *be the* $\delta_{\max}$*-neighborhood graph built on* $\hat{X}_n$. *Now, let* $f : X \to \mathbb{R}$ *be a* c*-Lipschitz function s.t.* $\mathrm{Dgm}_0(f)$ *is* (d_1, d_2)*-separated with* $\delta_{\max} < \frac{d_2-d_1}{16c}$. *Finally, let* $\tilde{G}$ *be the first barycentric subdivision of* G[6], *let* $\tilde{f} : V(\tilde{G}) \to \mathbb{R}$ *be the restriction of* f *to* $V(\tilde{G})$, *and let* $g : V(\tilde{G}) \to \mathbb{R}$ *be the function assigning* 0 *to vertices in* $V(G)$ *and the opposite of the lengths of the corresponding edges of* G *to vertices in* $V(\tilde{G}) \setminus V(G)$[7]. *Let* $\mathcal{L}$ *be an ordered family of diagonal lines at distance* $\eta \leq \frac{d_2-d_1-4c\delta_{\max}}{16}$, $d_1 + 4(c\delta_{\max} + \eta) \leq \tau \leq d_2 - 4(c\delta_{\max} + \eta)$, *and* $\{\mathrm{m}_\ell\}_{\ell\in\mathcal{L}}$ *as in Theorem 2. Then,* $\mathcal{C}_{\text{ToMAToMP}} := \mathcal{C}_{\text{ToMAToMP}}(\hat{X}_n, g, \tilde{f}, \tilde{G}, \tau, \mathcal{L}, \{\mathrm{m}_\ell\}_{\ell\in\mathcal{L}})$ *is* $4(c\delta_{\max} + \eta)$*-related to* $\mathcal{C}_{\text{ToMATo}} := \text{ToMATo}(X, f, \tau)$[8], *as long as a majority of lines in* $\mathcal{L}$ *crosses the rectangle* $[-\delta_{\max}, -4\varepsilon] \times \mathrm{im}(f)$ *vertically, i.e.,* $\pi_2(\ell \cap [-\delta_{\max}, -4\varepsilon] \times \mathrm{im}(f)) = \mathrm{im}(f)$ *for most lines* $\ell \in \mathcal{L}$.

In practice, rescaling the input functions can help ensuring that diagonal lines are located within the corresponding slabs, at the price of potentially creating less robust filtrations.

[5] This number N is the same for all $\ell \in \mathcal{L}$ thanks to (i).

[6] That is, the graph obtained by splitting every edge of G into two new edges sharing a new node, whose function value is the mean of the function values of the other endpoints.

[7] I.e., the superlevel sets of g are equivalent to the *Vietoris-Rips* filtration on $\hat{X}_n$—see Section A.2

[8] I.e., each $4(c\delta_{\max} + \varepsilon)$-core of a cluster of $\mathcal{C}_{\text{ToMAToMP}}$ is included in some cluster of $\mathcal{C}_{\text{ToMATo}}$.

Outlier robustness. Another application of Theorem 2 is the setting where input functions are plagued with outlier values. More specifically, if one is given some function $\tilde{f}$ that differs from f on a few points, but such that the difference between f and $\tilde{f}$ is very large on these points, the modes of f and $\tilde{f}$ might potentially be very different. In this case, using an outlier detection score as a separate function and combine it with $\tilde{f}$ allows ToMAToMP to recover the ToMATo clusters associated to f, provided that the points with outlier function values are *topologically robust*.

Definition 4. *Let $\hat{X}_n$ be a dataset, and G_δ be a δ-neighborhood graph built on top of $\hat{X}_n$. A point $x \in \hat{X}_n$ is called* topologically robust *if it has at least one neighbor $n(x)$ in G that is connected to all of its other neighbors.*

Corollary 2. *Let $\hat{X}_n$ be a geodesic ε-sample of a compact geodesic space X with convexity radius $\rho(X)$ s.t. $\varepsilon \leq \rho(X)/4$, let $\delta \in [4\varepsilon, \rho(X)[$, and let $G := G_\delta$ be the δ-neighborhood graph built on $\hat{X}_n$. Now, let $f : X \to \mathbb{R}$ be a c-Lipschitz function s.t. $\mathrm{Dgm}_0(f)$ is (d_1, d_2)-separated with $\delta < \frac{d_2 - d_1}{16c}$. Moreover, assume one is given another function $\tilde{f}$ s.t. there exists a set $\mathcal{O} \subset V(G)$ of non-overlapping[9], topologically robust points s.t. $f(x) = \tilde{f}(x)$ for $x \in V(G) \setminus \mathcal{O}$. Assume one is given an outlier score function $g : \hat{X}_n \to \mathbb{R}$ s.t. $K'_g \leq K_g$, where $K_g := \min g|_{\hat{X}_n \setminus \mathcal{O}}$, $K'_g := \max g|_{\mathcal{O}}$. Let $\mathcal{L}$ be an ordered family of diagonal lines at distance $\eta \leq \frac{d_2 - d_1 - 4c\delta}{16}$, $d_1 + 4(c\delta + \eta) \leq \tau \leq d_2 - 4(c\delta + \eta)$, and $\{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}}$ as in Theorem 2. Then, $\mathcal{C}_{\text{ToMAToMP}} := \mathcal{C}_{\text{ToMAToMP}}(\hat{X}_n, g, \tilde{f}|_{\hat{X}_n}, G, \tau, \mathcal{L}, \{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}})$ is $4(c\delta + \eta)$-related to $\mathcal{C}_{\text{ToMATo}} := \text{ToMATo}(X, f, \tau)|_{\hat{X}_n \setminus \mathcal{O}}$, as long as a majority of lines in $\mathcal{L}$ crosses the rectangle $[K'_g, K_g] \times \mathrm{im}(f)$ vertically.*

In practice, when outliers are not known a priori, one can always ensure that the assumptions of Corollary 2 are satisfied by manually adding edges to G_δ for all points whose function values are above a given threshold, which can typically be set as a quantile of the function value distribution. Note however that this also turn G_δ into another graph that is not necessarily a neighborhood graph.

4 Experiments

In this section, we provide numerical experiments with ToMAToMP. We study four use cases and datasets for topological clustering: (i) a synthetic dataset in $\mathbb{R}^2$ filtered with the Distance-To-Measure (DTM) [6] ($n = 1,000$ points) called `synthetic`, (ii) a dataset of 57 3D shapes from the Princeton database[10] (the first 3 of 19 categories) filtered with the Heat Kernel Signature ($t = 1000$) [27] ($n \sim 100$–$2,500$ points), called `3dshape`, (iii) a dataset of 20 morphological images representing cells coming from frames 20–40 of a video in the Bio-image Analysis Notebooks suite[11] filtered with image pixel values ($n = 70 \times 70$ pixels), called `image`, and (iv) two spatial transcriptomics datasets, `kpmp` and `spat` ($n = 645$ points and $n = 977$ points respectively) from [4] and [2], containing cells in two pieces of tissue, and filtered with the expression of several genes. See Figure 3.

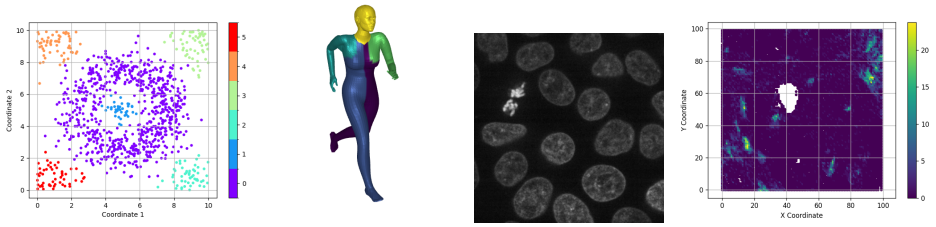

Figure 3: Datasets used in experiments. From left to right: `synthetic`, a mesh from `3dshape`, an example of morphological image from `image`, and one spatial transcriptomics datasets (`spat`).

[9] I.e., the sets of neighbors are disjoint.

[10] `https://segeval.cs.princeton.edu/`

[11] `https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/intro.html`

Tasks and scores. On datasets (i)–(iii), the goal is to recover the right number of clusters, and Adjusted Rand Index (ARI) and Average Mutual Information (AMI) are used as quality scores. On dataset (iv), the goal is to rank (pairs of) genes based on how many clusters they induce in data, a quantity called *coefficient of spatial structure* (CoSS) [4]. For single genes, this coefficient is measured as the squared sum of the prominences of TOMATO clusters induced by gene expression, while for pairs of genes, we use the average Jaccard similarities between the TOMATO clusters of both genes, following a known method for detecting co-localization [2]. Concerning single genes (1-gr), we rank the top 50 genes with largest variances, while for pairs of genes (2-gr), we rank all pairs from the top 10 genes with largest variances (i.e., 45 pairs). Two scores are used to measure a ranking quality against ground-truth ranking[12]: Pearson correlation and TopHits@10, which measures the fraction of top 10 hits that are also top 10 in the ground-truth ranking.

Concerning the baselines, we compare TOMATOMP against single-parameter topological clustering with TOMATO, as well as a range of standard clustering baselines. Note that some of such baselines (Louvain, Leiden) expect a graph input, in which case we either provide a neighborhood graph with highest δ from a given range (for the `no-radius` experiment, see below) or a manually, fine-tuned neighborhood graph (for the `outlier` experiment).

Computation details. All results are presented for a family of $|\mathcal{L}| = 100$ lines, evenly spaced within the function ranges (note that these ranges are rescaled if necessary in order to satisfy the assumptions of Corollaries 1 and 2). Matching functions are computed with the vineyards algorithm [11] (as they achieve the q-th diagram distances for sufficiently close lines), and the prominence threshold is either set so as to keep only the relevant number of clusters (when this number is known a priori, as in datasets (i)–(iii)), or to $\tau = 0$ (in datasets (iv)). Finally, for application (iv), we design a multi-parameter criterion for ranking (pairs of) genes by computing a quantile ($q = 10\%$) of the distribution of the (1-parameter) CoSS's obtained from the induced PDs produced by TOMATOMP. As for the non-topological baselines, we focus on hierarchical clustering, and compute the opposite of the squared sum of distances in the dendrogram (using both cell coordinates and gene expression to compute the distances) as ranking criterion. In Appendix E.2, we also provide plots showing the quality scores and running times dependence on the number of lines used in TOMATOMP. Our code is based on `multipers` [21], `gudhi` (`https://gudhi.inria.fr/python/latest/`) and `scikit-learn` [24], and was run on a cluster with Intel Xeon E5-2620 v3 CPUs with 6 cores and 128 GB RAM.

Neighborhood graph tuning. In this first set of experiments, we measure the quality of clustering and (pairs of) gene ranking when using TOMATOMP without explicitly specifying the δ parameter in the neighborhood graph G_δ, as an illustration of Corollary 1. These experiments are referred to as `no-radius`. More specifically, we compute TOMATOMP clusters with a range of δ ranging from either the 1% to the 5% quantiles of the positive data distance distributions (datasets (i)–(iii)), or to ranges of lengths 10^{-2} (`kpmp`) and 5×10^{-1} (`spat`) around the 1% quantiles. We compare against the mean score of TOMATO clusterings obtained after testing every line, as well as the worst score across lines. This second baseline is very pessimistic, and only illustrates how poor TOMATO clusters can be for unlucky line picks.

AMI scores can be found in the two first columns of Tables 1, 2 and 3 (ARI scores, correlations and running times can be found in Appendix E.1). As one can see from Tables 1 and 2, TOMATOMP can achieve comparable and sometimes superior results against TOMATO without any graph tuning, and against non-topological baselines as well (launched with default parameters using Scikit-Learn implementations), which sometimes struggle to capture the cluster structures. Importantly, concerning TOMATO, a wrong choice of δ can lead to very poor clusters, and even averaging the results is not as efficient as TOMATOMP, as the mean score is dragged down by those poor choices.

The TopHits@10 scores of Table 3 are not as good. The highest ranked pairs have lower intersection with the ground-truth ones for `kpmp`, but still manage to recover 6 pairs out of the best 10. On the other hand, dendrogram distances computed on cell coordinates and gene expression struggle more at capturing relevant pairs.

[12]For both datasets in (iv), the ground-truth ranking is obtained by ordering (pairs of) genes with the CoSS obtained after running TOMATO without outlier values and using a fine-tuned δ parameter for the neighborhood graph. Hence, as the ground-truth is intrinsically topological, we expect non-topological baselines to be less efficient.

Method	synthetic no-radius	3dshape no-radius	synthetic outlier	image outlier
KMeans	0.43	0.41 ± 0.20	0.38 ± 0.03	0.69 ± 0.00
SpectralClustering	0.69	0.38 ± 0.18	0.67 ± 0.00	0.69 ± 0.00
AgglomerativeClustering	0.45	**0.42 ± 0.19**	0.43 ± 0.03	0.67 ± 0.00
DBSCAN	0.00	0.03 ± 0.11	0.00 ± 0.00	0.00 ± 0.00
DensityPeaks	0.53	0.34 ± 0.18	0.53 ± 0.01	0.63 ± 0.00
HDBSCAN	0.75	0.25 ± 0.13	0.72 ± 0.02	0.12 ± 0.01
Louvain	0.57	0.41 ± 0.15	0.56 ± 0.02	0.67 ± 0.00
Leiden	0.56	0.41 ± 0.16	0.56 ± 0.02	0.68 ± 0.00
ToMATo (min)	0.00	0.26 ± 0.21	–	–
ToMATo (mean)	0.66	0.38 ± 0.19	–	–
ToMATo	–	–	0.59 ± 0.12	0.55 ± 0.02
ToMAToMP	**0.82**	0.35 ± 0.20	**0.93 ± 0.01**	**0.74 ± 0.00**

Table 1: AMI scores for `no-radius` and `outlier` experiments. Results are averaged over all meshes (resp. frames) of the `3dshape` (resp. `image`) dataset.

Method	kpmp no-radius	spat no-radius	kpmp outlier	spat outlier
AgglomerativeClustering	0.30	0.10	0.44 ± 0.10	0.70 ± 0.03
ToMATo (min)	0.90	0.90	–	–
ToMATo (mean)	0.99	0.97	–	–
ToMATo	–	–	0.49 ± 0.08	0.86 ± 0.09
ToMAToMP	**1.00**	**1.00**	**0.99 ± 0.02**	**0.97 ± 0.04**

Table 2: Top@10 scores for `no-radius` and `outlier` experiments in the single gene ranking (1-gr).

It should be noted however, that the running times of ToMAToMP are substantially longer: the fact that the algorithm requires barycentric subdivisions of large graphs as inputs (in order to turn Vietoris-Rips filtrations into superlevel set filtrations) leads to running times that are several orders of magnitude longer than both non-topological and single-parameter topological baselines. See also Figure 7 in Appendix E.2.

Outlier robustness. In this second set of experiments, we measure the quality of clustering and ranking when using ToMAToMP with an outlier detection score function (computed as the average absolute difference between the function value of a data point and the ones of its neighbors in G) after datasets have been randomly plagued with a small fraction of outlier values (10% for datasets (i)–(iii) and 10 random cells for dataset (iv), with results averaged across 50 trials), as an illustration of Corollary 2. These experiments are referred to as `outlier`.

AMI scores can be found in the two last columns of Tables 1, 2 and 3 (ARI scores, correlations and running times can be found in Appendix E.1). As one can see from these tables, non-topological and topological baselines definitely struggle in the face of outlier values as such values are immediately detected as persistent modes (see also Figure 8 in Appendix E.3). On the other hand, ToMAToMP clusterings remain highly accurate. As for rankings, Pearson correlation of ToMAToMP is again low for gene pairs but top hits remain competitive: while the rankings of ToMATo drop to 2.1 detected relevant pairs out of 10 in average for `kpmp`, ToMAToMP rankings stay (in average) around 5.7, despite being lowly correlated to the ground-truth CoSS.

As in the previous section, running times are still much longer for ToMAToMP, but as barycentric subdivisions are not required anymore (since Vietoris-Rips filtrations are not used), the running times are reduced by around 1 order of magnitude. See also Figure 7 in Appendix E.2.

Multiple functions. Finally, as a qualitative illustration of the high flexibility of ToMAToMP, we end this numerical experiment section by performing rankings of *triplets of genes*, using 3 different gene expression functions at the same time, and computing approximate decompositions of the corresponding 3-parameter persistence module. One can see in Figure 4 that for the `spat` dataset, the top ranked triplets do characterize genes that co-localize in the tissue, while lower ranked triplets

Method	kpmp no-radius	spat no-radius	kpmp outlier	spat outlier
AgglomerativeClustering	0.50	0.60	0.24 $\pm$ 0.13	0.63 $\pm$ 0.07
ToMATo (min)	0.70	0.80	–	–
ToMATo (mean)	**0.93**	**0.95**	–	–
ToMATo	–	–	0.21 $\pm$ 0.17	0.79 $\pm$ 0.11
ToMAToMP	0.60	0.80	**0.57** $\pm$ **0.06**	**0.80** $\pm$ **0.01**

Table 3: Top@10 scores for `no-radius` and `outlier` experiments in the gene pair ranking (2-gr).

have at least one gene in the triplet that is not highly expressed where the other two are. See Figure 9 in Appendix E.3 for similar examples with the `kpmp` dataset.

Investigating these rankings and interpreting them biologically is an important future work.

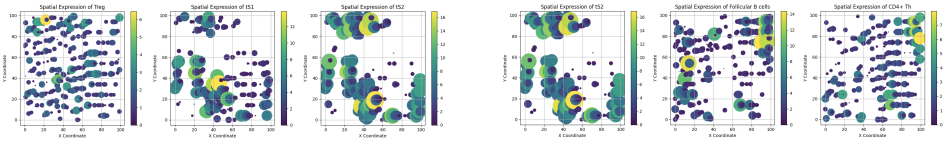

Figure 4: **(Left)** Highly ranked triplet of genes of the spat dataset. All three genes co-localize on the top-left and middle-down tissue regions. **(Right)** Lower ranked triplet.

5 Conclusion

In this article, we introduced ToMAToMP, the first topological clustering method able to handle several functions at the same time. We showed that, similar to the standard ToMATo algorithm, ToMAToMP enjoys robustness guarantees, and that it compares favorably to both topological and non-topological baselines on various datasets. Several directions remain open for future work:

- The long running times of ToMAToMP clearly leave room for improvement. We believe that leveraging recent, fast optimizations of persistent homology computations could help reducing this limitation [12].
- Developing a systematic procedure for checking the validity of the assumptions in Theorem 2 and Corollaries 1 and 2 would be very useful. This includes, e.g., developing bimodality tests on the PD point prominences in the distribution of induced PDs, or providing a canonical method for rescaling filter functions.
- Relaxing assumption $(A1)$ of Theorem 2 would be a considerable theoretical improvement, as this assumption is going to be harder to satisfy as the number of functions increases. We believe this could be achieved upon careful study of the matching functions $\{\mathrm{m}_\ell\}_{\ell\in\mathcal{L}}$.
- The study of co-localizing gene tuples is promising and will be further developed in future work, as we believe that ToMAToMP might help for the joint analysis of multiple genes without going through expensive pairwise comparisons. Applications to tissue segmentation is also another potential avenue for ToMAToMP in spatial transcriptomics.

Reproducibility

A public repository with code and run instructions is available at `https://github.com/MathieuCarriere/tomatomp`.

Acknowledgments and Disclosure of Funding

M.C. is supported by l'Agence Nationale de la Recherche (ANR) under grants TopModel ANR-23-CE23-0014. MC is also supported by 3IA ANR-23-IACL-0001 and the PERSNET project, part of

the PEPR IA programme. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

References

[1] Ethan André, Jingyi Li, David Loiseaux, and Steve Oudot. Estimating the Persistent Homology of Rn-valued Functions Using Function-Geometric Multifiltrations. *Journal of Applied and Computational Topology (JACT)*, 10(4):25, September 2026.

[2] Sungwoo Bae, Hyekyoung Lee, Kwon Joong Na, Dong Soo Lee, Hongyoon Choi, and Young Tae Kim. STopover Captures Spatial Colocalization and Interaction in the Tumor Microenvironment using Topological Analysis in Spatial Transcriptomics Data. *Genome Medicine*, 17(1):33, April 2025.

[3] Magnus Bakke Botnan and Michael Lesnick. An Introduction to Multiparameter Persistence. In Aslak Bakke Buan, Henning Krause, and Øyvind Solberg, editors, *Representations of Algebras and Related Structures*, pages 77–150. European Mathematical Society, November 2023.

[4] James Boyle, Gregory Hamm, Eleanor Williams, Robin JG Hartman, Magnus Söderburg, Ian Henry, and Michael Casey. Topological Data Analysis for Unsupervised Feature Selection in Large Scale Spatial Omics Data Sets. *Bulletin of Mathematical Biology*, 88(4):52, March 2026.

[5] Gunnar Carlsson and Facundo Mémoli. Characterization, Stability and Convergence of Hierarchical Clustering Methods. *Journal of Machine Learning Research (JMLR)*, 11(47):1425–1470, 2010.

[6] Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric Inference for Probability Measures. *Foundations of Computational Mathematics (FoCM)*, 11(6):733–751, December 2011.

[7] Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. SpringerBriefs in Mathematics. Springer International Publishing, 2016.

[8] Frédéric Chazal, Leonidas J. Guibas, Steve Oudot, and Primoz Skraba. Persistence-Based Clustering in Riemannian Manifolds. *Journal of the ACM*, 60(6):41:1–41:38, November 2013.

[9] Yizong Cheng. Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, August 1995.

[10] Yu-Min Chung, Chuan-Shen Hu, Emily Sun, and Henry C. Tseng. Morphological Multiparameter Filtration and Persistent Homology in Mitochondrial Image Analysis. *PLOS ONE*, 19(9):e0310157, September 2024.

[11] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and Vineyards by Updating Persistence in Linear Time. In *Proceedings of the 22nd Annual Symposium on Computational Geometry (SoCG 2006)*, pages 119–126. Association for Computing Machinery, June 2006.

[12] Tamal K. Dey and Tao Hou. Computing Zigzag Vineyard Efficiently Including Expansions and Contractions. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry (SoCG 2024)*, volume 293 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[13] Tamal K. Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2022.

[14] Herbert Edelsbrunner and John L. Harer. *Computational Topology: An Introduction*. American Mathematical Society, January 2010.

[15] Barbara Giunti and Jānis Lazovskis. Pruning Vineyards: Updating Barcodes and Representative Cycles by Removing Simplices. *Foundations of Data Science*, 11(0):207–232, 2026.

[16] Marius Huber, Sara Kališnik, and Patrick Schnider. AuToMATo: An Out-Of-The-Box Persistence-Based Clustering Algorithm. (arXiv:2408.06958), October 2024.

[17] Woojin Kim and Facundo Mémoli. Extracting Persistent Clusters in Dynamic Data via Möbius Inversion. *Discrete & Computational Geometry (DCG)*, 71(4):1276–1342, October 2023.

[18] W.L.G. Koontz, P.M. Narendra, and K. Fukunaga. A Graph-Theoretic Approach to Nonparametric Cluster Analysis. *IEEE Transactions on Computers*, C-25(9):936–944, September 1976.

[19] Claudia Landi. The Rank Invariant Stability via Interleavings. In Erin W. Chambers, Brittany Terese Fasy, and Lori Ziegelmeier, editors, *Research in Computational Topology*, volume 13 of *Association for Women in Mathematics*, pages 1–10. Springer International Publishing, 2018.

[20] David Loiseaux, Mathieu Carrière, and Andrew J. Blumberg. Multi-Parameter Module Approximation: An Efficient and Interpretable Invariant for Multi-Parameter Persistence Modules with Guarantees. *Journal of Applied and Computational Topology (JACT)*, 9(4):26, October 2025.

[21] David Loiseaux and Hannah Schreiber. Multipers: Multiparameter Persistence for Machine Learning. *Journal of Open Source Software (JOSS)*, 9(103):6773, November 2024.

[22] Dmitriy Morozov and Luis Scoccola. Computing Betti Tables and Minimal Presentations of Zero-Dimensional Persistent Homology. In Oswin Aichholzer and Haitao Wang, editors, *41st International Symposium on Computational Geometry (SoCG 2025)*, volume 332 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 69:1–69:15, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[23] Steve Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*, volume 209 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2015.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[25] Alexander Rolle and Luis Scoccola. Stable and Consistent Density-Based Clustering via Multiparameter Persistence. *Journal of Machine Learning Research (JMLR)*, 25(258):1–74, 2024.

[26] Luis Scoccola and Alexander Rolle. Persistable: Persistent and Stable Clustering. *Journal of Open Source Software (JOSS)*, 8(83):5022, March 2023.

[27] Primoz Skraba, Maks Ovsjanikov, Frédéric Chazal, and Leonidas J. Guibas. Persistence-Based Segmentation of Deformable Shapes. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2010)*, pages 45–52. IEEE, June 2010.

[28] Daniel J. Stein, Miles Tran, and Ilya Korsunsky. Accurate Tiling of Spatial Single-Cell Data with Tessera. page 2025.01.17.633630, 2025.

A Background on persistent homology and hierarchical clustering

A.1 Single- and multi-parameter persistent homology (PH)

Let X be a topological space. The main idea of single-parameter PH is to progressively track the *homology groups* of the *sublevel sets*[13] of a continuous function $f : X \to \mathbb{R}$. More precisely, let $F_\alpha = \{x \in X : f(x) \leq \alpha\}$ be the sublevel set of f at level α. Then, applying the homology functor H_* (computed with coefficients in a field $\mathbb{K}$) to the following growing sequence, that is also called the sublevel set *filtration* of X :

$$F_{\alpha_1} \subseteq F_{\alpha_2} \subseteq \cdots \subseteq F_{\alpha_n}, \quad \alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n, \tag{2}$$

induces the following sequence:

$$H_*(F_{\alpha_1}) \to H_*(F_{\alpha_2}) \to \cdots \to H_*(F_{\alpha_n}), \quad \alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n, \tag{3}$$

where the linear maps are induced by the inclusions. This can be straightforwardly extended to sequences indexed over $\mathbb{R}$ for defining *persistence modules*.

Definition 5 (Single-Parameter Persistence Module [7, Section 2.1])**.** *A* single-parameter persistence module M *is a family of vector spaces* $M = \{M_\alpha : \alpha \in \mathbb{R}\}$ *together with linear maps* $(M_{\alpha_1} \to M_{\alpha_2})$ *for any* $\alpha_1 \leq \alpha_2$*, such that:* (a) *for any* $\alpha_1 \leq \alpha_2 \leq \alpha_3$*, one has* $(M_{\alpha_2} \to M_{\alpha_3}) \circ (M_{\alpha_1} \to M_{\alpha_2}) = (M_{\alpha_1} \to M_{\alpha_3})$ *and* (b) *for any* $\alpha \in \mathbb{R}$*,* $(M_\alpha \to M_\alpha)$ *is the identity map.*

A nice property of persistence modules over $\mathbb{R}$ is that they decompose into direct sums of *interval modules*, which are then summarized into *persistence barcodes*:

Definition 6 (Single-Parameter Interval Module [7, Section 2.4])**.** *A single-parameter persistence module* $I = I_{(\alpha_b, \alpha_d)}$ *is an* interval module *if there exists an interval* (α_b, α_d) *(called the* support *of* I*) such that* $I_\alpha \simeq \mathbb{K}$ *if* $\alpha \in (\alpha_b, \alpha_d)$ *and* $I_\alpha \simeq 0$ *otherwise.*

Theorem 3 (Decomposition Theorem [7, Theorem 2.8])**.** *Let* M *be a single-parameter persistence module. If every* M_α *is finite-dimensional, then, there exist* $n \in \mathbb{N}$ *and interval modules* $I_1, \ldots, I_n$ *such that:*

$$M \simeq \oplus_{i=1}^n I_i. \tag{4}$$

The set of interval module supports in this decomposition is called the persistence barcode*, or* persistence diagram $\mathrm{Dgm}(M)$ *of* M*. When* M *is induced by the sublevel sets of a function* f*, we slightly abuse notations and denote the corresponding persistence diagram with* $\mathrm{Dgm}(f)$.

Persistence and interval modules can be generalized to several parameters straightforwardly:

Definition 7 (Multi-Parameter Persistence Module [3, Section 2.3])**.** *Let* $p \in \mathbb{N}^*$*. A* (p)-multi-parameter persistence module M *is a family of vector spaces* $M = \{M_\alpha : \alpha \in \mathbb{R}^p\}$ *together with linear maps* $(M_{\alpha_1} \to M_{\alpha_2})$ *for any* $\alpha_1 \leq_p \alpha_2$ *(where* $\leq_p$ *means regular* $\leq$ *on each coordinate) such that:* (a) *for any* $\alpha_1 \leq_p \alpha_2 \leq_p \alpha_3$*, one has* $(M_{\alpha_2} \to M_{\alpha_3}) \circ (M_{\alpha_1} \to M_{\alpha_2}) = (M_{\alpha_1} \to M_{\alpha_3})$ *and* (b) *for any* $\alpha \in \mathbb{R}^p$*,* $(M_\alpha \to M_\alpha)$ *is the identity map.*

Definition 8 (Multi-Parameter Interval Module [3, Definition 2.3])**.** *A* (p)*-multi-parameter persistence module* I *is an* interval module *if there exists an interval* $\mathrm{supp}(I) \subseteq \mathbb{R}^p$ *(called the* support *of* I*) such that* $I_\alpha \simeq \mathbb{K}$ *if* $\alpha \in \mathrm{supp}(I)$ *and* $I_\alpha \simeq 0$ *otherwise. Recall that an interval* I[14] *in* $\mathbb{R}^p$ *is a connected subset such that for any* $x, x' \in I$ *there exists at least one sequence* $x_1, \ldots, x_q \subset I$ *s.t.* $x \sim x_1 \sim \cdots \sim x_q \sim x'$ *where* $\sim$ *is either* $\leq_p$ *or* $\geq_p$*.*

Unfortunately, Theorem 3 does not hold anymore in the multi-parameter setting. However, modules can still be compared using a metric called the *interleaving distance*:

Definition 9 (Interleaving [3, Definition 6.1])**.** *Let* M, M' *be two multi-parameter persistence modules. Let* $\varepsilon > 0$ *and* $\vec{\varepsilon} = \varepsilon \cdot [1, \ldots, 1]^T \in \mathbb{R}^p$*. The modules* M *and* M' *are* ε*-interleaved if there exist maps* $\varphi_x : M_x \to M'_{x+\vec{\varepsilon}}$ *and* $\psi_x : M'_x \to M_{x+\vec{\varepsilon}}$ *such that* $\psi_{x+\vec{\varepsilon}} \circ \varphi_x = (M_x \to M_{x+2\vec{\varepsilon}})$ *and* $\varphi_{x+\vec{\varepsilon}} \circ \psi_x = (M'_x \to M'_{x+2\vec{\varepsilon}})$.

For single-parameter persistence modules, the interleaving distance is related to the *diagram distances* between persistence diagrams (PDs). These distances are defined by finding partial correspondences between PD points, and picking the one with lowest cost.

[13] Upon taking the opposite of f, this theory extends straighforwrdly to superlevel sets, as used in topological clustering.

[14] In the following, we slightly abuse notations and use I to denote either an interval module or its support.

Definition 10 (q-th diagram distance)**.** *Let* Dgm *and* Dgm' *be two PDs. The* q-th diagram distance d_q *between PDs is defined as:*

$$d_q(\mathrm{Dgm}, \mathrm{Dgm}') := \inf_{P \in \mathcal{P}(\mathrm{Dgm}, \mathrm{Dgm}')} c_q(P), \tag{5}$$

where $\mathcal{P}(\mathrm{Dgm}, \mathrm{Dgm}')$ *denotes the set of* partial correspondences *between* Dgm *and* Dgm'*, i.e., the set of subsets of* $\mathrm{Dgm} \times \mathrm{Dgm}'$ *s.t. the first and second projections* $\pi_1 : (p, p') \mapsto p$ *and* $\pi_2 : (p, p') \mapsto p'$ *are injective, and where the* q-th cost *of a partial correspondence* P *is defined as:*

$$c_q(P)^q := \sum_{(p,p') \in P} \|p - p'\|_\infty^q + \sum_{p \notin \mathrm{im}(\pi_1)} \|p - \pi_\Delta(p)\|_\infty^q + \sum_{p' \notin \mathrm{im}(\pi_2)} \|p' - \pi_\Delta(p')\|_\infty^q. \tag{6}$$

When $q \to +\infty$ *(i.e., with maximums instead of sums in Equation (6)), the* q*-th diagram distance is called the* bottleneck distance d_b *between PDs, and is equal to the interleaving distance between the corresponding persistence modules.*

A.2 (Multiple) scalar field analysis

Whenever the topological space X is unknown, and a finite sample $\hat{X}_n \subseteq X$ is provided, *scalar field analysis* allows to derive connections between the persistence modules of $\hat{X}_n$ (for a carefully chosen neighborhood graph) and the one of X.

Theorem 4 ([1, Theorem 4.2 (iii)])**.** *Let* X *be a compact geodesic space with convexity radius* $\rho(X)$*,* $f : X \to \mathbb{R}$ *be a* c*-Lipschitz function, and assume* $\hat{X}_n$ *is a finite geodesic* ε*-sample of* X*, i.e., every* $x \in X$ *is at geodesic distance at most* ε *from a point in* $\hat{X}_n$*. Let* $\delta \in [4\varepsilon, \rho(X)[$*, and let* G_δ *be the corresponding* δ*-neighborhood graph. Then, the single-parameter persistence modules induced by the* 0*-th dimensional homology groups of the sublevel sets of* $\hat{f} := f|_{\hat{X}_n}$ *and* f *are* $c\delta$*-interleaved, i.e.,* $d_b(\mathrm{Dgm}_0(f), \mathrm{Dgm}_0(\hat{f})) \leq c\delta$*.*

To avoid tuning δ, one can extend scalar field analysis to multi-parameter persistence modules using, in addition to the sublevel set filtration of $\hat{f}$, the *Vietoris-Rips (VR)* filtration. Recall that the VR filtration assigns, to every simplex of the complete simplicial complex Δ_n built on top of $\hat{X}_n$, the maximum pairwise distance between its vertices (assuming X is equipped with a metric), and computing the corresponding sublevel sets. Note also that the values of the filtration function f are only known on $\hat{X}_n$, so we use lower-star extensions to define it on Δ_n by picking, for every simplex, the maximum function values of its vertices.

We also use left Kan extensions to artificially increase the number of parameters of persistence modules and make them comparable in interleaving distance: given some (p)-multi-parameter persistence module M, its left Kan extension, denoted by $N = \mathrm{Lan}M$ is a $(p+1)$-multi-parameter persistence module whose horizontal morphisms of the form $(N_{u,x} \to N_{v,x})$, $u \leq v \in \mathbb{R}, x \in \mathbb{R}^p$ are identity maps and whose vertical morphisms of the form $(N_{u,x} \to N_{u,y})$, $u \in \mathbb{R}, x \leq_p y \in \mathbb{R}^p$ are the morphisms of M. See [1, Section 4.2] and Figure 1 therein for more details.

Theorem 5 ([1, Theorem 4.6 (iii)])**.** *Let* X *be a compact geodesic space with convexity radius* ρ_X*. Let* $f : X \to \mathbb{R}$ *be a* c*-Lipschitz function, and let* $\hat{X}_n$ *be a finite geodesic* ε*-sample of* X *for some* $\varepsilon > 0$*. Let* M *be the* 2*-multi-parameter persistence module induced by the* 0*-th dimensional homology groups of* (a) *the Vietoris-Rips filtration on* $\hat{X}_n$*, and* (b) *the lower-star extension of the sublevel set filtration of* $\hat{f}$*. Let* N *be the left Kan extension of the single-parameter persistence module induced by the* 0*-th dimensional homology groups of the sublevel sets of* f*. Assume* $\varepsilon < \rho_X/4$ *and let* $\delta_{\max} \in [4\varepsilon, \rho_X[$*. Then* M *and* N *are* $c\delta_{\max}$*-interleaved on the slab* $[4\varepsilon, \delta_{\max}] \times \mathbb{R}$*.*

A.3 Hierarchical clustering

Despite coming from different fields, hierarchical clusterings bear a lot of similarities with persistence modules induced by 0-th dimensional homology groups. In the following, we borrow some definitions and results from [25] (that we slightly adapt to our notations).

Definition 11 (Hierarchical Clustering [25, Definition 4])**.** *Let* X *be a topological space. A* clustering $\mathcal{C}$ *of* X *is a family of disjoint subsets of* X*. A clustering* $\mathcal{C}$ *is* comparable *to another clustering* $\mathcal{C}'$*, denoted as* $\mathcal{C} \preceq \mathcal{C}'$ *if* $\forall C \in \mathcal{C}, \exists! C' \in \mathcal{C}'$ *s.t.* $C \subseteq C'$*. A* hierarchical *clustering* $\mathcal{HC}$ *of* X *over* $\mathbb{R}$ *is a family of clusterings* $\{\mathcal{C}_\alpha\}_{\alpha \in \mathbb{R}}$ *s.t.* $\mathcal{C}_{\alpha_1} \preceq \mathcal{C}_{\alpha_2}$ *for any* $\alpha_1 \leq \alpha_2$*.*

Definition 12 (Interleaving [25, Definition 10])**.** *Let $\varepsilon > 0$. Two hierarchical clusterings $\mathcal{HC} = \{\mathcal{C}_\alpha\}_{\alpha\in\mathbb{R}}, \mathcal{HC}' = \{\mathcal{C}'_\alpha\}_{\alpha\in\mathbb{R}}$ are ε-interleaved if $\mathcal{C}_\alpha \preceq \mathcal{C}'_{\alpha+\varepsilon}$ and $\mathcal{C}'_\alpha \preceq \mathcal{C}_{\alpha+\varepsilon}$ for all $\alpha \in \mathbb{R}$.*

Definition 13 (Pruning [25, Definition 63])**.** *Let $\mathcal{HC}$ be a hierarchical clustering and $\tau > 0$. The* τ-pruning *of $\mathcal{HC}$ is another hierarchical clustering $\mathcal{HC}_{\geq\tau}$ defined as*

$$\mathcal{HC}_\tau = \{\{C \in \mathcal{C}_\alpha : \exists C' \in \mathcal{C}_{\alpha-\tau} \text{ s.t. } C' \subseteq C\}\}_{\alpha\in\mathbb{R}}. \tag{7}$$

Definition 14 (Persistent Clusters [25, Definitions 60 and 61])**.** *Let $\mathcal{HC}$ be a hierarchical clustering. The* persistent clusters $\mathrm{Pers}(\mathcal{HC})$ *of $\mathcal{HC}$ are the equivalence classes of $\sqcup_{\alpha\in\mathbb{R}} C_\alpha$ using the equivalence relation $\sim$ obtained as the symmetric closure of $C_1 \leftarrow C_2$ (for $C_1 \in \mathcal{C}_{\alpha_1}, C_2 \in \mathcal{C}_{\alpha_2}$, $\alpha_1 \leq \alpha_2$) if and only if $C_1 \subseteq C_2$ and $\forall \alpha_3 \in [\alpha_1, \alpha_2]$, $\exists! C_3 \in \mathcal{C}_{\alpha_3}$ s.t. $C_1 \subseteq C_3 \subseteq C_2$. The* lifetime *of a persistent cluster $\mathbf{C}$ is*

$$\mathrm{life}(\mathbf{C}) := \{\alpha \in \mathbb{R} : \exists C \in \mathcal{C}_\alpha \text{ (that we denote } \mathbf{C}_\alpha\text{) s.t. } \mathbf{C} = [C]_\sim\}. \tag{8}$$

Note that $\mathrm{life}(\mathbf{C})$ *is an interval of $\mathbb{R}$ by construction. The sequence of lifetime lengths (in decreasing order) of all persistent clusters is called the* prominence sequence $\mathrm{Prom}(\mathcal{HC})$ *of $\mathcal{HC}$.*

The underlying set $U(\mathbf{C})$ *is*

$$U(\mathbf{C}) = \cup_{\alpha\in\mathrm{life}(\mathbf{C})} \mathbf{C}_\alpha. \tag{9}$$

Finally, persistent clusters can be equipped with a partial order: $\mathbf{C} \preceq \mathbf{C}' \Leftrightarrow U(\mathbf{C}) \subseteq U(\mathbf{C}')$.

Definition 15 (Leaves [25, Definition 62])**.** *Let $\mathcal{HC}$ be a hierarchical clustering. The set of* leaves $\mathcal{L}(\mathcal{HC})$ *of $\mathcal{HC}$ are the minimal elements of* $\mathrm{Pers}(\mathcal{HC})$ *(equipped with its partial order), and the elements of $\mathcal{L}(\mathcal{HC}_\tau)$ (for some $\tau > 0$) are called the* τ-persistent clusters *of $\mathcal{HC}$.*

Theorem 6 ([25, Theorem 84])**.** *Let $\mathcal{HC}$ be a hierarchical clustering of a topological space X, and let $\mathrm{Prom}(\mathcal{HC})_{n+1} < \tau_n < \mathrm{Prom}(\mathcal{HC})_n$. Let $\mathcal{HC}'$ be another hierarchical clustering of X that is ε-interleaved with $\mathcal{HC}$, with $\varepsilon < \frac{1}{16}(\mathrm{Prom}(\mathcal{HC})_n - \mathrm{Prom}(\mathcal{HC})_{n+1})$, and let $\mathrm{Prom}(\mathcal{HC}')_{n+1} < \tau'_n < \mathrm{Prom}(\mathcal{HC}')_n$ such that $|\tau_n - \tau'_n| \leq 2\varepsilon$. Then, there exists a bijection b between the τ_n-persistent clusters of $\mathcal{HC}$ and the τ'_n-persistent clusters of $\mathcal{HC}'$ such that, for any $\mathbf{C} \in \mathcal{L}(\mathcal{HC}_{\tau_n})$ and corresponding $b(\mathbf{C}) \in \mathcal{L}(\mathcal{HC}'_{\tau'_n})$, one has that $\mathbf{C}$ and $b(\mathbf{C})$ are 3ε-interleaved.*

B Single-Parameter Persistence-Based Clustering: ToMATo

A particular instance of hierarchical clustering that fits within the framework of Section A.3 is *topological clustering*, and its corresponding algorithm, ToMATo (Topological Mode Analysis Tool), which was introduced in [8] as a clustering method blending hierarchical clustering and mode-seeking algorithms. See also [25, Section 6]. The core idea of ToMATo is to compute a hierarchy on the modes, or basins of attraction, of a given density estimator $f : \hat{X}_n \to \mathbb{R}$ (or, more generally, of any continuous function). Moreover, instead of a dendrogram (as in traditional hierarchical clustering), this hierarchy is encoded in the 0-th dimensional persistence diagram $\mathrm{Dgm}_0(f)$ (see Theorem 3).

Roughly speaking, ToMATo implements the following steps.

First, a graph G is computed on the data points, whose vertex set $V(G)$ is the dataset $\hat{X}_n$ itself. This graph is most of the time computed as a δ-neighborhood graph for some $\delta > 0$, but any user-provided graph can be used.

Then, ToMATo orders the data points based on their function values, from highest to lowest, and processes them one by one. The first ones (resp. last ones) to be considered are thus the local maxima (resp. minima) of the function. This sequence of growing subsets of the dataset is often called a *filtration* of the dataset.

Now, given some current point x, the graph G is first used to decide the type of x.

- If the function values of the neighbors of x are all below (resp. above) $f(x)$, its type is set as local maxima (resp. minima), and a new mode is introduced.
- If the set of neighbors have both larger and lower values, then the modes of the neighbors with larger function values, and the local maxima that they are associated with, are identified. If only one mode (resp. several modes) is found, the type of x is set as regular point (resp. saddle).

Finally, if x is a saddle, all modes are merged to the mode whose corresponding local maximum has highest function value. See Figure 1 and Algorithm 2.

As one can see, ToMATo creates a hierarchy between the modes, as every mode is pointing to the one that it is eventually merged with (except the mode associated to the global maximum, which is pointing to itself). Moreover, one can record, for each mode: (a) the function value of its local maximum, called its *birth time*, and (b) the function value of its saddle point inducing its merge, called its *death time* (if the mode is not associated to the global maximum).

Now, recall from Section A.1 that the persistence diagram (PD) of f (computed with superlevel sets instead of sublevel sets), denoted as $\mathrm{Dgm}_0(f)$, records exactly these birth and death times with intervals or, equivalently, with points in $\mathbb{R}^2$: every PD point represents a mode, whose birth and death times are used as coordinates. Note that the PD points of $\mathrm{Dgm}_0(f)$ are below the diagonal, as death time is always lower than birth time.

The absolute difference between the coordinates of a PD point p is usually called the *prominence* $\mathrm{prom}(p)$ of the corresponding mode. Finally, a clustering of $\hat{X}_n$ can be obtained by setting a threshold τ on the PD point prominences. In this case, the PD points with prominences at least τ are selected, and the corresponding modes are used as clusters. Then, for every remaining PD point and associated modes, the hierarchy built by ToMATo is used to decide the PD point and corresponding cluster (with prominence at least τ) that it should be merged with. Note that a nice feature of ToMATo is its complexity: using a Union-Find data structure, computing the hierarchy of ToMATo can be done in quasilinear time.

Data: $\hat{X}_n, f, G, \tau$
Result: $\mathcal{C}_{\text{ToMATo}}$
Sort $\{x_1, \dots, x_n\}$ in decreasing order of f: $f(x_1) \geq \cdots \geq f(x_n)$;
Initialize a union-find data structure $\mathcal{U}$ and two lists g, r of length n;
for $1 \leq i \leq n$ **do**
  $\mathcal{N} \leftarrow$ neighbors of x_i in G that have indices lower than i;
  if $\mathcal{N} = \varnothing$ **then**
    Create a new entry e in $\mathcal{U}$ and attach vertex x_i to it: $\mathcal{U}$.`MakeSet`(x_i);
    $r[e] \leftarrow x_i$;
  end
  else
    $g[i] \leftarrow \mathrm{argmax}\{f(x_j) : x_j \in \mathcal{N}\}$;
    $e_i \leftarrow \mathcal{U}$.`Find`$(g[i])$;
    Attach vertex x_i to the entry e_i: $\mathcal{U}$.`Union`(x_i, e_i);
    for $x_j \in \mathcal{N}$ **do**
      $e \leftarrow \mathcal{U}$.`Find`(x_j);
      if $e \neq e_i$ ***and*** $\min\{f(r[e]),\ f(r[e_i])\} < f(x_i) + \tau$ **then**
        $\mathcal{U}$.`Union`$(e,\ e_i)$;
        $r[e \cup e_i] \leftarrow \mathrm{argmax}\{f(r[e]),\ f(r[e_i])\}$;
        $e_i \leftarrow e \cup e_i$
      end
    end
  end
end
for $e \in \mathcal{U}$ **do**
  for $x \in e$ **do**
    $\mathcal{C}_{\text{ToMATo}}(x) \leftarrow r[e]$;
  end
end

Algorithm 2: The ToMATo algorithm

C Relationship between clustering and 0-th dimensional PH

As 0-th dimensional homology groups of a topological space X can always be represented with bases made of the connected components of X, hierarchical clusterings and 0-th dimensional persistence modules are *isomorphic*, and definitions is Sections A.1 and A.3 can be linked:

Lemma 1 ([22, Lemma 1])**.** *Let X be a topological space and $f : X \to \mathbb{R}$ be a continuous function. Let $\mathcal{HC}_{\downarrow f} := \{\pi_0(F_\alpha)\}_{\alpha\in\mathbb{R}}$ be the hierarchical clustering obtained by computing the connected components of the superlevel sets of f (slightly abusing the notations of Section A.1 in which they denote sublevel sets). Then, the map $\langle C_\alpha\rangle \to H_0(F_\alpha)$ is an isomorphism for all $\alpha \in \mathbb{R}$. As such, the following consequences hold:*

1. *the clusters in* $\textsc{ToMATo}(X, f, \tau)$ *are exactly the τ-persistent clusters of $\mathcal{HC}_{\downarrow f}$,*
2. *the sequence* $\mathrm{Prom}(\mathcal{HC}_{\downarrow f})$ *is exactly the sequence of mode prominences* $\{\mathrm{prom}(p)\}_{p\in\mathrm{Dgm}_0(f)}$ *(in decreasing order), and*
3. *for any function $g : X \to \mathbb{R}$ and associated $\mathcal{HC}_{\downarrow g} := \{\pi_0(G_\alpha)\}_{\alpha\in\mathbb{R}}$, where $\{G_\alpha\}_{\alpha\in\mathbb{R}}$ denotes the family of superlevel sets of g, one has that $\mathcal{HC}_{\downarrow f}$ and $\mathcal{HC}_{\downarrow g}$ are ε-interleaved if and only if $d_b(\mathrm{Dgm}_0(f), \mathrm{Dgm}_0(g)) \leq \varepsilon$.*

This leads to Theorem 1, which we now prove.

Proof. (Theorem 1) Let N be the number of PD points of $\mathrm{Dgm}_0(f)$ with prominences larger than d_2. By assumption, one has $d_2 \leq \mathrm{Prom}(\mathcal{C}_f)_N$ and $d_1 \geq \mathrm{Prom}(\mathcal{C}_f)_{N+1}$. As $d_b(\mathrm{Dgm}_0(f), \mathrm{Dgm}_0(\hat{f})) \leq \varepsilon$, $\mathrm{Dgm}_0(\hat{f})$ is $(d_1 + 2\varepsilon, d_2 - 2\varepsilon)$-separated, and the number of PD points of $\mathrm{Dgm}_0(\hat{f})$ above $d_2 - 2\varepsilon$ is equal to N (otherwise $d_b(\mathrm{Dgm}_0(f), \mathrm{Dgm}_0(\hat{f}))$ would be larger than ε). Hence, one has $\mathrm{Prom}(\mathcal{C}_{\hat{f}})_{N+1} < \hat{\tau} < \mathrm{Prom}(\mathcal{C}_{\hat{f}})_N$. The result follows from Theorem 6 and Lemma 1. Note that the relation between clusters is 4ε and not 3ε (as stated in Theorem 6) to ensure proper inclusion, as the death time of the persistent clusters might differ by ε by assumption. □

The following result is a immediate consequence of Theorem 1 (with $\varepsilon = c\delta$) for PDs obtained within the framework of Theorem 4 (applied with superlevel sets):

Theorem 7. *Let X be a compact geodesic space with convexity radius $\rho(X)$, $f : X \to \mathbb{R}$ be a c-Lipschitz function, and assume $\hat{X}_n$ is a finite geodesic ε-sample of X. Let $\delta \in [4\varepsilon, \rho(X)[$, and let G_δ be the corresponding δ-neighborhood graph. Then, the single-parameter persistence modules induced by the 0-th dimensional homology groups of the superlevel sets of $\hat{f}$ and f are $c\delta$-interleaved, i.e., $d_b(\mathrm{Dgm}_0(f), \mathrm{Dgm}_0(\hat{f})) \leq c\delta$.*

Moreover, if $\mathrm{Dgm}_0(f)$ is (d_1, d_2)-separated and $\delta < \frac{d_2-d_1}{16c}$, then $\textsc{ToMATo}(\hat{X}_n, \hat{f}, G_\delta, \hat{\tau})$ is $4c\delta$-related to $\textsc{ToMATo}(X, f, \tau)$, for any $d_1 + 4c\delta < \tau < d_2 - 4c\delta$ and $\hat{\tau}$ s.t. $|\tau - \hat{\tau}| \leq 2c\delta$.

Proof. The proof is exactly the same than the one of Theorem 1 with $\varepsilon = c\delta$. Note however that as f and $\hat{f}$ are defined on different spaces, one should actually apply Theorem 6 with *correspondences*, as stated in [25]. To alleviate the exposition, we leave this subtlety aside as the main arguments of the proof extend straightforwardly. □

The following result is a immediate consequence of Theorem 1 (with $\varepsilon = \eta$) for sliced PDs (recall the notations in Definition 2) combined with a stability result from [19]:

Theorem 8 ([19, Lemma 2])**.** *Let X be a topological space. Let $f_1, \ldots, f_p : X \to \mathbb{R}$ be continuous functions. Let ℓ, ℓ' be two diagonal lines at distance η from each other. Then,*

$$d_b(\mathrm{Dgm}_0(F_\ell(f_1, \ldots, f_p)), F_{\ell'}(f_1, \ldots, f_p))) \leq \eta. \tag{10}$$

Moreover, if $\mathrm{Dgm}_0(F_\ell(f_1, \ldots, f_p))$ is (d_1, d_2)-separated and $\eta < \frac{d_2-d_1}{16}$, then $\textsc{ToMATo}(X, F_\ell(f_1, \ldots, f_p), \tau)$ and $\textsc{ToMATo}(X, F_{\ell'}(f_1, \ldots, f_p), \tau')$ are mutually 4η-related, for any $d_1 + 4\eta < \tau < d_2 - 4\eta$ and $\hat{\tau}$ s.t. $|\tau - \hat{\tau}| \leq 2\eta$.

D Proofs of Section 3.2

The proof of Theorem 2 follows from the results in Sections A and C.

Proof. Theorem 2.

We first prove (i).

Let $\ell_i \in \mathcal{L}$. Theorem 8, Equation (10) ensures that the points of $\mathrm{Dgm}_0(F_{\ell_{i+1}}(f_1, \dots, f_p))$ are included in $\|\cdot\|_\infty$-balls of radius η around the points of $\mathrm{Dgm}_0(F_{\ell_i}(f_1, \dots, f_p))$. Then, Assumption (A2) ensures that, for PD points with prominences at least d_2, these balls do not intersect, making the possibilities of finding matching functions with cost less than η reduced to one candidate (the one that matches the points in $\mathrm{Dgm}_0(F_{\ell_i}(f_1, \dots, f_p))$ to the ones of $\mathrm{Dgm}_0(F_{\ell_{i+1}}(f_1, \dots, f_p))$ in their corresponding balls). Hence, the matching functions are unique when restricted to PD points with prominences at least d_2. Now these restricted matching functions must be bijections, otherwise the cost of such matching functions would exceed $\frac{d_2 - d_1}{2} > \eta$.

We now prove (ii) and (iii).

As the matching functions are unique and induced by inclusions (from Lemma 1), they must coincide with the bijections in Theorem 8. Now, let $x \in \bigcap_{\ell \in \mathcal{L}} \mathrm{core}_{\ell, \textsc{ToMATo}}(i)$, for some $1 \leq i \leq N$. Let ℓ_1 be the first line in $\mathcal{L}$. By definition, one has $\mathcal{C}_{\ell_1, \textsc{MMA}}(x) = i$. As $x \in \mathrm{core}_{\ell_1, \textsc{ToMATo}}(i)$, let $p_{\pi_{\ell_1}(i)}$ be the corresponding point of $\mathrm{Dgm}_0(F_{\ell_1}(f_1, \dots, f_p))$ associated to the mode that x belongs to. By definition, $p_{\pi_{\ell_1}(i)}$ must have prominence at least d_2. Then, Theorem 8 ensures that x also belongs to the cluster of $\mathcal{C}_{\ell_2, \textsc{ToMATo}}$ associated to the point $p_j := \mathrm{m}_{\ell_1}(p_{\pi_{\ell_1}(i)}) \in \mathrm{Dgm}(F_{\ell_2}(f_1, \dots, f_p))$, where $1 \leq j \leq N$. Moreover, by definition, one has $j = \pi_{\ell_2}(i)$, and $\mathcal{C}_{\ell_2, \textsc{MMA}}(x) = i$. As one can apply the same reasoning iteratively to the remaining lines of $\mathcal{L}$, it follows that $\mathcal{C}_{\ell, \textsc{MMA}}(x) = i$ for all $\ell \in \mathcal{L}$, and thus we finally have $\mathcal{C}_{\textsc{ToMAToMP}}(x) = i$. □

Extensions of Theorem 2. Theorem 2 can be straightforwardly strengthened in three ways:

- (i) The result still holds even if the PDs are (d_1, d_2)-separated only for a majority, i.e., a subset $\tilde{\mathcal{L}}$ of size at least $\frac{|\mathcal{L}|}{2}$, of consecutive lines in $\mathcal{L}$. Indeed, the most frequent cluster (computed with the argmax in Algorithm 1) for $\mathcal{L}$ will, in that case, coincide with the cluster obtained for $\tilde{\mathcal{L}}$ with Theorem 2. We use such majority statements in both Corollaries 1 and 2. Different thresholds $\tau_1, \dots, \tau_p$ can also be used as long as they are 2η close to τ.
- (ii) If the functions $f_1, \dots, f_p$ are not provided, and one only has access to perturbed versions $\tilde{f}_1, \dots, \tilde{f}_p$ instead, with $\|f_i - \tilde{f}_i\|_\infty \leq \varepsilon$ and $\varepsilon \leq \frac{d_2 - d_1}{4}$, then the PDs will remain $(d_1 + 2\varepsilon, d_2 - 2\varepsilon)$-separated, ensuring that the clusters will remain correct upon decreasing η.
- (iii) Concerning Assumption (A2), it might be impossible to find η s.t. $\eta \leq d^*$ as $d^* = 0$ if one of the sliced PDs has points with multiplicity greater or equal to two. In that case, one can get rid of this upper bound by using matching functions induced by inclusions (which can be obtained using, e.g., vineyard matching functions [11] and representative cycle updates [15]) instead of the q-th diagram distances. Hence, we neglect this assumption in Corollaries 1 and 2.

Proof. Corollary 1. Theorem 5 (applied with superlevel sets) guarantees that the sliced PDs used for computing the clusters of $\mathcal{C}_{\textsc{ToMAToMP}}(\hat{X}_n, g, \tilde{f}, \tilde{G}, \tau, \mathcal{L}, \{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}})$ come from a 2-parameter persistence module M that is $c\delta_{\max}$ away (in interleaving distance) from the left Kan extension N of $\mathrm{Dgm}_0(f)$ on the slab $[-\delta_{\max}, -4\varepsilon] \times \mathbb{R}$. As all sliced PDs of N are (d_1, d_2)-separated (using appropriate linear parametrizations), it follows that all sliced PDs of M on the slab are $(d_1 + 2c\delta_{\max}, d_2 - 2c\delta_{\max})$-separated. See Figure 5 (left). The corollary then follows directly from applying Theorem 2 and Theorem 7. □

Proof. Corollary 2. Using topologically robust points ensures that the sliced PDs used for computing the clusters of $\mathcal{C}_{\textsc{ToMAToMP}}(\hat{X}_n, g, \tilde{f}|_{\hat{X}_n}, G, \tau, \mathcal{L}, \{\mathrm{m}_\ell\}_{\ell \in \mathcal{L}})$ come from a 2-parameter persistence module M that is $c\delta$ away (in interleaving distance) from the left Kan extension N of $\mathrm{Dgm}_0(f)$ on the slab $[K'_g, K_g] \times \mathbb{R}$. Indeed, a topologically robust point x can be removed from G_δ without

changing its 0-th dimensional homology group. As for persistence modules obtained with the super-level sets of f, the only effect is to replace its function value by that of its neighbor $n(x)$, which is at most $c\delta$ away. See Figure 5 (right). As all sliced PDs of N are (d_1, d_2)-separated (using appropriate linear parametrizations), it follows that all sliced PDs of M on the slab are $(d_1 + 2c\delta, d_2 - 2c\delta)$-separated. The corollary then follows directly from applying Theorem 2 and Theorem 7. □

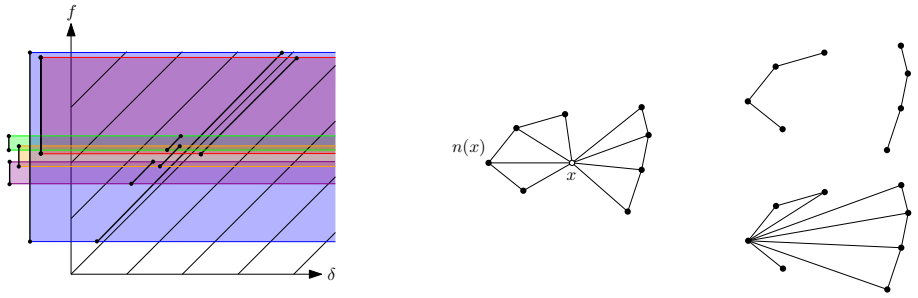

Figure 5: **(Left)** Left Kan extension of $\mathrm{Dgm}_0(f)$. **(Right)** *Left.* Example of point x with outlier value (empty disk) who is not topologically robust. *Top right.* Removing x changes the graph topology. *Bottom right.* Adding edges to the leftmost neighbor $n(x)$ makes x topologically robust.

E Additional Results

E.1 Additional tables

Method	AMI Score	ARI Score	Time (s)
KMeans	0.43	0.14	**0.03**
SpectralClustering	0.69	0.42	5.84
AgglomerativeClustering	0.45	0.14	0.04
DBSCAN	0.00	0.00	**0.03**
DensityPeaks	0.53	0.22	0.08
HDBSCAN	0.75	0.78	0.04
Louvain	0.57	0.17	0.38
Leiden	0.56	0.16	0.13
ToMATo (min)	0.00	0.00	7.21
ToMATo (mean)	0.66	0.61	7.21
ToMAToMP	**0.82**	**0.80**	4.22

Table 4: AMI, ARI scores and times for `synthetic` in the `no-radius` experiment.

Method	AMI Score	ARI Score	Time (s)
KMeans	0.38 ± 0.03	0.11 ± 0.02	0.23 ± 0.17
SpectralClustering	0.67 ± 0.00	0.41 ± 0.00	7.07 ± 3.21
AgglomerativeClustering	0.43 ± 0.03	0.12 ± 0.03	0.04 ± 0.02
DBSCAN	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.01
DensityPeaks	0.53 ± 0.01	0.23 ± 0.01	0.12 ± 0.06
HDBSCAN	0.72 ± 0.02	0.75 ± 0.04	0.04 ± 0.02
Louvain	0.56 ± 0.02	0.18 ± 0.02	0.79 ± 0.48
Leiden	0.56 ± 0.02	0.18 ± 0.02	0.29 ± 0.17
ToMATo	0.59 ± 0.12	0.52 ± 0.16	**0.00 ± 0.01**
ToMAToMP	**0.93 ± 0.01**	**0.96 ± 0.01**	18.09 ± 6.59

Table 5: AMI, ARI scores and times for `synthetic` in the `outlier` experiment.

Method	AMI Score	ARI Score	Time (s)
KMeans	0.41 ± 0.20	0.28 ± 0.18	0.18 ± 0.35
SpectralClustering	0.38 ± 0.18	0.25 ± 0.17	5.41 ± 3.96
AgglomerativeClustering	**0.42 ± 0.19**	0.28 ± 0.19	0.03 ± 0.03
DBSCAN	0.03 ± 0.11	0.02 ± 0.08	**0.01 ± 0.01**
DensityPeaks	0.34 ± 0.18	0.20 ± 0.16	0.09 ± 0.08
HDBSCAN	0.25 ± 0.13	0.12 ± 0.15	0.03 ± 0.02
Louvain	0.41 ± 0.15	0.21 ± 0.12	0.64 ± 0.89
Leiden	0.41 ± 0.16	0.21 ± 0.12	0.26 ± 0.39
ToMATo (min)	0.26 ± 0.21	0.16 ± 0.19	10.97 ± 12.90
ToMATo (mean)	0.38 ± 0.19	**0.29 ± 0.21**	10.97 ± 12.90
ToMAToMP	0.35 ± 0.20	0.27 ± 0.22	11.13 ± 19.23

Table 6: AMI, ARI scores and times for `3dshape` in the `no-radius` experiment.

Method	AMI Score	ARI Score	Time (s)
KMeans	0.69 ± 0.00	0.44 ± 0.01	0.21 ± 0.18
SpectralClustering	0.69 ± 0.00	0.43 ± 0.01	11.89 ± 3.94
AgglomerativeClustering	0.67 ± 0.00	0.41 ± 0.01	0.24 ± 0.06
DBSCAN	0.00 ± 0.00	0.00 ± 0.00	0.11 ± 0.02
DensityPeaks	0.63 ± 0.00	0.37 ± 0.01	1.01 ± 0.15
HDBSCAN	0.12 ± 0.01	0.01 ± 0.00	0.08 ± 0.02
Louvain	0.67 ± 0.00	0.39 ± 0.00	0.26 ± 0.10
Leiden	0.68 ± 0.00	0.42 ± 0.01	0.19 ± 0.06
ToMATo	0.55 ± 0.02	0.25 ± 0.02	**0.00 ± 0.00**
ToMAToMP	**0.74 ± 0.00**	**0.54 ± 0.00**	4.47 ± 1.20

Table 7: AMI, ARI scores and times for `image` in the `outlier` experiment.

Method	Corr.	Top-10	Time (s)
AgglomerativeClustering	0.47	0.30	**0.34**
ToMATo (min)	nan	0.90	2.91
ToMATo (mean)	nan	0.99	2.91
ToMAToMP	**1.00**	**1.00**	64.51

Table 8: Correlation, Top@10 scores and times for `kpmp` (1-gr) in the `no-radius` experiment. NaN values are induced by poor line choices in TOMATO for which all genes share the same score.

Method	Corr.	Top-10	Time (s)
AgglomerativeClustering	0.26	0.10	**0.35**
ToMATo (min)	**1.00**	0.90	7.91
ToMATo (mean)	**1.00**	0.97	7.91
ToMAToMP	**1.00**	**1.00**	111.82

Table 9: Correlation, Top@10 scores and times for `spat` (1-gr) in the `no-radius` experiment.

Method	Corr.	Top-10	Time (s)
AgglomerativeClustering	**0.27**	0.50	**0.65**
ToMATo (min)	nan	0.70	1.40
ToMATo (mean)	nan	**0.93**	1.40
ToMAToMP	0.20	0.60	151.48

Table 10: Correlation, Top@10 scores and times for `kpmp` (2-gr) in the `no-radius` experiment. NaN values are induced by poor line choices in TOMATO for which all genes share the same score.

Method	Corr.	Top-10	Time (s)
AgglomerativeClustering	0.50	0.60	**0.38**
ToMATo (min)	0.99	0.80	1.92
ToMATo (mean)	**1.00**	**0.95**	1.92
ToMAToMP	0.57	0.80	161.69

Table 11: Correlation, Top@10 scores and times for `spat` (2-gr) in the `no-radius` experiment.

Method	Corr.	Top-10	Time (s)
AgglomerativeClustering	0.40 ± 0.05	0.44 ± 0.10	0.48 ± 0.23
ToMATo	0.41 ± 0.06	0.49 ± 0.08	**0.02 ± 0.01**
ToMAToMP	**1.00 ± 0.01**	**0.99 ± 0.02**	32.73 ± 7.87

Table 12: Correlation, Top@10 scores and times for `kpmp` (1-gr) in the `outlier` experiment.

Method	Corr.	Top-10	Time (s)
AgglomerativeClustering	0.79 ± 0.01	0.70 ± 0.03	0.41 ± 0.23
ToMATo	0.93 ± 0.04	0.86 ± 0.09	**0.01 ± 0.01**
ToMAToMP	**0.98 ± 0.01**	**0.97 ± 0.04**	34.14 ± 7.92

Table 13: Correlation, Top@10 scores and times for `spat` (1-gr) in the `outlier` experiment.

Method	Corr.	Top-10	Time (s)
AgglomerativeClustering	0.23 ± 0.14	0.24 ± 0.13	0.31 ± 0.18
ToMATo	0.09 ± 0.20	0.21 ± 0.17	**0.01 ± 0.01**
ToMAToMP	**0.27 ± 0.09**	**0.57 ± 0.06**	61.34 ± 11.71

Table 14: Correlation, Top@10 scores and times for `kpmp` (2-gr) in the `outlier` experiment.

Method	Corr.	Top-10	Time (s)
AgglomerativeClustering	0.36 ± 0.03	0.63 ± 0.07	0.36 ± 0.22
ToMATo	**0.79 ± 0.12**	0.79 ± 0.11	**0.01 ± 0.01**
ToMAToMP	0.59 ± 0.03	**0.80 ± 0.01**	58.81 ± 8.79

Table 15: Correlation, Top@10 scores and times for `spat` (2-gr) in the `outlier` experiment.

E.2 Scores and running times vs Number of lines for TOMATOMP

In this section, we investigate the dependencies between the number of lines used in TOMATOMP and the corresponding scores and running times. As one can see from the plots, the scores stay roughly constant, which motivates our choice of keeping a moderate ($|\mathcal{L}| = 100$) number of lines.

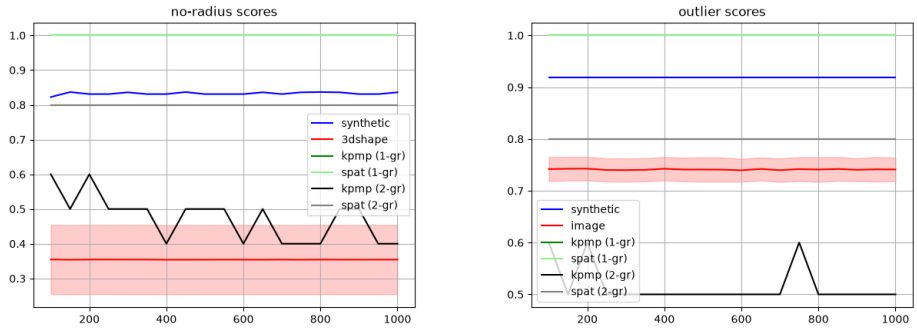

Figure 6: Quality scores vs number of lines $|\mathcal{L}|$ for the **(Left)** `no-radius` and **(Right)** `outlier` experiments.

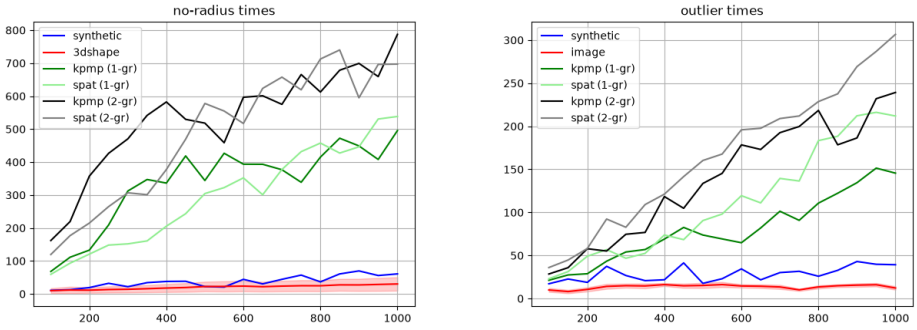

Figure 7: Running times (s) vs number of lines $|\mathcal{L}|$ for the **(Left)** `no-radius` and **(Right)** `outlier` experiments.

E.3 Additional plots

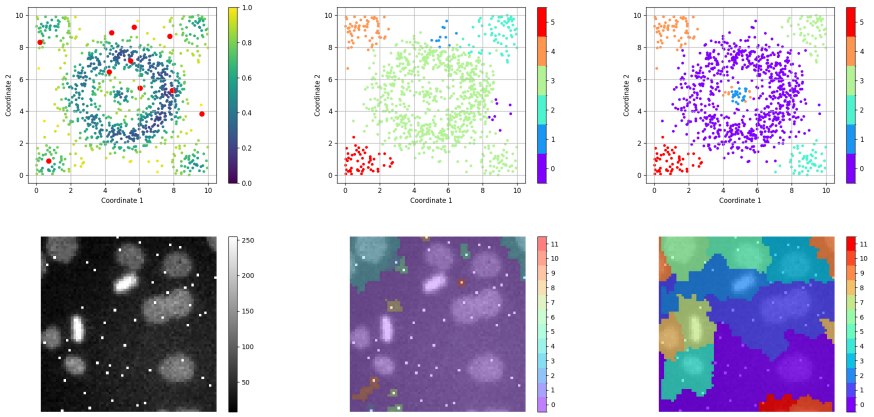

Figure 8: **(Left)** `synthetic` and `image` datasets plagued with outlier values. **(Middle)** ToMATo clusterings are highly perturbed by these outliers. **(Right)** ToMAToMP clusters remain robust.

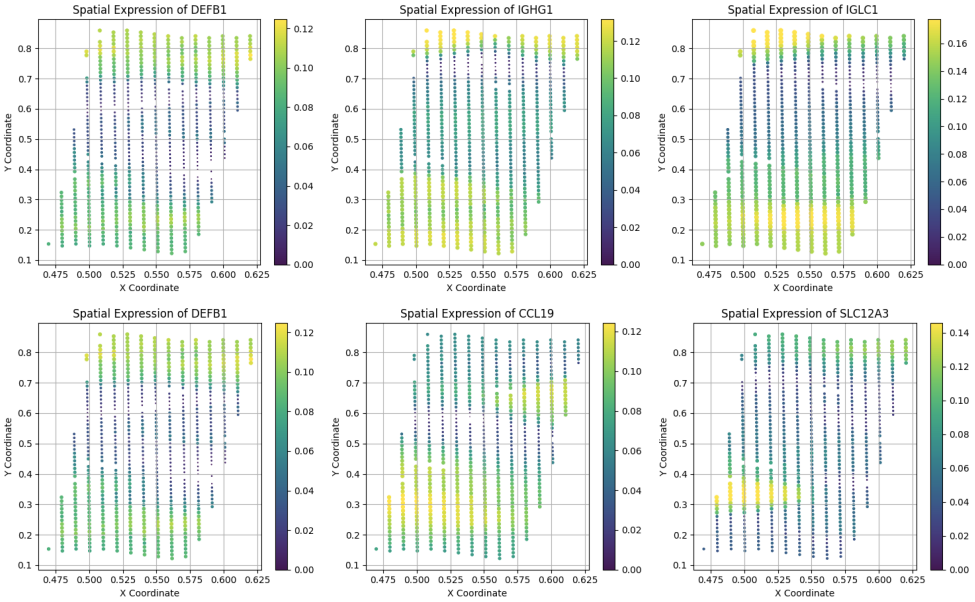

Figure 9: **(Top)** Top ranked triplet of genes for `kpmp`. **(Bottom)** Lower ranked triplet of genes.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The robustness guarantees for our method ToMAToMP are proved in Section 3.2, and the experimental evidence is provided in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Computational limitations of our approach (especially the long running times) are discussed in Section 4 and Section 5. Discussion of our assumptions are available in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.

- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Our main result Theorem 2 has two assumptions that are clearly stated. Its proof is available in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All hyperparameters that are necessary to run the method are provided in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide code for running experiments, together with datasets, in a self-explanatory notebook provided as supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [No]

Justification: As the article focuses on unsupervised learning, there are no train/test datasets, but hyperparameters are specified in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Error bars are reported for the experiments involving randomness (outliers). See Section 4, Appendix E.1, Appendix E.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Computation details are provided in Section 4 in the corresponding paragraph.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper has no ethics issue, as all used data is public and anonymized.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The proposed work has no societal impact, as it is not anchored to specific applications with such potential impacts. It is a methodological paper proposing a general unsupervised method with no particular intended usage in potentially positive or negative societal impact tasks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All code assets (multipers, gudhi and scikit-learn) are open source, and cited in Section 4. The datasets that we use are either public in open source databases (and we shared the URL in Section 4), or available as supplementary materials of the references that are cited (this concerns in particular the spatial transcriptomics datasets).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code is provided for reproducibility, and clean documentation is provided.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Our proposed method TOMATOMP does not use any LLM.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.