INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Persistence-Based Clustering in Riemannian Manifolds

Frédéric Chazal — Leonidas Guibas — Steve Oudot — Primoz Skraba

N° 6968

June 2009

ISSN 0249-6399 ISRN INRIA/RR--6968--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SACLAY - ÎLE-DE-FRANCE

Persistence-Based Clustering in Riemannian Manifolds

Frédéric Chazal*, Leonidas Guibas†, Steve Oudot*, Primoz Skraba*

Thème : SYM — Systèmes symboliques
Équipe-Projet Geometrica

Rapport de recherche n° 6968 — June 2009 — 47 pages

Abstract: We present a novel clustering algorithm that combines a mode-seeking phase with a cluster merging phase. While mode detection is performed by a standard graph-based hill-climbing scheme, the novelty of our approach resides in its use of *topological persistence* theory to guide the merges between clusters. An interesting feature of our algorithm is to provide additional feedback in the form of a finite set of points in the plane, called a *persistence diagram*, which provably reflects the prominence of each of the modes of the density. Such feedback is an invaluable tool in practice, as it enables the user to determine a set of parameter values that will make the algorithm compute a relevant clustering on the next run.

In terms of generality, our approach requires the sole knowledge of (approximate) pairwise distances between the data points, as well as of rough estimates of the density at these points. It is therefore virtually applicable in any arbitrary metric space. In the meantime, its complexity remains reasonable: although the size of the input distance matrix may be up to quadratic in the number of data points, a careful implementation only uses a linear amount of main memory and barely takes more time to run than the one spent reading the input.

Taking advantage of recent advances in topological persistence theory, we are able to give a theoretically sound notion of what the *correct* number k of clusters is, and to prove that under mild sampling conditions and a relevant choice of parameters (made possible in practice by the persistence diagram) our clustering scheme computes a set of k clusters whose spatial locations are bound to the ones of the basins of attraction of the peaks of the density. These guarantess hold in a large variety of contexts, including when data points are distributed along some unknown Riemannian manifold.

Key-words: clustering, topological persistence, Rips graph, barcode, unsupervised learning.

* INRIA Saclay – Île-de-France. Email: `firstname.lastname@inria.fr`
† Stanford University. Email: `guibas@cs.stanford.edu`

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

Clustering dans les variétés riemanniennes basé sur la persistance

Résumé : Nous présentons un nouvel algorithme de clustering qui combine une phase de recherche de modes avec une phase de fusion des clusters. Alors que la recherche de modes s'effectue par une méthode standard d'ascension de gradient dans un graphe de voisinage, la nouveauté de notre approche réside dans son utilisation de la *persistance topologique* pour guider la fusion entre clusters. Une propriété intéressante de notre algorithme est de retourner un *diagramme de persistance* en plus des clusters. Ce diagramme est utile en pratique pour déterminer des valeurs de paramètres qui permettent à l'algorithme de calculer un ensemble pertinent de clusters à la prochaine exécution.

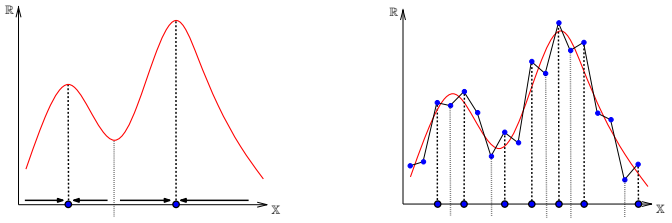
Notre approche requiert juste de connaître les distances entre points de données ainsi qu'une estimation de la densité en ces points. Elle peut donc être appliquée dans pratiquement n'importe quel espace métrique. De plus, sa complexité reste raisonnable : alors que la taille de la matrice des distances peut être jusqu'à quadratique en le nombre de points de données, une implémentation réfléchie de l'algorithme n'utilise qu'une quantité linéaire de mémoire et met à peine plus de temps à s'exécuter que le temps mis pour lire l'entrée du programme.

En nous appuyant sur des résultats récents sur la théorie de la persistance, nous donnons une définition théoriquement fondée du nombre *correct* k de clusters à construire, et nous prouvons que sous des hypothèses raisonnables d'échantillonnage et modulo un choix judicieux des paramètres notre algorithme calcule précisément k clusters dont les localisations géographiques sont corrélées aux bassins d'attraction des pics de la densité. Ces garanties théoriques sont valides dans un contexte très large, incluant entre autres le cas où les points de données sont échantillonnés le long d'une variété riemannienne inconnue.

Mots-clés : clustering, persistance topologique, graphe de Rips, code-barre, apprentissage non supervisé.

1 Introduction

Unsupervised learning or clustering is an important tool for understanding and interpreting data in a variety of fields. Although in many settings the *natural* clustering is obvious to a person, the problem of clustering remains ill-posed in general. Nevertheless, its importance as a tool for exploratory data analysis has grown with the increased avaliability of massive and high-dimensional datasets. On such data, interpretation by direct inspection is difficult, if not impossible. A common viewpoint is that a data set consists of samples drawn from some unknown density function f, and that the ultimate goal of the analysis is to understand the structure of that density. Since f is usually not provided, it must be estimated from the available samples. Clustering methods therefore rely on density estimators, which fall into two broad categories: parametric estimators, which presuppose a family of functions as a model for the density; non-parametric estimators, derived from the local behaviour of the density function. Methods based on parametric estimators use knowledge of the density to achieve better results, whereas non-parametric methods are more general as they are not tied to any particular model for the density.



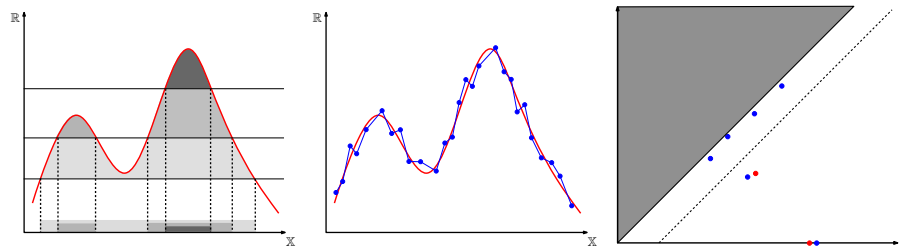
(a) (b)

Figure 1: (a.) A density function f with two peaks. The center-line gives the separation between their basins of attraction. (b.) The corresponding peaks and their basins in a piecewise-linear interpolation of f.

With the samples coming from a density function f, clusters can be naturally identified with the *basins of attraction* of the peaks of f. Intuitively, considering f as a terrain, a cluster is the set of all points *flowing* into the same local maximum (or peak) along the flow defined by the gradient vector field of f. This notion of clusters is not novel: it was already proposed by Koontz *et al.* in a graph-based gradient ascent algorithm [24], and used in numerous subsequent mode-seeking algorithms, including Mean-Shift [11] and its successors [30, 34]. A common problem faced by these methods is that the gradient flow and extremal points of the density function f are differential quantities that are notoriously unstable under (even arbitrarily small) perturbations of f. Since f remains usually unknown to us, we are dependent on an estimator $\tilde{f}$, whose basins of attraction are very unlikely to coincide with the ones of the true density. See Figure 1 for an illustration. To avoid this pitfall, methods such as Mean-Shift try to smooth the estimated density function, which brings up the question of how much smoothing is required.

Rather than directly studying the gradient flow of f, *topological persistence* [18, 35] studies the evolution of the topology of the superlevel-sets of

f, *i.e.* the sets of the form $f^{-1}([\alpha, +\infty))$, as parameter α decreases[1] from $+\infty$ to $-\infty$. In the context of clustering, we are mainly interested in the path-connectivity of the superlevel-sets, a special instance of persistence theory known as 0-*dimensional persistence* or *size theory* in the literature [13]. Figure 2(a) shows the connected components of three superlevel-sets. Each component appears when a local maximum of f is reached. Moreover, topological persistence imposes a strict hierarchy on the components: when two of them get connected to each other in some superlevel-set of f, the component generated by the lower peak is said to be *merged* into the one generated by the higher peak. Each component C can then be assigned a *lifespan*, encoded as a point p in the plane: the abcissa of p is the time at which C appears in the family of superlevel-sets of f; the ordinate of p is the time at which C gets merged into another component generated by some higher peak of f. The difference $p_x - p_y$ is an indicator of the *prominence* of C, or equivalently of its generating peak. This quantity is equal to twice the distance of point p to the diagonal $y = x$ in the plane. The collection of such points is called the 0-*th persistence diagram* of f, noted $\mathrm{D}_0 f$ [10]. See Figure 2 for an illustration.



(a) (b) (c)

Figure 2: Evolution of the connectivity of the superlevel-sets of a function f (image a.) and of an approximation $\tilde{f}$ (image b.). In image (c.), the persistence diagrams of f (red) and of $\tilde{f}$ (blue) are superimposed, thus showing that $\tilde{f}$ has two prominent peaks corresponding to the two prominent peaks of f, by the stability properties of such diagrams.

In practice, comparing the diagram of f with the one of an approximation $\tilde{f}$ (Figure 2(c)), one can see that the points far from the diagonal, which correspond to highly prominent peaks of f, are well-preserved under perturbation, in contrast to points close to the diagonal, which correspond to non-significant peaks and can therefore be regarded as noise. This stability property is a fundamental result of persistence theory [3, 10] that justifies its use in the context of data analysis: when the true density $f : \mathbb{X} \to \mathbb{R}$ remains unknown, it is still possible to approximate $\mathrm{D}_0 f$ via the persistence diagram of some (usually piecewise-linear) approximation $\tilde{f}$ of f. Nevertheless, building (piecewise-linear) approximations of f over the entire space $\mathbb{X}$ is prohibitively costly when the dimensionality of $\mathbb{X}$ is high, and it becomes virtually impossible when $\mathbb{X}$ is known solely through the matrix of pairwise distances between the data points, a typ-

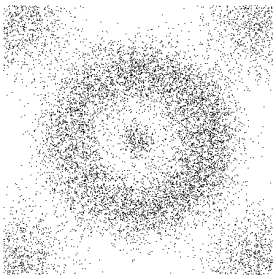
[1] We depart from the classical description of topological persistence by considering superlevel-sets and reversing the time flow. This is a purely formal choice that does not affect the validity of the theory.

ical scenario in clustering. This may explain why persistence has hardly been exploited in mode-seeking algorithms so far.

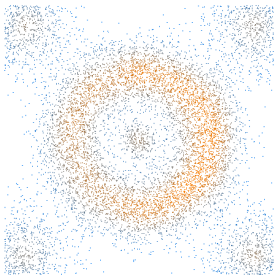
It should be noted however that persistence has been used in other clustering approaches. For instance, the dendrograms produced by single-linkage clustering are nothing but alternate representations for persistence diagrams. Here, the considered function is not the density underlying the input point cloud L, but rather the opposite of the distance restricted to the product $L \times L$. The hierarchy of clusterings induced by persistence provides a coarse-to-fine representation of the input point cloud, which may help the user find the *best* scale(s) at which to process the data.

Our contributions. In this paper we take advantage of a recent stability result for persistence diagrams [3] that enables the comparison of the diagrams of functions defined over different spaces. If f is an unknown real-valued function defined over an unknown space $\mathbb{X}$, of which a finite sampling L is given together with an approximation $\tilde{f}$ of f over L, then the result of [3] makes it possible to recover (an approximation of) $\mathrm{D}_0 f$ by building an auxiliary data structure on top of the point cloud L, such as a neighborhood graph G, and by extending $\tilde{f}$ over this structure. As shown in [6], this approximation property holds provided that some minimum sampling density is achieved throughout the space $\mathbb{X}$, which unfortunately may not be the case in the context of clustering, where the input point cloud L is sampled according to some density function f that may not have full support. Our first contribution is to show that a weaker version of the approximation result of [6] holds when only some superlevel-set of the function f is densely sampled (Theorem 4.5). This weaker setting is well-suited for clustering applications, where superlevel-sets of density functions are precisely the regions where more sample points are likely to be present.

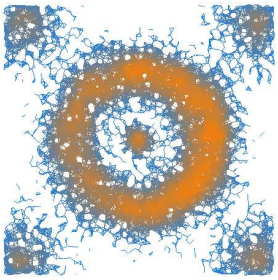
With this new result at hand, we propose a novel clustering scheme (Section 3) that combines a graph-based mode-seeking step *à la* Koontz *et al.* [24] with a merging step guided by topological persistence, thus taking advantage of both worlds. Literally, our mode-seeking step is the algorithm of [24]: given a parameter $\delta \geq 0$ and a density estimator $\tilde{f}$, we build a neighborhood graph G (also called *Rips graph*) by connecting every pair of input points lying within distance δ of each other (Figure 3(c)); we then build a spanning forest of G by connecting each vertex v to its neighbor in G at which the estimator $\tilde{f}$ is highest. If all the neighbors of v have lower $\tilde{f}$-values than v, then v is connected to itself and declared a peak of $\tilde{f}$: it thus becomes the root of some tree in the forest, and as such a cluster center. As mentioned above and illustrated in Figure 3(d), this construction is very sensitive to perturbations of the function: this instability becomes deadly in pratice since density estimators tend to be noisy. The novelty of our approach resides in the way we use persistence to guide the merging of the clusters during our second phase, and thus regain some stability: given a parameter τ, we merge every cluster of prominence less than τ into its parent cluster in the hierarchy defined by persistence. Both the prominences and the hierarchy can be computed on the fly during the first phase, provided that the vertices are processed in an order prescribed by $\tilde{f}$. In fact, the second phase itself can be done simultaneously to the first phase, as will be seen in Section 3. The output of the algorithm is a collection of (merged) clusters whose prominences are at least τ. Additional feedback is provided in the form



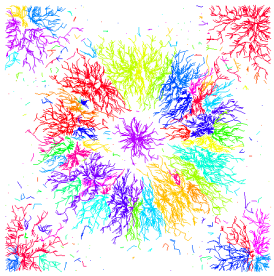
(a) input point cloud (10,000 pts).



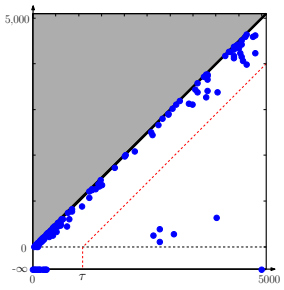
(b) approximate density function $\tilde{f}$ at the data points.



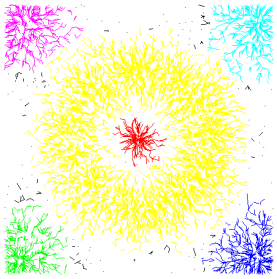
(c) Rips graph built on top of the point cloud, and its upper-star filtration induced by $\tilde{f}$.



(d) clusters obtained from the simple hill-climbing scheme performed in the Rips graph, before any merge.



(e) persistence diagram showing six points corresponding to the six prominent peaks of f, plus a myriad of close-to-diagonal or late-appearing points corresponding to topological noise in the upper-star filtration.



(f) result obtained after merging the clusters of Figure 3(d) according to the τ-thresholded persistence diagram of Figure 3(e). The small black clusters are independent connected components of the Rips graph that appear very lately in the upper-star filtration and can therefore be treated as noise.

Figure 3: Illustration of our clustering method on a point cloud (a) drawn in i.i.d. fashion from some unknown probability density function f: after preprocessing (b and c) the data by applying a density estimator $\tilde{f}$ and building the upper-star filtration of the Rips graph induced by $\tilde{f}$, we perform our two main operations sequentially: first, the algorithm of Section 3 is applied with $\tau = +\infty$ to approximate the persistence diagram $\mathrm{D}_0 f$ and determine a relevant value for parameter τ (e); second, the algorithm of Section 3 is applied again with the new value of τ to produce the final clustering (f). The result of the simple hill-climbing scheme of [24] (which corresponds to applying the algorithm of Section 3 without any merge ($\tau = 0$)) is shown for comparison (d).

of a persistence diagram that coincides[2] with $\mathrm{D}_0\tilde{f}$ when parameter τ is set to $+\infty$ (that is, when every cluster is merged into its parent in the hierarchy).

Parameters δ and τ are of very different natures. As in the basic graph-based gradient ascent algorithm [24], parameter δ controls the spatial scale at which the input point cloud must be processed. In practice it may be difficult to tune

[2]Recall that the estimator $\tilde{f}$ is viewed as a function $G \to \mathbb{R}$ here.

without any prior knowledge of the data, however dendrograms provided by single-linkage clustering can greatly help in this task. Differently, parameter τ controls the degree of prominence above which a peak of the density is considered as meaningful. Relevant values for this parameter can be inferred by the user from the diagram output by the algorithm, which suggests a bootstrapping approach in practice, illustrated in Figure 3: in a first stage, the algorithm is run with τ set to $+\infty$, in order to compute $\mathrm{D}_0\tilde{f}$; then, in a second stage, the algorithm is re-run with the value of τ picked up by the user from the persistence diagram.

The validity of this approach is guaranteed by a sound theoretical framework. Provided that the input point cloud is large enough and that a suitable choice of parameter δ is made, our adaptation of the result of [6] guarantees that the diagram computed by the algorithm is close to the one of the true density function f. Assuming that there is a clear gap between prominent and non-prominent peaks of the density in $\mathrm{D}_0 f$, we can prove that the same kind of gap appears in the diagram computed during the first run of the algorithm, as it is the case for instance in Figure 3(e). This means that the user can easily find a value for parameter τ within the range of admissible values, so that the second run of the algorithm will produce a number of clusters that corresponds exactly to the number of significant peaks of the density (Theorem 4.8). This gives a theoretically-sound meaning to what we mean by *correct* number of clusters. In addition to this guarantee on the number of clusters, we can correlate to some extent the spatial locations of our clusters with the basins of attraction of the corresponding prominent peaks of f (Theorem 4.9), as suggested by Figure 3(f). These results are detailed in Sections 4 and 5 of the paper.

In order to illustrate the practicality of our clustering scheme, we provide in Section 6 a series of experimental results obtained in several applications, including segmenting color images and classifying protein configurations. Beyond these sample applications, our algorithm can be used in a large variety of contexts, including when the data are massive, high-dimensional, or non-Euclidean. Such versatility and effectiveness are made possible by the following two properties of our approach:

- It only requires to know the (approximate) pairwise distances between the data points, as well as rough estimates of the density at these points. It is therefore virtually applicable in any arbitrary metric space.
- In the meantime, its complexity remains reasonable: although the size of the input distance matrix may be up to quadratic in the number n of data points, our implementation only uses an amount of main memory that is linear in n, and it has a running time of $O(n + m\alpha(n))$, where m is the number of edges in the neighborhood graph G and α is the inverse Ackermann function. This means that a practical run of the program barely takes more time than the one necessary to read the input distance matrix.

2 Mathematical Background

Our analysis uses singular homology with coefficients in a commutative ring, assumed to be a field throughout the paper and omitted in the notations. We also use some elements of Riemannian geometry and Morse theory. We refer the reader to [19, 20, 25] for comprehensive introductions to these topics.

2.1 Riemannian manifolds and probability density functions

Throughout the paper, and unless otherwise stated, $\mathbb{X}$ denotes a Riemannian manifold possibly with boundary, and $\mathrm{d}_{\mathbb{X}}$ denotes its geodesic distance. Given a point $x \in \mathbb{X}$ and a real value $r \geq 0$, let $B_{\mathbb{X}}(x,r)$ denote the closed geodesic ball of center x and radius r, namely: $B_{\mathbb{X}}(x,r) = \{y \in \mathbb{X},\ \mathrm{d}_{\mathbb{X}}(x,y) \leq r\}$. For all sufficiently small values $r \geq 0$, the ball $B_{\mathbb{X}}(x,r)$ is known to be *strongly convex*, that is: for every pair of points y, y' in $B_{\mathbb{X}}(x,r)$, there exists a unique shortest path in $\mathbb{X}$ between y and y', and this path is included in $B_{\mathbb{X}}(x,r)$. Let $\varrho_c(x) > 0$ be the supremum of the radii such that this property holds. The infimum of $\varrho_c(x)$ over the points of $\mathbb{X}$ is known as the *strong convexity radius* of $\mathbb{X}$, noted $\varrho_c(\mathbb{X})$. This quantity is positive for instance when $\mathbb{X}$ is compact [19, §2.89] or when $\mathbb{X} = \mathbb{R}^m$.

Given an m-dimensional Riemannian manifold $\mathbb{X}$, we call $\mathcal{H}^m$ the m-dimensional Hausdorff measure determined by the Riemannian metric of $\mathbb{X}$ [1, §5.5]. By *probability density function over* $\mathbb{X}$ *with respect to* $\mathcal{H}^m$ we mean a non-negative function $f : \mathbb{X} \to \mathbb{R}$ that is integrable with respect to $\mathcal{H}^m$ and such that $\int_{\mathbb{X}} f \, d\mathcal{H}^m = 1$. In the rest of the paper, all probability density functions will be understood as being defined over $\mathbb{X}$ with respect to $\mathcal{H}^m$.

2.2 Filtrations and Persistent Homology

Persistent homology is one of the central concepts used in the paper. It was first introduced by Edelsbrunner *et al.* [17] and later developped in [18, 35]. It has proven to be a powerful tool for data analysis, as reported in two recent surveys [2, 16]. We only give a brief description here, and refer the reader to these surveys for further details.

A *filtration* $\mathcal{X}$ of a topological space $\mathbb{X}$ is a finite sequence of nested subspaces $\emptyset = \mathbb{X}^{\alpha_m} \subseteq \mathbb{X}^{\alpha_{m-1}} \subseteq \cdots \subseteq \mathbb{X}^{\alpha_1} \subseteq \mathbb{X}^{\alpha_0} = \mathbb{X}$, where $\alpha_m > \alpha_{m-1} > \cdots > \alpha_1 > \alpha_0$ is a decreasing sequence of real numbers. The inclusion maps between the subspaces induce a directed system of vector spaces, called a *persistence module*, involving their k-dimensional homology groups:

$$H_k(\mathbb{X}^{\alpha_m}) \xrightarrow{\phi^m_{m-1}} H_k(\mathbb{X}^{\alpha_{m-1}}) \xrightarrow{\phi^{m-1}_{m-2}} \cdots \xrightarrow{\phi^2_1} H_k(\mathbb{X}^{\alpha_1}) \xrightarrow{\phi^1_0} H_k(\mathbb{X}^{\alpha_0}). \qquad (1)$$

The structure of this persistence module can be encoded as a multi-set $\mathrm{D}_k\mathcal{X}$, called the *k-th persistence diagram* of $\mathcal{X}$, and defined as follows: $\mathrm{D}_k\mathcal{X}$ is a multi-set of points in the extended plane $\overline{\mathbb{R}}^2$, where $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$, contained in the union of the extended diagonal $\Delta = \{(x,x) : x \in \overline{\mathbb{R}}\}$ and of the grid $\{+\infty = \alpha_\infty, \alpha_m, \alpha_{m-1}, \cdots, \alpha_1, \alpha_0\} \times \{\alpha_m, \alpha_{m-1}, \cdots, \alpha_1, \alpha_0\}$. The multiplicity of the points of Δ is set to $+\infty$, whereas the multiplicities of the (α_i, α_j), $+\infty \geq i > j \geq 0$, are defined in terms of the ranks of the homomorphisms $\phi^i_j = \phi^i_{i-1} \circ \cdots \circ \phi^{j+1}_j$ (see e.g. [3]). Intuitively, each point (p_x, p_y) of $\mathrm{D}_k\mathcal{X}$ encodes the lifespan of some k-dimensional homological feature appearing at time p_x and dying at time $p_y \leq p_x$ in the filtration $\mathcal{X}$. Note that we depart from the usual way of introducing persistence by reversing the time flow, which goes from $+\infty$ to $-\infty$ here. As mentioned in the introduction, this choice is purely formal and does not affect the validity of the theory. Its motivation will become clear thereafter.

In this paper we consider uncountable sequences of nested spaces, defined as the superlevel-sets of some continuous function $f : \mathbb{X} \to \mathbb{R}$. Specifically, the subspace of $\mathbb{X}$ of index $\alpha \in \mathbb{R}$ is the closed superlevel-set $\mathbb{F}^\alpha = [\alpha, +\infty)$. Here, the sequence of spaces is indexed over whole $\mathbb{R}$, and no longer over a finite subset. However, the notion of k-th persistence diagram $\mathrm{D}_k f$ can be extended to this continuous setting under some *tameness* condition [10] stating basically that the family of inclusions $\mathbb{F}^\alpha \subseteq \mathbb{F}^\beta$ for $\alpha \geq \beta$ induces at k-dimensional homology level a persistence module of the same finite type as in Eq. (1). Under this condition, the persistence diagram $\mathrm{D}_k f$ contains only finitely many points off the extended diagonal Δ.

In the context of clustering, we are primarily concerned with 0-dimensional homology, which encodes the path-connectivity of spaces. As far as 0-dimensional homology is concerned, assuming that the filtration defined by the superlevel-sets of f is tame boils down to assuming that f has only finitely many peaks. In the rest of the paper, we will restrict our focus to the 0-dimensional homology level to simplify the exposition and make our algorithms and theoretical results more intuitive. It should be noted however that our approach is more general and ultimately enables to compute the topology of the clusters, and not only their number and locations.

In order to make effective computations, we will build discrete structures on top of point clouds. Since we are mainly concerned with 0-dimensional homology, our structures will be simple unoriented graphs, as opposed to general abstract simplicial complexes when higher-dimensional homologies are concerned. A type of graph that will play a central role in our work is the so-called *Rips graph*, also known as *δ-neighborhood graph* in the literature:

Definition 2.1 *Given a finite point cloud L in a metric space $(\mathbb{X}, \mathrm{d}_\mathbb{X})$, and a parameter $\delta > 0$, the* Rips graph $R_\delta(L, \mathrm{d}_\mathbb{X})$ *is the graph of vertex set L whose edges correspond to the pairs of points $x, y \in L$ such that $\mathrm{d}_\mathbb{X}(x, y) \leq \delta$.*

In the rest of the paper, the choice of the metric $\mathrm{d}_\mathbb{X}$ will be obvious and therefore omitted in our notations. Given a real-valued function $f : \mathbb{X} \to \mathbb{R}$ and a parameter $\alpha \in \mathbb{R}$, let L^α denote the trace of the superlevel-set $\mathbb{F}^\alpha$ over the point cloud L, that is:

$$L^\alpha = L \cap \mathbb{F}^\alpha. \tag{2}$$

The *upper-star Rips filtration*, noted $\mathcal{R}_\delta^f(L)$, is the following nested family of subgraphs of $R_\delta(L)$:

$$\mathcal{R}_\delta^f(L) = \{R_\delta(L^\alpha)\}_{\alpha \in \mathbb{R}}, \tag{3}$$

where parameter α ranges from $+\infty$ to $-\infty$. The name *upper-star filtration* stems from the fact that whenever a vertex $v \in L$ enters the filtration, its whole upper star (*i.e.* the set of edges of $R_\delta(L)$ connecting v to other vertices with higher function values) enters at the same time. Observe that, even though parameter α ranges over whole $\mathbb{R}$, the connectivity of the subgraph $R_\delta(L^\alpha)$ only changes when a new vertex v is added, at time $\alpha = f(v)$. As a result, the filtration $\mathcal{R}_\delta^f(L)$ is composed of a finite family of different graphs, therefore it induces a persistence module of the same finite type as in Eq. (1) at homology level.

2.3 0-Dimensional Persistence of Morse Functions

Consider an m-dimensional Riemannian manifold $\mathbb{X}$ and a real-valued function $f : \mathbb{X} \to \mathbb{R}$ that is assumed to be of Morse type, *i.e.* at least C^2-continuous with non-degenerate critical points. Assume also that f has a finite number of critical points. The *ascending region* of a critical point m, noted $A(m)$, is the subset of the points of $\mathbb{X}$ that eventually reach m by moving along the flow induced by the gradient vector field of f. For all $x \in A(m)$, we call m the *root* of x. Ascending regions of the peaks of f are known to form pairwise-disjoint open cells homeomorphic to $\mathbb{R}^m$. Furthermore, assuming $\mathbb{X}$ to have no boundary and f to be bounded from above and proper[3], the ascending regions of the peaks of f cover $\mathbb{X}$ up to a subset of Hausdorff measure zero. It is then natural to use them to partition (almost all) the space $\mathbb{X}$ into regions of influence.

For any $x \in \mathbb{X}$ and $\alpha \in \mathbb{R}$, let $C(x, \alpha) \subseteq \mathbb{F}^\alpha$ denote the path-connected component of the superlevel-set $\mathbb{F}^\alpha$ that contains x. Morse theory tells us that when a local maximum m_p of f enters the superlevel-sets filtration, at time $\alpha = f(m_p)$, a new path-connected component $C(m_p, \alpha)$ appears in the superlevel-set $\mathbb{F}^\alpha$. In homological terms, the peak m_p is called the *generator* of the component born at time $f(m_p)$. $C(m_p, \alpha)$ ceases to be an independent connected component in $\mathbb{F}^\alpha$ when it gets connected to another component generated by a higher peak m_q. At that particular time α, persistence theory tells us that the component $C(m_p, \alpha)$ gets *merged* into $C(m_q, \alpha)$. While m_q remains the generator of the component $C(m_q, \alpha)$, m_p ceases to be a generator, and by analogy we call m_q its root, noted $m_q = r(m_p)$. In the 0-th persistence diagram $\mathrm{D}_0 f$, the lifespan of m_p as a generator is encoded by the point p of coordinates $p_x = f(m_p)$ and $p_y = \alpha \leq p_x$. The difference $\tau = p_x - p_y \geq 0$ between birth and death times is called the *prominence* of the peak m_p. Equivalently, we say that p is *τ-prominent*.

Given a thresholding parameter $\tau \geq 0$, we restrict our focus to the peaks m_p of f of prominence at least τ. Intuitively, the points of $\mathbb{X}$ that are *attracted* by m_p are the ones belonging to ascending regions that are eventually merged by persistence into the connected component of m_p before being merged into the component of any other peak of prominence at least τ. Formally, for every peak m_q of f (of arbitrary prominence), let us iterate the *root map* $m_q \mapsto r(m_q)$ until some peak of prominence at least τ is reached[4]. We call r^*_τ the thus iterated root map, and we point out that every peak of prominence at least τ is a fixed point of r^*_τ. The *basin of attraction* of m_p (of parameter τ) is defined as the union of the ascending regions of all the peaks mapped to m_p through r^*_τ:

$$\forall m_p \text{ s.t. } p_x - p_y \geq \tau,\ B_\tau(m_p) = \bigcup_{r^*_\tau(m_q) = m_p} A(m_q). \tag{4}$$

Note that $B_\tau(m_p)$ contains $A(m_p)$ since m_p is a fixed point of r^*_τ. More precisely, we have $A(m_p) = B_0(m_p) \subseteq B_\tau(m_p)$. In addition, since the iterated root map $m_q \mapsto r^*_\tau(m_q)$ is uniquely defined, the basins of attraction form a partition of the union of all ascending regions.

[3] This means that for any bounded closed interval $[a, b] \subset \mathbb{R}$, the pre-image $f^{-1}([a, b])$ is a compact subset of $\mathbb{X}$.

[4] Such a prominent peak is always reached, since the tame function f has finitely many peaks and since the root map satisfies $f(m_q) < f(r(m_q))$, meaning that $r(m_q)$ is more prominent than m_q.

3 Algorithm

Our clustering algorithm takes as input a n-dimensional vector f with real coordinates, a $n \times n$ symmetric matrix D with non-negative real coefficients, and two real parameters $\delta, \tau \geq 0$. The n dimensions represent the n points of a point cloud L; the vector represents a function $f : L \to \mathbb{R}$, while the entries $D_{i,j} = D_{j,i}$ of the matrix give the distance between the i-th and j-th points of L. No geographic coordinates are assumed to be given, so that the algorithm can be applied virtually in any metric space. However, for the sake of our proofs, we will later assume that the point cloud L lies on some unknown Riemannian manifold $\mathbb{X}$, such that the matrix D encodes the pairwise geodesic distances between the points of L, while f encodes the values (at the points of L) of the probability density function according to which the dataset has been generated. Details on how these quantities can be estimated in practice are provided in Section 5.

In a preprocessing step, our algorithm computes the Rips graph $R_\delta(L)$ from the input D and δ. Then, the main phase of our algorithm consists in mimicking within the Rips graph the construction of the basins of attraction of parameter τ described in Section 2.3. We proceed as follows:

1. First, we iterate over the points of L by decreasing function values: at each vertex i, we approximate the gradient of the underlying probability density function by connecting i to its neighbor in the graph $R_\delta(L)$ with highest function value. If all neighbors of i haver lower function values, then i is declared a peak of f and its gradient nullified. The resulting collection of gradients forms a spanning forest of the graph $R_\delta(L)$: each tree in this forest can be viewed as the analog within the graph $R_\delta(L)$ of the ascending region of a peak in the continuous setting.
2. Second, to handle merges between trees, we maintain a *union-find* data structure [12, Chapter 21] where each entry corresponds to a union of trees of the spanning forest. We call *root* of an entry e, or $r(e)$ for short, the vertex contained in e whose function value is highest. By construction, this vertex must be the root of one of the trees contained in e, and therefore a peak of f in the graph $R_\delta(L)$. The merge of an entry into another entry is the analog in our discrete setting to the merge of a basin of attraction into another basin in the continuous setting. Merges are performed in the order prescribed by persistence. More precisely, we iterate once again over the vertices of $R_\delta(L)$ by decreasing order of function values, considering at each vertex i the edges of the upper star of i in $R_\delta(L)$. Letting e_i be the entry of the union-find data structure containing i, if any edge of the upper star of i connects e_i to some other entry e_j whose root $r(e_j)$ has lower function value than the root $r(e_i)$, then the persistence algorithm prescribes that e_j be merged into e_i: we depart from this prescription and perform the merge only if the prominence of $r(e_j)$ (viewed as a peak of f in the graph $R_\delta(L)$) is less than the threshold τ. This condition comes down to checking whether $f_{r(e_j)} - f_i < \tau$. Once all non-prominent neighboring clusters have been merged into e_i, we check whether e_i itself should be merged. Letting $\bar{e}$ be the neighboring cluster with highest root, we merge e_i into $\bar{e}$ if and only if the prominence of $r(e_i)$ is less than τ, *i.e.* $f_{r(e_i)} - f_i < \tau$.

The pseudo-codes for steps 1. and 2. of our algorithm are given in Procedures 1 and 2 below. As shown in Procedure 1, both steps can be performed during a single pass over the vertices of $R_\delta(L)$: for each considered vertex i, the

approximate gradient at i is computed, then the possible merges in the union-find data structure are operated[5]. The neighborhood graph $R_\delta(L)$ itself does not have to be pre-computed, since only the upper star of i is involved when vertex i is processed.

Procedure 1 `Clustering`

Input: n-dimensional vector f, $n \times n$ symmetric matrix D, real parameters $\delta, \tau \geq 0$.

Sort the index set L so that $f_1 \leq f_2 \leq \cdots \leq f_n$;
Initialize the union-find data structure $\mathcal{U}$;
for $i = n$ to 1 **do**
  compute the upper star $S_i = \{(i, j_1), \cdots, (i, j_k)\}$ of vertex i in $R_\delta(L)$;
  if $S_i = \emptyset$ **then** {*vertex i is a local maximum of f within $R_\delta(L)$*}
    $g(i) \leftarrow null$; {*$g(i)$ stores the approximate gradient at vertex i*}
    Create a new entry in $\mathcal{U}$ containing the tree $\{i\}$;
  else {*vertex i is not a local maximum of f within $R_\delta(L)$*}
    $g(i) \leftarrow \arg\max_{j \in \{j_1, \cdots, j_k\}} f(j)$;
    Attach vertex i to the tree t containing $g(i)$;
    $\mathcal{U} \leftarrow$ `Merge` $(f, \mathcal{U}, i, S_i, \tau)$;
  end if
end for

Output: the set of entries e of $\mathcal{U}$ satisfying $f_{r(e)} \geq \tau$.

Upon termination, the algorithm outputs the collection of entries of the union-find data structure, which partitions the input point cloud L into clusters. In fact, it only outputs those entries e whose root $r(e)$ satisfies $f_{r(e)} \geq \tau$. This additional filtering step is motivated by the fact that some outliers in the point cloud L may form independent connected components in the graph $R_\delta(L)$ that cannot be merged, as shown in Figures 3(c) and 3(f). Such connected components generate entries in the union-find data structure that have very low (density) function values and can therefore be discarded using the above filtering criterion, illustrated in Figure 3(e).

Step 2. of our algorithm provides additional feedback in the form of a collection of intervals, each representing the lifespan of an entry in the union-find data structure (the endpoints of the interval correspond to the creation and merge times of the entry). When parameter τ is set to infinity, step 2. becomes nothing but the standard persistence algorithm applied to the upper-star filtration $\mathcal{R}^f_\delta(L)$, therefore the output collection of intervals coincides with the 0-th persistence diagram of this filtration. We will see in Section 4 that this diagram faithfully approximates the persistence diagram of the underlying density function under some mild conditions on the input.

This observation suggests a two-stages recipe to cluster point cloud data in practice, illustrated in Figure 3: in a first stage, the clustering algorithm is run with $\tau = +\infty$ to approximate the persistence diagram of the underlying density function; this diagram is used as feedback by the user to choose a relevant value for parameter τ, which is then fed to the algorithm in a second stage to produce the desired number of clusters.

[5]These involve only previously visited vertices.

Procedure 2 `Merge`

Input: n-dimensional vector f, union-find data structure $\mathcal{U}$, integer i, integer list $S = \{j_1, \cdots, j_k\}$, a parameter $\tau \geq 0$.

Let e_i be the entry of $\mathcal{U}$ containing i;
{*find entries of* $\mathcal{U}$ *intersecting* S *whose roots are less than* τ*-prominent; merge those into* e_i}
for $j \in \{j_1, \cdots, j_k\}$ **do**
  Let e_j be the entry of $\mathcal{U}$ containing j;
  if $e_j \neq e_i$ and $f_{r(e_j)} - f_i < \tau$ **then**
    Remove entry e_j from $\mathcal{U}$ and attach it to e_i;
  end if
end for
{*find entry* $\bar{e}$ *of* $\mathcal{U}$ *intersecting* S *whose root is highest*}
$\bar{e} \leftarrow$ *null*;
for $j \in \{j_1, \cdots, j_k\}$ **do**
  Let e_j be the entry of $\mathcal{U}$ containing j;
  if $\bar{e} =$ *null* or $f_{r(e_j)} > f_{r(\bar{e})}$ **then**
    $\bar{e} \leftarrow e_j$;
  end if
end for
{*merge* e_i *into* $\bar{e}$ *if the prominence of the root of* e_i *is less than* τ}
if $\bar{e} \neq e_i$ and $f_{r(e_i)} - f(i) < \tau$ **then**
  Remove entry e_i from $\mathcal{U}$ and attach it to $\bar{e}$;
end if

Output: updated union-find data structure $\mathcal{U}$.

Running time and main memory usage. As mentioned above, the neighborhood graph $R_\delta(L)$ does not have to be pre-computed and stored, since only the star of the node being processed is involved at each step of the algorithm. This means that the main memory usage is $O(n)$, where n is the size of the input point cloud. In addition, each vertex of $R_\delta(L)$ creates a new entry in the union-find data structure $\mathcal{U}$, while each edge of $R_\delta(L)$ generates two finds plus potentially one union in $\mathcal{U}$. Since there are n vertices and $m = O(n^2)$ edges, there cannot be more than $n - 1$ unions and $2m$ finds, therefore the total running time of the algorithm is $O(n + m\alpha(n))$, where α is the inverse Ackermann function. In many practical scenarios, such as the ones considered in Section 6, parameter δ is chosen small enough so that $m = O(n)$. The running time of the algorithm becomes then almost-linear in the size of the point cloud, excluding the time spent reading the input.

4 Theoretical Guarantees

Throughout our analysis (Sections 4 and 5), we assume $\mathbb{X}$ to be an m-dimensional Riemannian manifold with positive convexity radius, and $f : \mathbb{X} \to \mathbb{R}$ to be a c-Lipschitz probability density function with respect to the m-dimensional Hausdorff measure. We further assume that the input point cloud L has been sampled over $\mathbb{X}$ according to f in i.i.d. fashion.

In this section we assume for simplicity that the values of f at the points of L, as well as the pairwise geodesic distances between the points of L, are given as input to the algorithm. The analysis of practical scenarios where geodesic distances or function values are unknown and need to be approximated is deferred to Section 5. Both sections make an extensive use of the following concept of *geodesic ε-sample*:

Definition 4.1 *Given a subset $\mathbb{Y} \subseteq \mathbb{X}$ and a parameter $\varepsilon > 0$, L is a* geodesic ε-sample of $\mathbb{Y}$ *if every point of $\mathbb{Y}$ lies within geodesic distance ε of L, that is: $\forall y \in \mathbb{Y}$, $\min_{v \in L} \mathrm{d}_{\mathbb{X}}(y, v) \leq \varepsilon$.*

They also rely on a *well-separatedness* condition applied to the 0-th persistence diagram of f, as defined below and illustrated in Figure 4:

Definition 4.2 *Given two values $d_2 > d_1 \geq 0$, the persistence diagram $\mathrm{D}_0 f$ is called* (d_1, d_2)-separated *if every point of $\mathrm{D}_0 f$ lies either in the region D_1 above the diagonal line $y = x - d_1$ or in the region D_2 below the diagonal $y = x - d_2$ and to the right of the vertical line $x = d_2$.*

This condition makes precise the intuitive notion that the points of the persistence diagram can be separated between prominent peaks (region D_2) and topological noise (region D_1). This acts very similarly to a signal-to-noise ratio condition: the larger the difference $d_2 - d_1$, the more clearly we can separate the prominent peaks from the noise. In the limit scenario where $d_1 = 0$, all peaks of f must be at least d_2-prominent and none of them is considered as noise. The additional condition that the points of D_2 must lie to the right of the vertical line $x = d_2$ is purely technical and will be explained in Section 4.4.

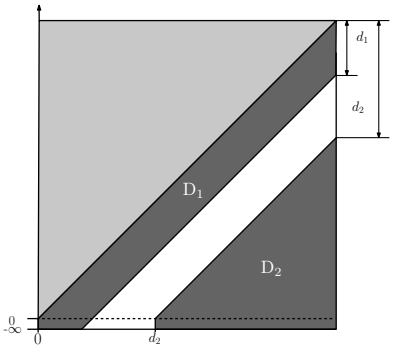


Figure 4: The separation of the persistence diagram $\mathrm{D}_0 f$ between prominent peaks (region D_2) and topological noise (region D_1).

4.1 Overview of the results of the section

Our first main result relates the number of clusters computed by the algorithm to the number of prominent peaks of f. Using the stabillity of persistence diagrams to relate the diagram of f to the diagram ouptut by step 2. of the algorithm, we can prove that the regions D_1 and D_2 remain disjoint under pertubations caused by our approximation, and can therefore be separated using some thresholding parameter τ (to be determined by the user). With such a value of parameter

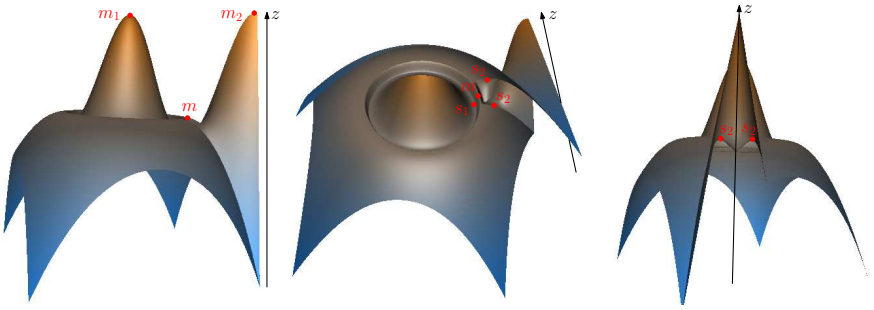


Figure 5: A function f with unstable basins of attraction, defined over the unit square $[0,1]^2 \subset \mathbb{R}^2$. For a persistence threshold $\tau > f(m) - f(s_2)$, the ascending region of the peak m is merged into the basin of attraction of the peak m_2 of parameter τ. However, since $f(s_2) - f(s_1)$ is arbitrarily small, $A(m)$ can be merged into $B_\tau(m_1)$ instead under small perturbations of f.

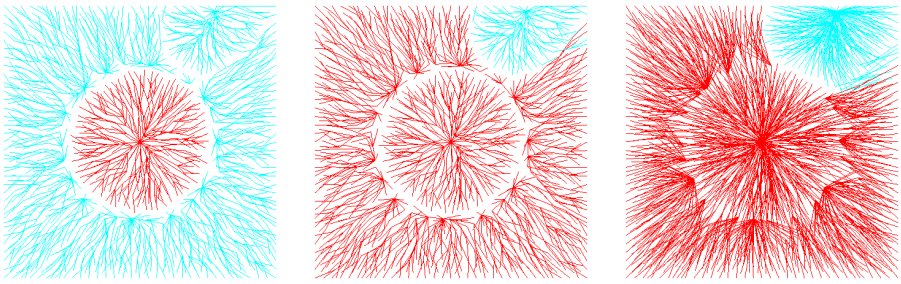


Figure 6: Outputs of the algorithm obtained from a uniform ε-sample L of the unit square ($\varepsilon = 0.15$) endowed with the function f of Fig. 5. We chose a value of τ that gives two clusters, and we used three different values for the Rips parameter: $\delta = 0.27$ (left), $\delta = 0.28$ (center), $\delta = 0.6$ (right). Notice how some values of δ induce a correct merge of $A(m)$ into $B_\tau(m_2)$ whereas others induce an incorrect merge of $A(m)$ into $B_\tau(m_1)$. The limit value of ε below which no such failure of the algorithm occurs depends on the arbitrarily small quantity $f(s_2) - f(s_1)$.

τ as input, the algorithm computes the correct number of clusters with high probability:

Result 1 (Theorem 4.8) *If $\mathrm{D}_0 f$ is well-separated and the number n of input points is large enough, then there exist values for the Rips parameter δ and the thresholding parameter τ such that the number of clusters computed by the algorithm is equal to the number of peaks of f of prominence at least τ with high probability.*

Explicit bounds are given in the full statement of the theorem. Once it is known that there exist some values for parameters δ, τ that make the algorithm separate prominent peaks from noise in the persistence diagram of f, it is natural to ask how such values can be found in practice. This aspect will be discussed in the experimental Section 6.

Another question is how well the output of the algorithm approximates the basins of attraction of the prominent peaks over the point cloud, assuming that f is of Morse type. In full generality, this is a hopeless question since the basins of attaction are not stable even in the smooth case. There are indeed many examples of very close Morse functions having very different basins of attraction, and clearly the algorithm cannot provably-well approximate the unstable parts of the basins. An illustrative example is given in Figures 5 and 6. Nevertheless, we can guarantee that the algorithm does provably well approximate some stable parts of the basins:

Result 2 (Theorem 4.9) *If $\mathrm{D}_0 f$ is well-separated and the number of input points is large enough, then there exist values of the Rips parameter δ and thresholding parameter τ such that, for each peak m of f of prominence at least τ, with high probability the algorithm outputs a cluster that coincides (over the point cloud L) with the basin of attraction $B_\tau(m)$ up to the time $\alpha_\tau(m)$ when $B_\tau(m)$ gets connected to the basin of another peak of f of prominence at least τ.*

As shown by the example of Figure 5, the basin $B_\tau(m)$ may start being unstable as soon as time $\alpha_\tau(m)$, therefore Theorem 4.9 means that the algorithm provides approximations to the basins of attractions that are the best possible in the worst case. Our proof of the theorem also shows an important fact, namely: that each basin of attraction $B_\tau(m)$ is stable under small perturbations of the function f, at least between times $f(m)$ and $\alpha_\tau(m)$. This fact opens the door to a more statistical approach to clustering: since we know the top parts of the basins (and therefore of the clusters computed by the algorithm) are stable under small perturbations of the function, we can conduct multiple runs of the algorithm with random pertubations of the input data, and then find correspondences between the outputs of different runs. Each point can then be assigned a quantitative measure of its classification stability over the runs. This aspect is deferred to the conclusion Section 7, as it lies somewhat beyond the scope of the paper.

Note finally that our main results are inherently probabilistic. This is not due to the algorithm itself, which is deterministic, but rather to the simple fact that the input point set L must form a dense sampling of some superlevel-set of f for the algorithm to have a chance of approximating $\mathrm{D}_0 f$ accurately, which can only occur with high probability since the points of L are sampled at random. This point will be addressed in Section 4.2. We will then introduce some background results in scalar field analysis (Section 4.3), which will be instrumental in proving Theorems 4.8 and 4.9 (Sections 4.4 and 4.5 respectively).

4.2 Sampling superlevel-sets of a probability density function

As mentioned previously, our main results rely on the property that the input point cloud L forms a geodesic ε-sample of some superlevel-set of f. Intuitively, since the points of L are drawn according to f in i.i.d. fashion, the more points are drawn the more chances we have that L satisfies the above property. This simple fact is proved formally in Lemma 4.3 below. Before stating the lemma, we need to introduce a few measure-theoretic concepts. For any subset A of

$\mathbb{X}$ and any parameter $r > 0$, we let $\mathcal{V}_r(A)$ be the infimum of the Hausdorff measures achieved by geodesic balls of radius r centered in A, namely:

$$\mathcal{V}_r(A) = \inf_{x \in A} \mathcal{H}^m(B_{\mathbb{X}}(x,r)) \geq 0, \text{ where } B_{\mathbb{X}}(x,r) = \{y \in \mathbb{X},\ \mathrm{d}_{\mathbb{X}}(x,y) \leq r\}. \quad (5)$$

Let also $\mathcal{N}_r(A) \in \mathbb{N} \cup \{+\infty\}$ be the *r-covering number* of A, that is, the minimum number of closed geodesic balls of same radius r needed to cover A (the balls do not have to be centered in A).

Lemma 4.3 *Let $\mathbb{X}$ be an m-dimensional Riemannian manifold, and $f : \mathbb{X} \to \mathbb{R}$ a c-Lipschitz probability density function. Consider a set L of n points sampled according to f in i.i.d. fashion. Then, for any parameters $\varepsilon > 0$ and $\alpha > c\varepsilon$, we are guaranteed that L forms an ε-sample of $\mathbb{F}^\alpha$ with probability at least $1 - \mathcal{N}_{\varepsilon/2}(\mathbb{F}^\alpha)\ e^{-n(\alpha - c\varepsilon)\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)}$.*

This result can be interpreted in various different ways:

- When the probability density function f is provided and a fixed superlevel-set $\mathbb{F}^\alpha$ is considered, the lemma ensures that after drawing sufficiently many points according to f in i.i.d. fashion the superlevel set $\mathbb{F}^\alpha$ will be densely sampled with high probability.
- Conversely, when the set L of sample points is fixed and a target sampling parameter ε is given, the lemma ensures that for large enough values[6] of α the superlevel-set $\mathbb{F}^\alpha$ is ε-sampled by L with high probability. In particular, α has to be larger than $c\varepsilon$.

In both scenarios, the probability of success is influenced by two quantities that are intrinsic to the Riemannian manifold $\mathbb{X}$: the covering number $\mathcal{N}_{\varepsilon/2}(\mathbb{F}^\alpha)$, and the minimum geodesic ball measure $\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)$. Note that the probability of success can be positive only when $\mathcal{N}_{\varepsilon/2}(\mathbb{F}^\alpha)$ is finite and $\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)$ is positive. These conditions are met in a wide range of settings, including:

a) $\mathbb{X}$ is compact. In this case, $\mathbb{F}^\alpha$ is compact since it is a closed subset of $\mathbb{X}$, f being continuous and $[\alpha, +\infty)$ being closed. Then, $\mathcal{N}_{\varepsilon/2}(\mathbb{F}^\alpha)$ is trivially finite. Moreover, since the map $x \mapsto \mathcal{H}^m(B_{\mathbb{X}}(x, \varepsilon/2))$ is continuous, the infimum in Eq. (5) is a minimum, which by the Area Formula [26, §3.7] is positive. Therefore, $\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha) > 0$.

b) $\mathbb{X}$ has bounded absolute sectional curvature[7]. In this case, the Bishop-Gunter inequality [19, Theorem 3.101] ensures that for any $r > 0$, the Hausdorff measures of the geodesic balls of same radius r are bounded from below by the same positive quantity. As a consequence, $\mathcal{V}_r(\mathbb{F}^\alpha) \geq \mathcal{V}_r(\mathbb{X}) > 0$. In addition, since f is a probability density function, we have:

$$1 = \int_{\mathbb{X}} f\ d\mathcal{H}^m \geq \int_{\mathbb{F}^\alpha} f\ d\mathcal{H}^m \geq \alpha\ \mathcal{H}^m(\mathbb{F}^\alpha),$$

which implies that $\mathcal{H}^m(\mathbb{F}^\alpha)$ is finite when $\alpha > 0$. Combining this with the fact that $\mathcal{V}_{\varepsilon/4}(\mathbb{F}^\alpha) > 0$, we deduce that no more than $\frac{\mathcal{H}^m(\mathbb{F}^\alpha)}{\mathcal{V}_{\varepsilon/4}(\mathbb{F}^\alpha)} < +\infty$ pairwise-disjoint geodesic balls of same radius $\frac{\varepsilon}{4}$ can be packed inside $\mathbb{F}^\alpha$, which by the Kolmogorov-Tikhomirov inequality [23] implies that $\mathcal{N}_{\varepsilon/2}(\mathbb{F}^\alpha) < +\infty$.

[6] As α grows, $\mathcal{N}_{\varepsilon/2}(\mathbb{F}^\alpha)$ decreases while $\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)$ increases, therefore the probability of success increases.

[7] This case occurs *e.g.* when $\mathbb{X}$ is the Euclidean space $\mathbb{R}^m$.

Our proof of Lemma 4.3 is an easy application of the union bound:

Proof of Lemma 4.3. If $\mathcal{N}_{\varepsilon/2}(\mathbb{F}^\alpha) = +\infty$ or $\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha) = 0$, then the lower bound provided by the lemma is non-positive and therefore the conclusion holds trivially.

Assume from now on that $\mathcal{N}_{\varepsilon/2}(\mathbb{F}^\alpha) < +\infty$ and $\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha) > 0$. Consider a family $\{B_i\}_{1\leq i\leq l}$ of closed geodesic balls of same radius $\frac{\varepsilon}{2}$ such that $\mathbb{F}^\alpha \subseteq \bigcup_{i=1}^l B_i$ and $l = \mathcal{N}_{\varepsilon/2}(\mathbb{F}^\alpha)$ is minimal. Let p_i be a point of $B_i \cap \mathbb{F}^\alpha$. Such a point exists because otherwise the cover would not be minimal. Since f is c-Lipschitz, at every point $p \in B_i$ we have $f(p) \geq f(p_i) - c\, \mathrm{d}_{\mathbb{X}}(p, p_i) \geq \alpha - c\varepsilon > 0$. Therefore,

$$\forall i \in \{1, \cdots, l\},\ \int_{B_i} f\, d\mathcal{H}^m \geq (\alpha - c\varepsilon)\mathcal{H}^m(B_i) \geq (\alpha - c\varepsilon)\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha).$$

Let E_i denote the event that $L \cap B_i = \emptyset$. The probability with which this event occurs is

$$\mathrm{P}[E_i] = \left(1 - \int_{B_i} f\, d\mathcal{H}^m\right)^n \leq \left(1 - (\alpha - c\varepsilon)\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)\right)^n.$$

Then, by the union bound, we have

$$\mathrm{P}[\cup_i E_i] \leq \sum_{i=1}^{l} \mathrm{P}[E_i] \leq l\left(1 - (\alpha - c\varepsilon)\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)\right)^n.$$

Note that when $\cup_i E_i$ does not occur, every ball B_i contains at least one point of L, therefore by the triangle inequality L is an ε-sample of $\mathbb{F}^\alpha$. Hence, our goal is to work out an upper bound on $\mathrm{P}[\cup_i E_i]$. Observe that the quantity $g(x) = e^{-x} + x - 1$ is non-negative for all $x \geq 0$. Letting x be equal to $(\alpha - c\varepsilon)\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)$, we obtain:

$$1 - (\alpha - c\varepsilon)\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha) \leq e^{-(\alpha - c\varepsilon)\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)},$$

which implies that

$$\mathrm{P}[\cup_i E_i] \leq l\left(1 - (\alpha - c\varepsilon)\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)\right)^n \leq l\ e^{-n(\alpha - c\varepsilon)\mathcal{V}_{\varepsilon/2}(\mathbb{F}^\alpha)}.$$

□

4.3 Background on scalar field analysis over point cloud data

From now on, we endow the extended plane $\overline{\mathbb{R}}^2$ with the l^∞-norm, noted $\|\cdot\|_\infty$. Given two multi-subsets A, B of $\overline{\mathbb{R}}^2$, a *multi-bijection* γ between A and B is a bijection

$$\gamma : \bigcup_{p\in|A|} \coprod_{i=1}^{\mu(p)} p \to \bigcup_{q\in|B|} \coprod_{i=1}^{\mu(q)} q,$$

where $|A|$ denotes the *support* of A, *i.e.* the set A considered as a subset of $\overline{\mathbb{R}}^2$ without any multiplicities, and where $\mu(p)$ denotes the multiplicity of point $p \in |A|$ in A. The *bottleneck distance* $\mathrm{d}_B^\infty(A, B)$ between A and B is the quantity

$\min_\gamma \max_{p\in A} \|p-\gamma(p)\|_\infty$, where γ ranges over all multi-bijections between A and B. The bottleneck distance is a natural measure of proximity between persistence diagrams [10]. With this concept at hand, we can introduce a result from [6] that will play a central role in our analysis[8]:

Theorem 4.4 *Let $\mathbb{X}$ be a compact Riemannian manifold, possibly with boundary, and $f : \mathbb{X} \to \mathbb{R}$ a tame c-Lipschitz function. Let also L be a geodesic ε-sample of $\mathbb{X}$. If $\varepsilon < \frac{1}{4}\varrho_c(\mathbb{X})$, then for any $\delta \in [4\varepsilon,\ \varrho_c(\mathbb{X}))$, the bottleneck distance between the 0-dimensional persistence diagrams of f and of the upper-star Rips filtration $\mathcal{R}^f_\delta(L)$ is at most $c\delta$.*

This result suffers from two major limitations that make it inapplicable as is to our context:

1. the point cloud L must be dense over the entire manifold $\mathbb{X}$, which is not true when the data points are drawn from some non-uniform probability distribution;
2. the manifold $\mathbb{X}$ must be compact, which prohibits simple scenarios such as $\mathbb{X} = \mathbb{R}^m$.

Theorem 4.5 below addresses these two issues provided that the point cloud L forms a dense sampling of some superlevel-set of the function f, as guaranteed by Lemma 4.3. In the statement of the theorem, Q^{NE}_α, Q^{SE}_α , Q^{SW}_α , and Q^{NW}_α denote respectively the quadrants $(\alpha,+\infty]\times(\alpha,+\infty]$, $(\alpha,+\infty]\times[-\infty,\alpha]$, $[-\infty,\alpha]\times[-\infty,\alpha]$, and $[-\infty,\alpha]\times(\alpha,+\infty]$ in the extended plane $\overline{\mathbb{R}}^2$.

Theorem 4.5 *Let $\mathbb{X}$ be a Riemannian manifold, possibly non-compact, possibly with boundary. Assume that the convexity radius $\varrho_c(\mathbb{X})$ is positive. Let $L \subseteq \mathbb{X}$ be a finite point cloud and $f : \mathbb{X} \to \mathbb{R}$ be a tame c-Lipschitz function. Then, for any positive $\delta < \varrho_c(\mathbb{X})$, for any $\alpha \in \mathbb{R}$ such that L is a geodesic $\frac{\delta}{4}$-sample of $\mathbb{F}^\alpha = f^{-1}([\alpha,\infty))$, there is a multi-bijection γ between the 0-th persistence diagrams of f and of the upper-star Rips filtration $\mathcal{R}^f_\delta(L)$, such that:*

(i) $\forall p \in \mathrm{D}_0 f \cap Q^{\mathrm{NE}}_\alpha,\ \|p-\gamma(p)\|_\infty \le c\delta$.

(ii) $\forall q \in \mathrm{D}_0\mathcal{R}^f_\delta(L) \cap Q^{\mathrm{NE}}_\alpha,\ \|\gamma^{-1}(q)-q\|_\infty \le c\delta$.

(iii) $\forall p \in \mathrm{D}_0 f \cap Q^{\mathrm{SE}}_\alpha,\ |p_x-\gamma(p)_x| \le c\delta$.

(iv) $\forall q \in \mathrm{D}_0\mathcal{R}^f_\delta(L) \cap Q^{\mathrm{SE}}_\alpha,\ |\gamma^{-1}(q)_x-q_x| \le c\delta$.

The theorem is illustrated in Figure 7 (left). Assertions (i)-(ii) ensure that the multi-bijection γ does not move the points of both diagrams by more than $c\delta$ within the upper-right quadrant Q^{NE}_α corresponding to the superlevel-set of f that is $\frac{\delta}{4}$-sampled by L. In cases where L is a $\frac{\delta}{4}$-sample of the entire manifold $\mathbb{X}$ ($\alpha = -\infty$), assertions (i)-(ii) imply that the bottleneck distance between both persistence diagrams is at most $c\delta$, as stated in Theorem 4.4.

Assertions (iii)-(iv) provide weaker guarantees in the lower-right quadrant Q^{SE}_α, by ensuring that every 0-dimensional homology class $[c]$ appearing at time $\alpha_b > \alpha$ in the superlevel-sets filtration of f must appear within $[\alpha_b - c\delta, \alpha_b + c\delta]$ in the filtration $\mathcal{R}^f_\delta(L)$, and vice-versa. By contrast, death times are not

[8] The result of [6] holds in fact for persistence diagrams of arbitrary dimensions, but it uses two upper-star Rips filtrations in parallel: $\mathcal{R}^f_{\delta/2}(L)$ and $\mathcal{R}^f_\delta(L)$. As reported in the research report version of that paper [5, §4.3], in the special case of 0-dimensional homology, using both filtrations or only $\mathcal{R}^f_\delta(L)$ gives the same result.

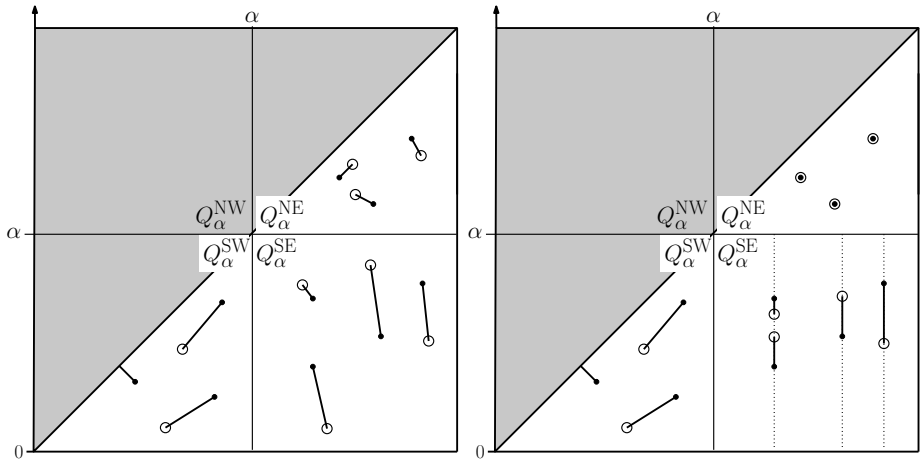


Figure 7: Left: the multi-bijection of Theorem 4.5. Right: for the proof of Lemma 4.6.

fully controlled: if $[c]$ dies at time $\alpha_d < \alpha$ in the superlevel-set filtration of f, then all we can say is that its death time in $\mathcal{R}_\delta^f(L)$ must be less than $\alpha + c\delta$, since otherwise by (ii) $[c]$ would be located in Q_α^{NE}, thereby contradicting the assumption that $[c] \in Q_\alpha^{\mathrm{SE}}$.

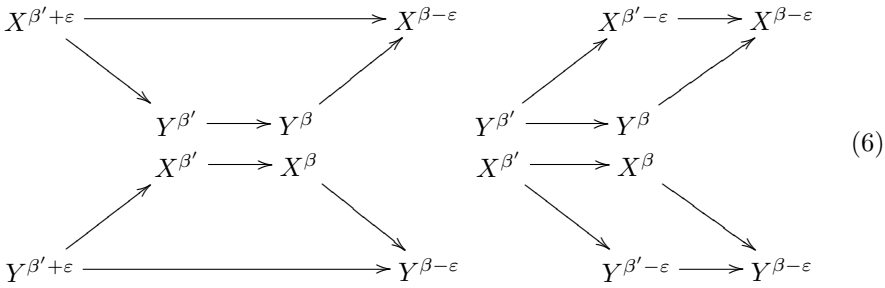
Due to the lack of sample points outside the superlevel-set $\mathbb{F}^\alpha$, there is no guarantee concerning the portion of $\mathrm{D}_0 f$ lying in the quadrants Q_α^{NW} and Q_α^{SW} located to the left of the vertical line $x = \alpha$. This part of the diagram corresponds indeed to homological features appearing at times less than α in the superlevel-sets filtration of f, which may not be captured at all in $\mathcal{R}_\delta^f(L)$.

The proof of Theorem 4.5 relies on the following technical result, inspired from recent advances on the stability of persistence diagrams [3], whose purely algebraic proof is deferred to Appendix A:

Lemma 4.6 *Let $\mathcal{X}$ and $\mathcal{Y}$ be two tame persistence modules that are* (strongly) ε-interleaved above some given time $\alpha \in \mathbb{R}$. *Then, there is a multi-bijection $\gamma : \mathrm{D}\mathcal{X} \to \mathrm{D}\mathcal{Y}$ satisfying assertions* (i) *through* (iv) *of Theorem 4.5, with $\mathrm{D}_0 f$ replaced by $\mathrm{D}\mathcal{X}_\mathbb{R}$ and $\mathrm{D}_0\mathcal{R}_\delta^f(L)$ replaced by $\mathrm{D}\mathcal{Y}_\mathbb{R}$.*

Here, $\mathcal{X}$ and $\mathcal{Y}$ stand for two persistence modules $\{X^\alpha\}_{\alpha\in\mathbb{R}}$ and $\{Y^\alpha\}_{\alpha\in\mathbb{R}}$, indexed over whole $\mathbb{R}$, that are of same finite type as in Eq. (1). The notion of *(strong) ε-interleaving above time α* is derived from [3]. It means that there exist two families of homomorphisms $\{\phi_\beta : X^\beta \to Y^{\beta-\varepsilon}\}_{\beta\geq\alpha}$ and $\{\psi_\beta : Y^\beta \to X^{\beta-\varepsilon}\}_{\beta\geq\alpha}$, such that for all values $\beta' \geq \beta \geq \alpha$ the following diagrams of vector

spaces commute:



(6)

Intuitively, the commutativity of these diagrams means that every homological feature appearing (resp. dying) in $\mathcal{X}$ at some time $\beta \geq \alpha$ must appear (resp. die) in $\mathcal{Y}$ within $[\beta - \varepsilon, \beta + \varepsilon]$, and vice-versa. This statement is the analog of assertions (i)-(ii) of Theorem 4.5. Furthermore, every homological feature appearing in $\mathcal{X}$ at time $\beta_b \geq \alpha$ and dying at time $\beta_d \leq \alpha$ must appear within $[\beta_b - \varepsilon, \beta_b + \varepsilon]$ and die at some time below $\alpha + \varepsilon$ in $\mathcal{Y}$, and vice-versa. This statement is the analog of assertions (iii)-(iv) of Theorem 4.5. Thus, the conclusion of Lemma 4.6 is intuitively clear.

Proof of Theorem 4.5. With Lemma 4.6 at hand, the proof of the theorem becomes a straightforward adaptation of the proof of Theorem 4.4 given in [6]. Indeed, the same exact sequence of arguments as in [6, §3.1] shows that there exist two families of homomorphisms $\{\phi_\beta : H_0(R_\delta(L^\beta)) \to H_0(\mathbb{F}^{\beta - c\delta})\}_{\beta \geq \alpha}$ and $\{\psi_\beta : H_0(\mathbb{F}^\beta) \to H_0(R_\delta(L^{\beta - c\delta}))\}_{\beta \geq \alpha}$ that make the persistence modules $\{H_0(\mathbb{F}^\beta)\}_{\beta \in \mathbb{R}}$ and $\{H_0(R_\delta(L^\beta))\}_{\beta \in \mathbb{R}}$ (strongly) $c\delta$-interleaved above time α. It follows then from Lemma 4.6 that there is a multi-bijection $\gamma : \mathrm{D}_0 f \to \mathrm{D}_0 \mathcal{R}^f_\delta(L)$ satisfying assertions (i) through (iv) of the theorem. □

4.4 Estimating the number of prominent peaks

In this section, we prove that if the peaks of the density function f are prominent enough compared to the noise, the algorithm will recover the correct number of clusters. To state the result formally, we need to define some notation for partitioning the persistence diagram of f.

In the extended plane $\bar{\mathbb{R}}^2$, let Δ denote the diagonal $y = x$. For any $d > 0$, we call Δ_d the line $y = x - d$, parallel to Δ, lying below Δ, at l^∞ distance $\frac{d}{2}$ of Δ. Let Δ^{S}_d denote the closed half-plane lying below Δ_d, and Δ^{N}_d the open half-plane lying above Δ_d. Similarly, we call Λ^{W}_d (resp. Λ^{E}_d) the closed (resp. open) half-plane lying to the left (resp. right) of the vertical line $x = d$, and Λ^{S}_d (resp. Λ^{N}_d) the closed (resp. open) half-plane lying below (resp. above) the horizontal line $y = d$. Definition 4.2 can now be restated as follows:

Definition 4.7 *Given two values $d_2 > d_1 \geq 0$, the persistence diagram of f is called* (d_1, d_2)-separated *if it has the following structure:*

$$\mathrm{D}_0(f) = \mathrm{D}_1 \cup \mathrm{D}_2, \text{ where } \mathrm{D}_1 \subset \Delta^{\mathrm{N}}_{d_1} \text{ and } \mathrm{D}_2 \subset \Delta^{\mathrm{S}}_{d_2} \cap \Lambda^{\mathrm{E}}_{d_2}.$$

As mentioned at the beginning of Section 4, the condition that $\mathrm{D}_0 f$ is partitioned into two disjoint subsets $\mathrm{D}_1 \subset \Delta^{\mathrm{N}}_{d_1}$ and $\mathrm{D}_2 \subset \Delta^{\mathrm{S}}_{d_2}$ with $d_2 > d_1$ can be

interpreted as a signal-to-noise ratio condition: the relevant peaks of f (in D_2) must be significantly more prominent than the non-relevant ones (in D_1) for the algorithm to be able to detect the correct number of clusters. The additional condition that $\mathrm{D}_2 \subset \Lambda^{\mathrm{E}}_{d_2}$ stems from the observation that only some superlevel-set $\mathbb{F}^\alpha$ of f can be densely sampled by the input point set L drawn according to f, as expressed in Lemma 4.3. Due to a lack of sample points outside $\mathbb{F}^\alpha$, the persistence diagram of the upper-star Rips filtration built by the algorithm cannot be controlled in the region $\Lambda^{\mathrm{W}}_\alpha$, as illustrated in Figure 3(e). This region must therefore be discarded by the algorithm, and the condition $\mathrm{D}_2 \subset \Lambda^{\mathrm{E}}_{d_2}$ simply states that the prominent peaks of f must reach high enough altitudes so as not to be discarded themselves.

Theorem 4.8 *Let $\mathbb{X}$ be a Riemannian manifold with positive convexity radius, and let $f : \mathbb{X} \to \mathbb{R}$ be a tame c-Lipschitz probability density function. If $\mathrm{D}_0(f)$ is (d_1, d_2)-separated, with $d_2 > d_1 \geq 0$, then for any positive parameter $\delta < \min\{\varrho_c(\mathbb{X}),\ \frac{d_2-d_1}{5c}\}$ and any threshold $\tau \in (d_1 + 2c\delta,\ d_2 - 3c\delta)$, on any input of n sample points drawn according to f in an i.i.d. fashion the number of clusters computed by the algorithm is equal to the number of peaks of f of prominence at least d_2 with probability at least $1 - \mathcal{N}_{\delta/8}(\mathbb{F}^{c\delta})e^{-n\frac{3}{4}c\delta\mathcal{V}_{\delta/8}(\mathbb{F}^{c\delta})}$.*

Proof. Let $\alpha = c\delta$ and $\varepsilon = \delta/4$. According to Lemma 4.3, the input point set L forms a $\frac{\delta}{4}$-sample of the superlevel-set $\mathbb{F}^{c\delta}$ with probability at least $1-\mathcal{N}_{\delta/8}(\mathbb{F}^{c\delta})e^{-n\frac{3}{4}c\delta\mathcal{V}_{\delta/8}(\mathbb{F}^{c\delta})}$. Assume from now on that L is indeed a $\frac{\delta}{4}$-sample of $\mathbb{F}^{c\delta}$. By Theorem 4.5, there is a multi-bijection $\gamma : \mathrm{D}_0 f \to \mathrm{D}_0\mathcal{R}^f_\delta(L)$ satisfying conditions (i) through (iv) of Theorem 4.5. Let us prove that under these conditions the diagram of $\mathcal{R}^f_\delta(L)$ is separated into two parts, one of which is in (multi-)bijection with the set of peaks of f of prominence at least d_2. The proof requires us to analyze where an arbitrary point p of $\mathrm{D}_0(f)$ can be mapped to by γ. Our analysis is split into five different cases, depending on which region of Figure 8(a) point p belongs to. We first consider Regions I and II, which correspond to cases where $p \in \mathrm{D}_1$:

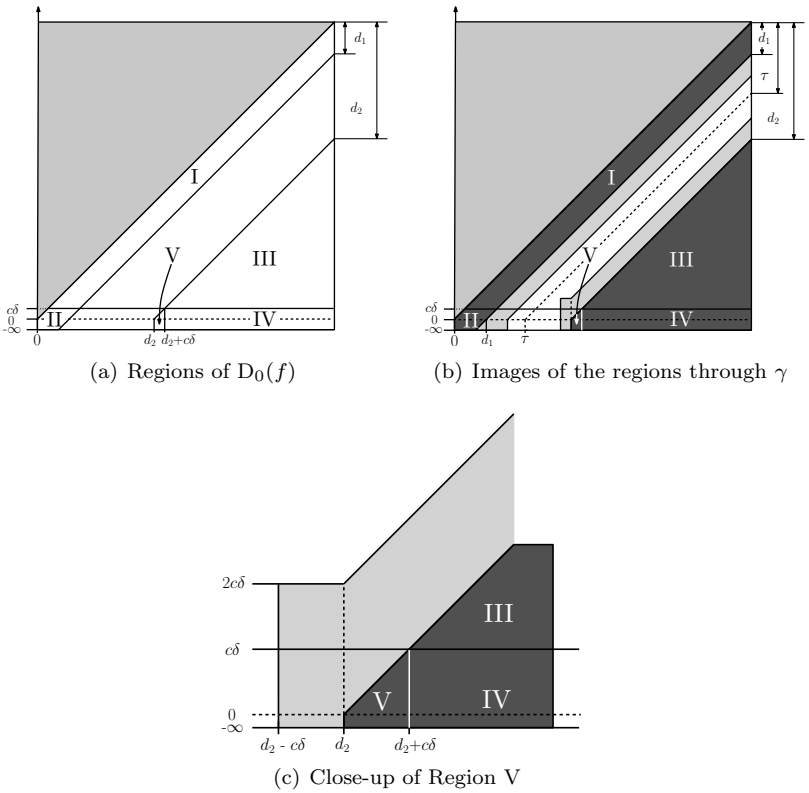
- p lies in Region I, *i.e.* $p \in \Delta^{\mathrm{N}}_{d_1} \cap \Lambda^{\mathrm{N}}_{c\delta}$. Then, we have $p \in Q^{\mathrm{NE}}_{c\delta}$, and (i) implies that $||p - \gamma(p)||_\infty \leq c\delta$. Therefore, $\gamma(p) \in \Delta^{\mathrm{N}}_{d_1+2c\delta}$.
- p lies in Region II, *i.e.* $p \in \Delta^{\mathrm{N}}_{d_1} \cap \Lambda^{\mathrm{S}}_{c\delta}$. Then, a quick computation (see Figure 8(b)) shows that p lies in $\Lambda^{\mathrm{W}}_{d_1+c\delta}$. If $\gamma(p)$ were located in $\Lambda^{\mathrm{E}}_{d_1+2c\delta}$, then (iv) would imply that $p = \gamma^{-1}(\gamma(p)) \in \Lambda^{\mathrm{E}}_{d_1+c\delta}$, thereby raising a contradiction. Therefore, $\gamma(p) \in \Lambda^{\mathrm{W}}_{d_1+2c\delta}$.

It follows that $\gamma(\mathrm{D}_1) \subseteq \Delta^{\mathrm{N}}_{d_1+2c\delta} \cup \Lambda^{\mathrm{W}}_{d_1+2c\delta}$. We now proceed with Regions III, IV, V, which correspond to cases where $p \in \mathrm{D}_2$, and we show that under our assumptions their images through γ do not intersect $\Delta^{\mathrm{N}}_{d_1+2c\delta} \cup \Lambda^{\mathrm{W}}_{d_1+2c\delta}$:

- p lies in Region III, *i.e.* $p \in \Delta^{\mathrm{S}}_{d_2} \cap \Lambda^{\mathrm{N}}_{c\delta}$. Then, we have $p \in Q^{\mathrm{NE}}_{c\delta}$ and therefore $||\gamma(p)-p||_\infty \leq c\delta$, by (i). This implies that $\gamma(p) \in \Delta^{\mathrm{S}}_{d_2-2c\delta} \cap \Lambda^{\mathrm{E}}_{d_2}$, since $\Delta^{\mathrm{S}}_{d_2} \cap \Lambda^{\mathrm{N}}_{c\delta} \subset \Delta^{\mathrm{S}}_{d_2} \cap \Lambda^{\mathrm{E}}_{d_2+c\delta}$. Now, $\Delta^{\mathrm{S}}_{d_2-2c\delta} \cap \Lambda^{\mathrm{E}}_{d_2}$ is disjoint from $\Delta^{\mathrm{N}}_{d_1+2c\delta} \cup \Lambda^{\mathrm{W}}_{d_1+2c\delta}$ because by hypothesis we have $d_2 > d_1 + 4c\delta$.
- p lies in Region IV, *i.e.* $p \in \Lambda^{E}_{d_2+c\delta} \cap \Lambda^{\mathrm{S}}_{c\delta}$. Then, (iii) implies that $\gamma(p) \in \Lambda^{\mathrm{E}}_{d_2}$. In addition, we have $\gamma(p) \in \Lambda^{\mathrm{S}}_{2c\delta}$ since otherwise $\gamma(p)$ would belong to $Q^{\mathrm{NE}}_{c\delta}$ and by (ii) $p = \gamma^{-1}(\gamma(p))$ would belong to $\Lambda^{\mathrm{N}}_{c\delta}$, a contradiction. Thus, we have $\gamma(p) \in \Lambda^{\mathrm{E}}_{d_2} \cap \Lambda^{\mathrm{S}}_{2c\delta}$, which is disjoint from $\Delta^{\mathrm{N}}_{d_1+2c\delta} \cup \Lambda^{\mathrm{W}}_{d_1+2c\delta}$ since by hypothesis we have $d_2 > d_1 + 4c\delta$.

- p lies in Region V, *i.e.* $p \in \Delta^{\mathrm{S}}_{d_2} \cap \Lambda^{\mathrm{E}}_{d_2} \cap \Lambda^{\mathrm{W}}_{d_2+c\delta}$. Then, p belongs to $Q^{\mathrm{SE}}_{c\delta}$, therefore (iii) implies that $\gamma(p) \in \Lambda^{\mathrm{E}}_{d_2-c\delta}$. In addition, $\gamma(p)$ must lie in $\Lambda^{\mathrm{S}}_{2c\delta}$ or we have a contradiction by (ii) as in the previous case. Hence, $\gamma(p) \in \Lambda^{\mathrm{E}}_{d_2-c\delta} \cap \Lambda^{\mathrm{S}}_{2c\delta}$, which is disjoint from $\Delta^{\mathrm{N}}_{d_1+2c\delta} \cup \Lambda^{\mathrm{W}}_{d_1+2c\delta}$ since by hypothesis we have $d_2 > d_1 + 5c\delta$.

Thus, the persistence diagram $\mathrm{D}_0\mathcal{R}^f_\delta(L)$ is partitioned into two disjoint subsets: $\mathrm{D}^{\mathcal{R}}_1$ and $\mathrm{D}^{\mathcal{R}}_2$, which are the respective images of D_1 and D_2 through γ, and which lie respectively in the disjoint regions $\gamma(\mathrm{I} \cup \mathrm{II})$ and $\gamma(\mathrm{III} \cup \mathrm{IV} \cup \mathrm{V})$, as depicted in Figure 8(b). Then, for any choice of parameter τ within the range $(d_1 + 2c\delta,\ d_2 - 3c\delta)$, the subset $\mathrm{D}^{\mathcal{R}}_2$ (as well as D_2) is located in the region $\Delta^{\mathrm{S}}_\tau \cap \Lambda^{\mathrm{E}}_\tau$, whereas $\mathrm{D}^{\mathcal{R}}_1$ (as well as D_1) is located in its complement $\Delta^{\mathrm{N}}_\tau \cup \Lambda^{\mathrm{W}}_\tau$. This implies that the algorithm discards $\mathrm{D}^{\mathcal{R}}_1$ and keeps only $\mathrm{D}^{\mathcal{R}}_2$, which has same (finite) total multiplicity as D_2 since both sets contain no point of the diagonal Δ and are in multi-bijection. This concludes the proof, since D_2 represents precisely the set of peaks of f of prominence at least d_2. □



(a) Regions of $\mathrm{D}_0(f)$

(b) Images of the regions through γ

(c) Close-up of Region V

Figure 8: For the proof of Theorem 4.8.

4.5 Approximating the basins of attraction of the prominent peaks

The next natural question is whether the clusters output by the algorithm are faithful approximations to the actual basins of attraction of the underlying probability density function f. Using the terminology of Section 2.3, given a parameter $\tau \geq 0$ and a peak m_p of f of prominence at least τ, we call basin of attraction of m_p of parameter τ, noted $B_\tau(m_p)$, the union of the ascending regions of all the peaks mapped to m_p through the iterated root map r_τ^*, as per Eq. (4). Recall that the root map r takes a peak of f and maps it to the peak of higher prominence such that the connected component generated by the first peak in the superlevel-sets filtration of f gets merged by persistence into the component generated by the second peak. The iterated root map r_τ^* iterates this process until some peak of prominence at least τ is reached. Given such a peak m_p, we call $\alpha_\tau(m_p)$ the time at which the connected component generated by m_p gets connected to the one generated by another peak of prominence at least τ. Assuming $\mathrm{D}_0 f$ to be (d_1, d_2)-separated and τ to lie within the range $[d_1, d_2]$, we have the following inequalities:

$$\forall m_p \text{ s.t. } p_x - p_y \geq \tau,\ p_x - d_2 \geq \alpha_\tau(m_p) \geq p_y. \tag{7}$$

The first inequality follows from the fact that for any peak $m_q \neq m_p$ of prominence at least τ, $C(m_p, \alpha)$ and $C(m_q, \alpha)$ cannot get connected with each other above time $\alpha = p_x - d_2$, because otherwise the prominence of the younger connected component would be less than d_2 and therefore less than τ since $\mathrm{D}_0 f$ is (d_1, d_2)-separated. The second inequality follows from the fact that, at time p_y, $C(m_p, p_y)$ is merged into some older connected component $C(q, p_y)$ such that $q_x - q_y \geq p_x - p_y \geq \tau$.

As reported in the overview Section 4.1, guaranteeing that the entire basins of attraction of the prominent peaks are approximated is hopeless. However, Theorem 4.9 gives a partial approximation guarantee (where we abuse notations by letting $B_\tau(p) = B_\tau(m_p)$ and $\alpha_\tau(p) = \alpha_\tau(m_p)$):

Theorem 4.9 *Let $\mathbb{X}$ be a Riemannian manifold with positive convexity radius, and let $f : \mathbb{X} \to \mathbb{R}$ be a tame c-Lipschitz probability density function. If $\mathrm{D}_0(f)$ is (d_1, d_2)-separated, with $d_2 > d_1 \geq 0$, then for any positive parameter $\delta < \min\{\varrho_c(\mathbb{X}),\ \frac{d_2 - d_1}{5c}\}$ and any threshold $\tau \in (d_1 + 2c\delta,\ d_2 - 3c\delta)$, on any input L of n sample points drawn according to f in an i.i.d. fashion the following is true with probability at least $1 - \mathcal{N}_{\delta/8}(\mathbb{F}^{c\delta}) e^{-n \frac{3}{4} c\delta \mathcal{V}_{\delta/8}(\mathbb{F}^{c\delta})}$: for each point $p \in \mathrm{D}_2$ there is a cluster $B_\tau^{\mathcal{R}}(p)$ output by the algorithm such that $B_\tau^{\mathcal{R}}(p) \cap \mathbb{F}^\alpha = B_\tau(p) \cap L \cap \mathbb{F}^\alpha$ at all times $\alpha \in (\alpha_\tau(p) + d_1 + \frac{5}{2} c\delta,\ p_x]$.*

In plain words, the conclusion of the theorem means that, within the superlevel-set $\mathbb{F}^\alpha$, the cluster $B_\tau^{\mathcal{R}}(p)$ is the *trace* of the basin of attraction $B_\tau(p)$ over the point cloud L. This holds from the time p_x at which the basin $B_\tau(p)$ appears in the superlevel-sets filtration of f, almost until the time $\alpha_\tau(p)$ at which $B_\tau(p)$ ceases to be disconnected from the other basins of attraction in the filtration. In view of Eq. (7), the duration of this phase is at least $d_2 - d_1 - \frac{5}{2} c\delta > 0$, which as in Theorem 4.8 can be interpreted as a *signal-to-noise* ratio condition. As explained in Section 4.1 and illustrated in Figures 5 and 6, below time $\alpha_\tau(p)$ it is not possible to guarantee the approximation of the basin of attraction $B_\tau(p)$ on all instances.

The rest of Section 4.5 is devoted to the proof of Theorem 4.9. A noticeable feature of our proof is to not rely on a precise definition of approximate gradient within the Rips graph $R_\delta(L)$, as used in the algorithm (see Procedure 1). This means that the theorem is true regardless of the actual choice of approximate gradient $g(i)$ at point i, as long as this choice satisfies $f_{g(i)} > f_i$. This is yet another indicator of the stability of our clustering approach.

Proof of Theorem 4.9. Recall from Lemma 4.3 that, under the hypotheses of Theorem 4.8, the point cloud L forms a $\frac{\delta}{4}$-sample of $\mathbb{F}^{c\delta}$ with probability at least $1 - \mathcal{N}_{\delta/8}(\mathbb{F}^{c\delta})e^{-n\frac{3}{4}c\delta\mathcal{V}_{\delta/8}(\mathbb{F}^{c\delta})}$. We assume from now on that L is indeed a $\frac{\delta}{4}$-sample of $\mathbb{F}^{c\delta}$.

The equality $B^{\mathcal{R}}_\tau(p)\cap\mathbb{F}^\alpha = B_\tau(p)\cap L\cap\mathbb{F}^\alpha$ will be proved by mutual inclusion: $B^{\mathcal{R}}_\tau(p)\cap\mathbb{F}^\alpha \subseteq B_\tau(p)\cap L\cap\mathbb{F}^\alpha$ (Lemma 4.14) and $B^{\mathcal{R}}_\tau(p)\cap\mathbb{F}^\alpha \supseteq B_\tau(p)\cap L\cap\mathbb{F}^\alpha$ (Lemma 4.15). We begin with a series of easy technical results (Lemmas 4.10 through 4.13) that will be key to proving the theorem:

Lemma 4.10 *For any $p, q \in \mathrm{D}_2$ and any $\alpha, \alpha' \in \mathbb{R}$, if $p \neq q$ then*

$$\forall x \in B_\tau(p)\cap\mathbb{F}^\alpha,\ \forall y \in B_\tau(q)\cap\mathbb{F}^{\alpha'},\ \mathrm{d}_{\mathbb{X}}(x,y) \geq \frac{\max\{\alpha-\alpha_m,\ 0\}+\max\{\alpha'-\alpha_m,\ 0\}}{c},$$

where $\alpha_m = \min\{\alpha_\tau(p),\ \alpha_\tau(q)\}$.

Proof. If $\alpha > f(m_p)$ or $\alpha' > f(m_q)$, then $B_\tau(p)\cap\mathbb{F}^\alpha = \emptyset$ or $B_\tau(q)\cap\mathbb{F}^{\alpha'} = \emptyset$ and the conclusion trivially holds. If $B_\tau(p)\cap\mathbb{F}^\alpha \neq \emptyset$ and $B_\tau(q)\cap\mathbb{F}^{\alpha'} \neq \emptyset$, then take $x \in B_\tau(p)\cap\mathbb{F}^\alpha$, $y \in B_\tau(q)\cap\mathbb{F}^\alpha$, and consider a shortest path[9] $[x,y]$ between x and y in $\mathbb{X}$. Let z be a point of $[x,y]$ where the value of f is minimal. Since $B_\tau(p)$ and $B_\tau(q)$ can only get connected to each other at time α_m, we have $f(z) \leq \alpha_m$. By the fact that f is a c-Lipschitz function, we deduce that $\mathrm{d}_{\mathbb{X}}(x,z) \geq \frac{\alpha-\alpha_m}{c}$ and $\mathrm{d}_{\mathbb{X}}(y,z) \geq \frac{\alpha'-\alpha_m}{c}$. Note that these lower bounds are negative when $\alpha, \alpha' < \alpha_m$. Since z is on a shortest path between x and y, we conclude that

$$\mathrm{d}_{\mathbb{X}}(x,y) = \mathrm{d}_{\mathbb{X}}(x,z) + \mathrm{d}_{\mathbb{X}}(z,y) \geq \frac{\max\{\alpha-\alpha_m,\ 0\}+\max\{\alpha'-\alpha_m,\ 0\}}{c}.$$

□

For any $p \in \mathrm{D}_2$, we let

$$v_p = \mathrm{argmax}_{v\in B_\tau(p)\cap L} f(v).$$

Lemma 4.11 *For any $p \in \mathrm{D}_2$, we have $f(m_p) \geq f(v_p) \geq f(m_p) - c\frac{\delta}{4}$.*

Proof. The first inequality follows from the definition of m_p as the argmax of f over $B_\tau(p)$, which contains v_p. To prove the second inequality, we use our assumption that L forms a $\frac{\delta}{4}$-sample of $\mathbb{F}^{c\delta}$ and therefore of $\mathbb{F}^{d_2}$ since $d_2 \geq c\delta$ by hypothesis. Then, because p belongs to $\mathrm{D}_2 \subset \Lambda^{\mathrm{E}}_{d_2}$, we have $m_p \in \mathbb{F}^{d_2}$

[9]Since we did not assume $\mathbb{X}$ to be a compact manifold, it may happen that no shortest path exists between x and y. However, we can always consider paths $[x,y]$ of length at most $\mathrm{d}_{\mathbb{X}}(x,y)+\zeta$, for arbitrary positive values of ζ.

and therefore there is a point $v \in L$ such that $\mathrm{d}_{\mathbb{X}}(v, m_p) \leq \delta/4$. Since f is c-Lipschitz, we have $f(v) \geq f(m_p) - c\delta/4$. To complete the proof, we only need to show that v actually lies in the basin $B_\tau(p)$, which will imply that $f(v_p) \geq f(v) \geq f(m_p) - c\delta/4$. By Lemma 4.10, the geodesic distance of m_p to $\mathbb{X} \setminus B_\tau(p)$ is at least $\frac{f(m_p)-\alpha_\tau(p)}{c} = \frac{p_x-\alpha_\tau(p)}{c}$, which by Eq. (7) is at least $\frac{d_2}{c}$, which by hypothesis is greater than 5δ. It follows then from the triangle inequality that the geodesic distance of v to $\mathbb{X} \setminus B(p)$ is strictly positive, which means that $v \in B_\tau(p)$. □

It follows from the above result that v_p is a peak of f in the Rips graph $R_\delta(L)$. Indeed, Lemma 4.11 guarantees that $f(v_p) \geq f(m_p) - c\delta/4 = p_x - c\delta/4$, which by Eq. (7) is at least $\alpha_\tau(p) + d_2 - c\delta/4$. Therefore, Lemma 4.10 ensures that the geodesic distance of v_p to $\mathbb{X} \setminus B_\tau(p)$ is at least $\frac{d_2}{c} - \frac{\delta}{4}$, which by hypothesis is greater than δ. This implies that every neighbor v of v_p in the Rips graph $R_\delta(L)$ lies in the basin $B_\tau(p)$, and by definition of v_p that $f(v) \leq f(v_p)$. Thus, v_p is a local maximum in $R_\delta(L)$. As a result, at time $f(v_p)$ a new connected component $C^{\mathcal{R}}(v_p, f(v_p))$ appears in the upper-star Rips filtration $\mathcal{R}^f_\delta(L)$, or more precisely in the subgraph $R_\delta(L \cap \mathbb{F}^\alpha)$. In homological terms, this connected component is *generated* by the peak v_p. Its lifespan is encoded as a point $p^{\mathcal{R}}$ in the persistence diagram $\mathrm{D}_0\mathcal{R}^f_\delta(L)$. Note that this point may or may not be identical to the point $\gamma(p)$ associated with p by the multi-bijection introduced in the proof of Theorem 4.8. Defining regions $\mathrm{D}^{\mathcal{R}}_1$ and $\mathrm{D}^{\mathcal{R}}_2$ as in the proof of Theorem 4.8, we have:

Lemma 4.12 *For all $p \in \mathrm{D}_2$, $p^{\mathcal{R}} \in \mathrm{D}^{\mathcal{R}}_2$.*

Proof. At any time $\alpha \in (\alpha_\tau(p) + c\delta/2, f(v_p)]$, Lemma 4.10 guarantees that every point of $L \cap \mathbb{F}^\alpha \cap B_\tau(p)$ (including v_p itself) is disconnected from every point of $L \cap \mathbb{F}^\alpha \setminus B_\tau(p)$ in the subgraph $R_\delta(L \cap \mathbb{F}^\alpha)$, therefore the connected component $C^{\mathcal{R}}(v_p, \alpha)$ is included in $B_\tau(p)$. This implies that v_p remains the argmax of f over $C^{\mathcal{R}}(v_p, \alpha)$, and therefore that $C^{\mathcal{R}}(v_p, \alpha)$ still exists as an independent connected component in the subgraph $R_\delta(L \cap \mathbb{F}^\alpha)$. It follows that $p^{\mathcal{R}}_y \leq \alpha_\tau(p) + c\delta/2$, which in turn implies that $p^{\mathcal{R}}_x - p^{\mathcal{R}}_y \geq f(v_p) - \alpha_\tau(p) - c\delta/2$. By Lemma 4.11, this quantity is at least $f(m_p) - \alpha_\tau(p) - 3c\delta/4 = p_x - \alpha_\tau(p) - 3c\delta/4$, which by Eq. (7) is at least $d_2 - 3c\delta/4$. Thus, $p^{\mathcal{R}}$ lies in $\Delta^{\mathrm{S}}_{d_2-3c\delta/4} \subset \Delta^{\mathrm{S}}_{d_2-3c\delta}$. In addition, we have $p^{\mathcal{R}}_x = f(v_p) \geq f(m_p) - \frac{c\delta}{4} = p_x - \frac{c\delta}{4}$, which is at most $d_2 - c\frac{\delta}{4}$ since by hypothesis $p \in \mathrm{D}_2 \subset \Lambda^{\mathrm{E}}_{d_2}$. Hence, $p^{\mathcal{R}}$ also lies in $\Lambda^{\mathrm{E}}_{d_2-c\delta/4} \subset \Lambda^{\mathrm{E}}_{d_2-3c\delta}$, which proves that $p^{\mathcal{R}} \in \mathrm{D}^{\mathcal{R}}_2$ since $d_2 > d_1 + 5c\delta$. □

According to Lemma 4.12, $p \mapsto p^{\mathcal{R}}$ is a map $\mathrm{D}_2 \to \mathrm{D}^{\mathcal{R}}_2$. This map is clearly injective, since by definition $p^{\mathcal{R}}$ corresponds to the connected component of $\mathcal{R}^f_\delta(L)$ generated by the peak v_p which belongs to the basin $B_\tau(p)$ and to no other. In fact, the map is bijective since by Theorem 4.8 the cardinalities of D_2 and $\mathrm{D}^{\mathcal{R}}_2$ are the same. Another important consequence of Lemma 4.12 is that v_p is in fact the generator of a whole cluster output by the algorithm. We call $B^{\mathcal{R}}_\tau(p)$ this cluster.

Given a point $x \in L$, we denote by $r(x)$ the root of the tree to which x is attached in the forest built at step 1. of the algorithm of Section 3. For each merge of an entry e into another entry e' performed in the union-find data structure at step 2. of the algorithm, we call e' the root of e, noted $e' = r(e)$. We can then iterate the root map, starting at x, until we reach the

root of the cluster containing x in the output of the algorithm. This root is denoted $r^*_\tau(x)$, by analogy with the continuous setting described in Section 2.3. By construction, $r^*_\tau(x)$ is the only peak (of f within the Rips graph $R_\delta(L)$) of prominence at least τ in its cluster. Therefore, in the persistence diagram $\mathrm{D}_0\mathcal{R}^f_\delta(L)$, $r^*_\tau(x)$ corresponds to some point $[r^*_\tau(x)] \in \mathrm{D}^{\mathcal{R}}_2$. Let $p \in \mathrm{D}_2$ be such that $p^{\mathcal{R}} = [r^*_\tau(x)]$. Such a point exists since the map $p \mapsto p^{\mathcal{R}}$ is a bijection $\mathrm{D}_2 \to \mathrm{D}^{\mathcal{R}}_2$. The cluster of root $r^*_\tau(x)$ output by the algorithm is then identified with $B^{\mathcal{R}}_\tau(p)$, and the root itself is identified with v_p.

Lemma 4.13 $\forall x \in L$, $\forall \alpha \leq f(x) - d_1 - 2c\delta$, $C^{\mathcal{R}}(x, \alpha) = C^{\mathcal{R}}(r^*_\tau(x), \alpha)$.

Proof. By definition of the root $r(x)$, there is a path from x to $r(x)$ in the Rips graph $R_\delta(L)$ such that f increases along this path. This means that x and $r(x)$ belong to the same connected component of the subgraph $R_\delta(L \cap \mathbb{F}^{f(x)})$. Since $\alpha \leq f(x)$, we deduce that $C^{\mathcal{R}}(x, \alpha) = C^{\mathcal{R}}(r(x), \alpha)$.

For convenience, we let $x_0 = r(x)$, $x_1 = r(x_0)$, $\cdots$, $x_{l-1} = r(x_{l-2})$, and $x_l = r(x_{l-1}) = r^*_\tau(x)$. We have $f(x_l) \geq f(x_{l-1}) \geq \cdots \geq f(x_0) \geq f(x)$. By construction, the cluster output by the algorithm that contains the x_i does not contain any peak of f of prominence τ or more beside x_l. This means that, for any $i < l$, the peak x_i is less than τ-prominent and therefore corresponds to some point of $\mathrm{D}^{\mathcal{R}}_1$ in the diagram $\mathrm{D}_0\mathcal{R}^f_\delta(L)$. It follows in particular that the prominence of x_i is less than $d_1 + 2c\delta$, which means that $C^{\mathcal{R}}(x_i, f(x_i) - d_1 - 2c\delta) = C^{\mathcal{R}}(x_{i+1}, f(x_i) - d_1 - 2c\delta)$. Now, we have $f(x_i) - d_1 - 2c\delta \geq f(x) - d_1 - 2c\delta \geq \alpha$, which implies that $C^{\mathcal{R}}(x_i, \alpha) = C^{\mathcal{R}}(x_{i+1}, \alpha)$. Since this is true for all $i < l$, we conclude that $C^{\mathcal{R}}(x_0, \alpha) = C^{\mathcal{R}}(x_1, \alpha) = \cdots = C^{\mathcal{R}}(x_l, \alpha) = C^{\mathcal{R}}(r^*_\tau(x), \alpha)$. Combined with the fact that $C^{\mathcal{R}}(x, \alpha) = C^{\mathcal{R}}(r(x), \alpha) = C^{\mathcal{R}}(x_0, \alpha)$, this proves the lemma. □

We are now ready to prove our first inclusion:

Lemma 4.14 *For all* $p \in \mathrm{D}_2$ *and all* $\alpha > \alpha_\tau(p) + d_1 + \frac{5}{2}c\delta$, $B^{\mathcal{R}}_\tau(p) \cap \mathbb{F}^\alpha \subseteq B_\tau(p) \cap L \cap \mathbb{F}^\alpha$.

Proof. For any $\alpha > f(v_p)$, $B^{\mathcal{R}}_\tau(p) \cap \mathbb{F}^\alpha$ is empty and so the inclusion holds trivially. Otherwise, consider a point $x \in B^{\mathcal{R}}_\tau(p) \cap \mathbb{F}^\alpha$. Since $f(x) \geq \alpha$, Lemma 4.13 guarantees that $C^{\mathcal{R}}(x, \alpha - d_1 - 2c\delta) = C^{\mathcal{R}}(r^*_\tau(x), \alpha - d_1 - 2c\delta)$. In other words, x and $r^*(x)$ belong to the same connected component of the subgraph $R_\delta(L \cap \mathbb{F}^{\alpha - d_1 - 2c\delta})$. Since by hypothesis $\alpha - d_1 - 2c\delta$ is greater than $\alpha_\tau(p) + c\delta/2$, Lemma 4.10 ensures that every point of $L \cap \mathbb{F}^{\alpha - d_1 - 2c\delta} \cap B_\tau(p)$, including $v_p = r^*(x)$ itself, is disconnected from every point of $L \cap \mathbb{F}^{\alpha - d_1 - 2c\delta} \setminus B_\tau(p)$ in the subgraph $R_\delta(L \cap \mathbb{F}^{\alpha - d_1 - 2c\delta})$. This implies that x belongs to $B_\tau(p)$. □

We now proceed with the inclusion in the other direction:

Lemma 4.15 *For all* $p \in \mathrm{D}_2$ *and all* $\alpha > \alpha_\tau(p) + d_1 + \frac{5}{2}c\delta$, $B_\tau(p) \cap L \cap \mathbb{F}^\alpha \subseteq B^{\mathcal{R}}_\tau(p) \cap \mathbb{F}^\alpha$.

Proof. Since by definition v_p is the argmax of f over $B_\tau(p) \cap L$, for all $\alpha > f(v_p)$ the set $B_\tau(p) \cap L \cap \mathbb{F}^\alpha$ is empty and so the inclusion holds trivially. Assume from now on that $\alpha_\tau(p) + d_1 + \frac{5}{2}c\delta < \alpha \leq f(v_p)$, and let $x \in B_\tau(p) \cap L \cap \mathbb{F}^\alpha$. Let $q \in \mathrm{D}_2$ be such that $v_q = r^*(x)$. Since $f(x) \geq \alpha$, Lemma 4.13 guarantees that $C^{\mathcal{R}}(x, \alpha - d_1 - 2c\delta) = C^{\mathcal{R}}(v_q, \alpha - d_1 - 2c\delta)$. Now, since $\alpha - d_1 - 2c\delta > \alpha_\tau(p) + c\delta/2$,

Lemma 4.10 ensures that every point of $L \cap \mathbb{F}^{\alpha-d_1-2c\delta} \cap B_\tau(p)$, including x itself, is disconnected from every point of $L \cap \mathbb{F}^{\alpha-d_1-2c\delta} \setminus B_\tau(p)$ in the subgraph $R_\delta(L \cap \mathbb{F}^{\alpha-d_1-2c\delta})$. This implies that v_q belongs to $B_\tau(p)$, and therefore that $v_q = v_p$. Hence, x belongs to $B_\tau^{\mathcal{R}}(p)$. □

The conclusion of Theorem 4.9 follows from the mutual inclusions stated in Lemmas 4.14 and 4.15. □

5 Practicality of the approach

In some practical scenarios the probability density distribution f according to which the input point cloud L has been sampled is known. However, in most cases it remains unknown, and in order to apply the clustering algorithm the values of f at the points of L must be estimated. Density estimation is an extensive research area and many methods to estimate the values of f from the data L can be used (see *e.g.* [14]). Nevertheless, one has to take care that the choice of the estimator does not break the validity of the assumptions made by the theoretical results of Section 4. To bridge the gap between the theoretical results of previous section and the practical cases where only L is known, we provide a simple condition on the estimated density to guarantee the quality of the output of our algorithm. We also show through standard arguments that this condition is satisfied for some simple estimators used in the experiments reported in Section 6.

Let $\mathbb{X}$ be a Riemannian manifold, possibly non-compact, possibly with boundary, with positive convexity radius $\varrho_c(\mathbb{X})$, and let $L \subset \mathbb{X}$ be a finite set sampled according to some c-Lipschitz probability density $f : \mathbb{X} \to \mathbb{R}$. Since Theorem 4.5 and Lemma 4.3 are the key properties from which all our guarantees on the output of the algorithm derive, all we need to do is to adapt them to the present setting where only an approximation of f is known. This will ensure that the theoretical guarantees provided in the previous section still hold in the new setting:

Theorem 5.1 *Let $C > 0$ and $\tilde{f}$ be an approximation of f such that*

$$\sup_{\ell \in L} |f(\ell) - \tilde{f}(\ell)| < C.$$

For any positive $\delta < \varrho_c(\mathbb{X})$ and any $\alpha > 0$, with probability at least $\big(1 - \mathcal{N}_{\delta/8}(\mathbb{F}^\alpha)\, e^{-|L|(\alpha-c\delta/4)\mathcal{V}_{\delta/8}(\mathbb{F}^\alpha)}\big)$ there is a multi-bijection γ between the 0-th persistence diagrams of f and of the upper-star filtration $\mathcal{R}_\delta^{\tilde{f}}(L)$ induced by $\tilde{f}$ on the Rips graph $R_\delta(L)$, such that:

(i) $\forall p \in \mathrm{D}_0 f \cap Q^{\mathrm{NE}}_{\alpha+C}$, $\|p - \gamma(p)\|_\infty \leq c\delta + C$.

(ii) $\forall q \in \mathrm{D}_0 \mathcal{R}_\delta^{\tilde{f}}(L) \cap Q^{\mathrm{NE}}_{\alpha+C}$, $\|\gamma^{-1}(q) - q\|_\infty \leq c\delta + C$.

(iii) $\forall p \in \mathrm{D}_0 f \cap Q^{\mathrm{SE}}_{\alpha+C}$, $|p_x - \gamma(p)_x| \leq c\delta + C$.

(iv) $\forall q \in \mathrm{D}_0 \mathcal{R}_\delta^{\tilde{f}}(L) \cap Q^{\mathrm{SE}}_{\alpha+C}$, $|\gamma^{-1}(q)_x - q_x| \leq c\delta + C$.

Proof. First, it follows from Lemma 4.3 that L forms a $\frac{\delta}{4}$-sample of the superlevel-set $\mathbb{F}^\alpha$ with probability $1 - \mathcal{N}_{\delta/8}(\mathbb{F}^\alpha)\, e^{-|L|(\alpha-c\delta/4)\mathcal{V}_{\delta/8}(\mathbb{F}^\alpha)}$. So, with

the same probability L statisfies the assumptions of Theorem 4.5, which means that $\mathrm{D}_0 f$ and $\mathrm{D}_0 \mathcal{R}_\delta^f(L)$ satisfy assertions (i) through (iv) of Theorem 4.5. Now, Theorem 3.7 of [5] ensures that the upper-star filtrations $\mathcal{R}_\delta^f(L)$ and $\mathcal{R}_\delta^{\tilde{f}}(L)$ are (strongly) C-interleaved. As a consequence, the bottleneck distance between their 0-th persistence diagrams is bounded by C, by the extended stability result of [3]. Combining these two results concludes the proof. □

As mentioned above, density estimation is an extensive research area, and identifying the families of density estimators that satisfy the conditions of Theorem 5.1 is beyond the scope of this paper. Nevertheless, in many cases constructing such an estimator is not difficult. We illustrate this in the Euclidean case with a very simple estimator that is used in the experimental section. We now assume that $\mathbb{X} = \mathbb{R}^m$ is endowed with the Euclidean metric and we denote by $\mathcal{V}_r = \mathcal{H}^m(\mathcal{B}(\ell, r))$ the m-dimensional Hausdorff measure of the ball of radius r. Let L be a finite set of data points sampled according to some probability density function $f : \mathbb{R}^m \to \mathbb{R}$. We assume that the coordinates of the points of L are given, so that their pairwise Euclidean distances can be computed exactly. The density f can thus be approximated using the following *ball estimator*:

$$\tilde{f}_r(\ell) = \frac{1}{\mathcal{V}_r} \frac{|L \cap \mathcal{B}(\ell, r)|}{|L|}. \tag{8}$$

Lemma 5.2 *If f is c-Lipschitz, then for any value of parameter r and any $\zeta \geq 0$,*

$$\sup_{\ell \in L} |f(\ell) - \tilde{f}_r(\ell)| \leq cr + \zeta$$

with probability at least $1 - 2|L|e^{-2|L|(\zeta \mathcal{V}_r)^2)}$.

Proof. Let μ be the probability measure such that for any ball $\mathcal{B}(\ell, r)$,

$$\mu(\mathcal{B}(\ell, r)) = \int_{\mathcal{B}(\ell, r)} f d\mathcal{H}^m. \tag{9}$$

By the Intermediate Value Theorem, there exists a point $y \in \mathcal{B}(\ell, r)$ such that $f(y)$ equals the average value of f inside the ball, that is: $f(y) = \frac{\mu(\mathcal{B}(\ell, r))}{\mathcal{H}^m(\mathcal{B}(\ell, r))}$. Since f is c-Lipschitz, we have $|f(y) - f(\ell)| \leq cr$, which implies:

$$\left| f(\ell) - \frac{\mu(\mathcal{B}(\ell, r))}{\mathcal{V}_r} \right| \leq cr. \tag{10}$$

By the Bounded Difference inequality, we know that

$$\left| \frac{|L \cap \mathcal{B}(\ell, r)|}{|L|} - \mu(\mathcal{B}(\ell, r)) \right| \leq \eta \tag{11}$$

with probability at least $e^{-2|L|\eta^2}$. Letting $\eta = \zeta \mathcal{V}_r$ in the above expression and combining it with Eq. (10), we obtain a bound on the difference between the ball estimator (8) and the true density value at point l. The conclusion of the lemma follows then from the application of the union bound. □

Notice that the ball estimator (8) strongly relies on the property that the volume of a ball of radius r in $\mathbb{R}^m$ does not depend on its center. This is not the case in general Riemannian manifolds. To overcome this issue it is possible to consider kernel based estimators of the following form:

$$\tilde{g}(\ell) = \frac{\sum_{p \in L} K(d_{\mathbb{X}}(p, \ell))}{|L|}, \tag{12}$$

where $K : \mathbb{R} \to \mathbb{R}$ is a non negative function such that $\int_{-\infty}^{\infty} K(u)du = 1$ and $K(u) = K(-u)$. Under some conditions and if L is sampled according to f, $\tilde{g}$ can be seen as an estimator of the convolution of f with $K \circ d_{\mathbb{X}}$. We refer the reader to [15, 33] for further details on kernel density estimation.

Another issue that may occur in practice is that the geodesic distances between the data points are not given. It is then necessary to approximate them in order to feed in our clustering algorithm. The data structure used for the approximation depends on the particular scenario considered: for instance, when the data points lie in Euclidean space $\mathbb{R}^d$ with known coordinates, and $\mathbb{X}$ is an unknown m-submanifold of $\mathbb{R}^d$, geodesic distances in $\mathbb{X}$ can be approximated through graph distances within some well-chosen neighborhood graph; in wireless sensor networks scenarios, the geographic locations of the nodes are usually not available but graph distances within the communication network can be used instead. We refer the reader to [5, §3.3] for further discussion on geodesic distances approximation: the bottom line is that in many practical scenarios geodesic distances can be approximated within some small additive error. It is then possible to combine Theorem 4.5 with Theorem 3.9 of [5] in the same way as we did in the proof of Theorem 5.1, to prove an equivalent of Theorem 5.1 that guarantees that the theoretical results of Section 4 still hold. We refer the reader to [5, §3.3] for the formal statement and proof of this result, which is rather technical and does not present any conceptual difficulty.

6 Experimental Results

We will now present the results of our clustering scheme on various datasets. The latter are of three different types: synthetic (Section 6.1), color components of 2-d images (Section 6.2), and protein conformations (Section 6.3).

The synthetic dataset is sampled from 4 interlocking rings in $\mathbb{R}^3$. It is highly non-linear and non-separable, implying that standard techniques such as k-means will fail. On the other hand, it is relatively easy to interpret by visual inspection, which makes it suitable for highlighting the main ideas underlying our approach, especially the interpretation of the output of the algorithm. We use this toy example to discuss the role of the persistence diagram in choosing suitable parameter values; we also show how this choice influences the final result.

The second type of data comes from 2-d color image segmentation applications. Each pixel in a given image is mapped to a point in some 3-dimensional color space (typically Luv), to which two additional coordinates corresponding to the spatial location of the pixel may be added. These experiments illustrate how different estimators emphasize different features of the image at different scales.

The third type of data comes from computational biology. We consider a set of protein conformations computed from simulations of the dynamics of the alanine-dipeptide molecule. Each point in the dataset represents a conformation of the molecule. It lives in 22-dimensional space equipped with a non-Euclidean norm. The resulting clusters are called *metastable states*, meaning that transitions between these states are largely Markovian. This example is perhaps the closest to unsupervised learning, as direct data inspection in conformation space is not possible. Although it is an already well-studied data, we show that, in addition to finding 6 clusters as previous work has done, the persistence diagram produced by our algorithm suggests that 7 clusters would be just as reasonable.

Our density estimators. In all our experiments, our first task was to estimate the density. To do so we used two estimators: a truncated Gaussian estimator, and the *distance to a measure* introduced in [4]. For completeness, we recall the definitions of both estimators. The truncated Gaussian is given by

$$f(x) = \frac{\sum_{\ell \in L} K(d(x,p))}{|L|}, \tag{13}$$

where $K(\cdot)$ is define as follows:

$$K(d(x,p)) = \begin{cases} e^{-d^2(x,p)/2h} & d(p,q) \leq h \\ 0 & \text{else} \end{cases} \tag{14}$$

Here, parameter h is called the *bandwidth* of the estimator.

Our second estimator is the so-called distance to a measure, which computes the root-mean-squared distance to the k nearest neighbors of the considered query point:

$$f(x) = \sqrt{\frac{1}{k}\sum_{i=1}^{k} d^2(x,p_i)}. \tag{15}$$

Here, p_i denotes the i-th nearest neighbor of x among the point set L. This estimator can be seen as an improved version of the standard k-NN estimator. It enjoys interesting stability properties – see [4] for more details. In contrast to the Gaussian estimator, it has a fixed complexity since it only considers the k nearest neighbors of the query point. Note that the resulting function f is a distance rather than a density, so we take $(\text{-}f)$ as the input of the clustering algorithm.

Our choices of values for parameters δ and τ. As mentioned in the introduction, parameters δ and τ are of very different natures. To set δ, we computed a single-linkage clustering and chose a relevant scale from the resulting dendrogram (omitted in our experimental results). Then, we ran our clustering algorithm with the chosen value of δ and with $\tau = +\infty$, to produce an approximation of the persistence diagram of the density, from which we inferred a relevant value for parameter τ. The diagram and its interpretation are shown in our experimental results for demonstration purposes. Finally, a second run of our algorithm was performed, with the chosen parameter values, to produce the final clustering.

6.1 Synthetic dataset

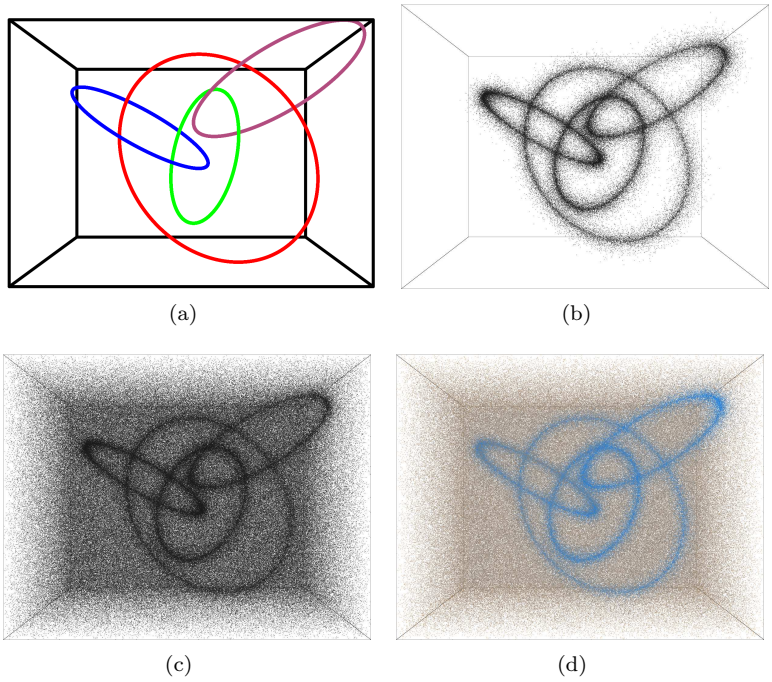
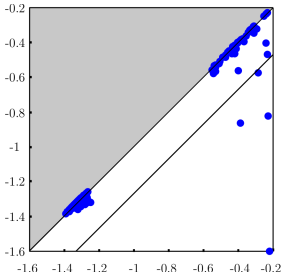


Figure 9: Our synthetic dataset. (a.) The four interlocked rings. (b.) The four rings convolved with a gaussian distribution and sampled with 100,000 points. (c.) The input point cloud (600,000 points), obtained by adding background noise to the sample of Figure (b). (d.) Density estimation on the input point cloud using the distance to a measure. Exceptionally, density values range from orange to blue, and not from blue to orange. This choice was made for visualization purposes.

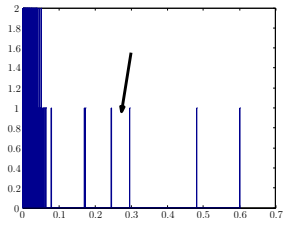
The dataset is shown in Figure 9. The clusters are highly non-linear and non-separable. To generate the point cloud, one of the rings was first chosen at random with equal probabilities, and then a sample point was chosen from the uniform distribution over this ring convolved with a Gaussian distribution. Using this procedure, 100,000 points were generated, shown in Figure 9(b). Note that differences in lengths of the rings result in differences in sampling densities. For instance, the larger outer rings are sampled more sparsely that the smaller inner rings. To ensure that the rings were not completely disjoint, 500,000 uniformly distributed points were added, giving the input point cloud shown in Figure 9(c).

As a density estimator we used the distance to a measure with $k = 500$ nearest neighbors. Different numbers of points and neighbors were used as well, yielding similar results. The resulting approximate density function is shown in Figure 9(d). The Rips graph in the standard Euclidean metric was computed using the ANN library [36]. We tested the algorithm with up to 5 million input points, in which case it concluded in around 1 hour. The example shown in the paper has 600k points and took approximately 5 minutes to process. It is

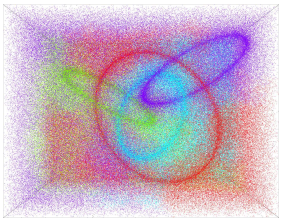
important to note that the experiments were run with various samplings and numbers of points, to ensure that the results were not obtained by chance.



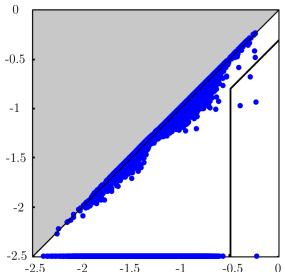
(a) Persistence diagram using Rips parameter $\delta = 1$



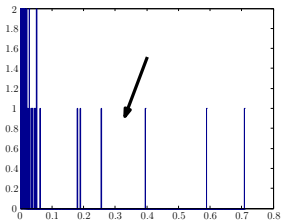
(b) Histogram of prominences ignoring infinitely persistent components.



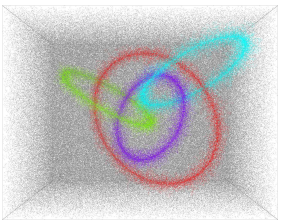
(c) Resulting clustering.



(d) Persistence diagram using Rips parameter $\delta = 0.3$.



(e) Histogram of prominences ignoring infinitely persistent components and components born after 0.5.



(f) Resulting clustering with independent components born after 0.5 filtered out.

Figure 10: Influence of parameter δ on the result: $\delta = 1$ (top row) versus $\delta = 0.3$ (bottom row).

The influence of the choice of parameter δ on the output of the algorithm is illustrated in Figure 10. With $\delta = 1$, the persistence diagram (Figure 10(a)) shows one infinitely-persistent cluster, which means that with this Rips parameter all the points are connected in the Rips graph. Upon closer inspection, it can be seen that there are potentially 1 to 7 clusters. This is more clearly visible in the histogram of the prominence values shown in Figure 10(b). The histograms were used to choose parameter τ, as they clearly stress the relationship between the merging parameter and the number of clusters[10]. The arrow in the figure shows the gap corresponding to values of τ that make the algorithm recover the four rings. The gap, though narrow, is still significant, especially considering the amount of noise present. The resulting clustering (Figure 10(c)) illustrates that the four most prominent clusters do indeed correspond to the four rings. The persistence diagram shows that the next three most prominent clusters are also significantly more prominent than the others. If the merging parameter is set to recover these clusters as well, we see that each of the outer rings is split into two pieces. This is a consequence of the sparser and uneven sampling on the outer rings. While all the points are clustered, the outermost points are assigned

[10]In the histograms we do not show the clusters with infinite lifetimes.

somewhat randomly, since they are generated from a uniform distribution and so are sensitive to the instantiation of the samples.

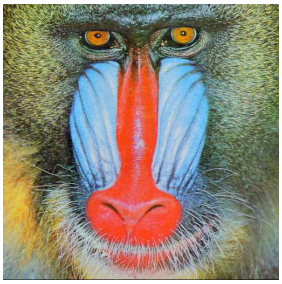
As we have a rather dense sampling, we can obtain better results by choosing a smaller Rips parameter. With $\mathcal{R}_\delta = 0.3$, we obtain many local maxima as well as many independent components. The persistence diagram (Figure 10(d)) has over 200,000 points! However, almost all of them correspond to hardly persistent or late-appearing connected components. As before, we choose the merging parameter from the persistence histogram (Figure 10(e)). We also choose to ignore infinitely persistent components that born after -0.5, that is, that lie to the left of the vertical line $x = -0.5$ in Figure 10(d). Note that there is only one infinitely persistent component remaining, which means that the four rings are still connected with each other in the Rips graph. The resulting clustering is shown in Figure 10(f). The original four rings are recovered as before but most of the background noise is removed since it generated late-appearing clusters. Notice also that the gap in the histogram has increased. The large number of points along the diagonal in the diagram indicates how the smaller Rips parameter captures smaller variations in density as more local maxima are captured. As these are not significant, they are quickly merged into more relevant clusters.

6.2 Image Segmentation

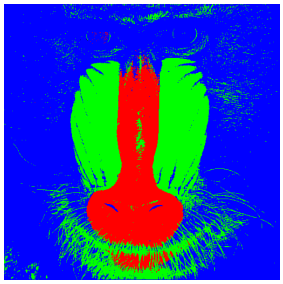
We ran two sets of experiments: the first one was performed in Luv color space, while during the second one we incorporated spatial information into the point cloud data. The reason why Luv components were used instead of, say, RGB components, is that in Luv space the Euclidean distance is known to capture the subjective notion of perceptual difference reasonably well [11].

For each of the test images, we show the original and corresponding point cloud in Luv space as well as the resulting segmentation. Along with the histogram, we also show the persistence diagram in the form of a *barcode*. In this representation, each point $p = (p_x, p_y)$ becomes an interval $[p_y, p_x]$ (recall that p_x is larger than p_y since time flows from $+\infty$ to $-\infty$). When $p_x = +\infty$, the interval is infinite and therefore terminated by an arrow. This visualization contains exactly the same information as the persistence diagram, but it can sometimes be clearer since the more prominent intervals are longer and so are more easily seen.

In the first set of experiments we cluster the point clouds in the 3-d Luv space directly, ignoring all spatial information in the image domain. Initially, areas of uniform color proved challenging because a large number of points were in close proximity in Luv space, resulting in a large number of edges in the Rips graph. To speed-up the computation, we used the following downsampling procedure: We begin by making all points unmarked. Considering each point p in order: if p is unmarked, then we detect all the data points within Euclidean distance δ of p, and we create Rips edges as usual. Then, we detect all the data points within a smaller distance δ/m of p and mark them: these points are to be removed from the dataset for the clustering phase. Their cluster center will be the same as p's. Typically, we chose m between 10 and 20 in our experiments. This downsampling procedure can be shown to induce a small additive error (in the order of δ/m) on the persistence diagram approximation, therefore it is provably safe. We now go through the examples in detail:

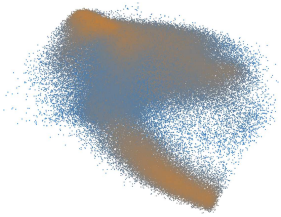


(a) The original 512×512 image.

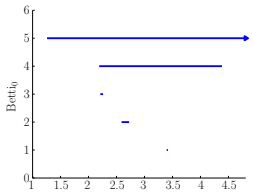


(b) Segmentation into 3 clusters using the distance to a measure.

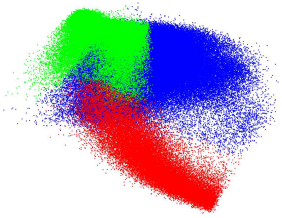
(c) Segmentation into 5 clusters using the Gaussian estimator.



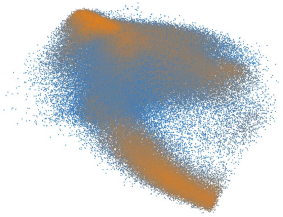
(d) Point cloud in Luv space and estimated density using the distance to a measure with $k = 1000$.



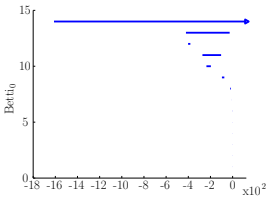
(e) Persistence barcode obtained from the point cloud of (d).



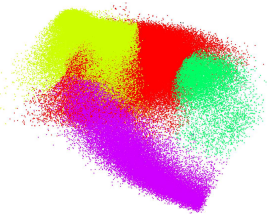
(f) Segmentation of the point cloud of (d) in 3 clusters.



(g) Point cloud and estimated density using the Gaussian estimator with $h = 25$.



(h) Persistence barcode obtained from the point cloud of (g)



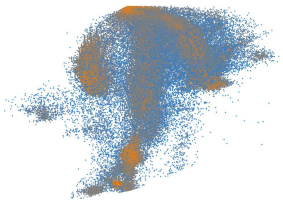
(i) Segmentation of the point cloud of (g) into 5 clusters.

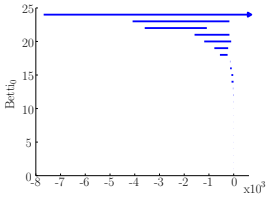
Figure 11: Results of our approach on the Mandrill dataset.

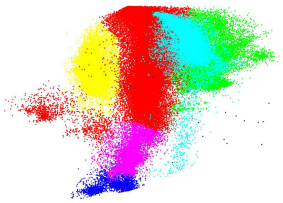
Mandrill dataset. The results are shown in Figure 11. We tried both the distance to a measure estimator with $k = 1000$ and a Gaussian estimator with a bandwidth of $h = 25$. These parameter values were chosen by examining the range of the dataset. In the case of the distance to a measure (Figure 6.2(d-f)), we show the result with 3 clusters. They can be clearly seen in the point cloud 11(f). When we map this back to the image we see that the clusters correspond to the nose, cheeks and fur. Using the Gaussian estimator (Figure 6.2(g-i)), we show 5 clusters where further features can be identified including the eyes and the yellow part of the fur. Note that the 5th cluster corresponds to very dark colors and lies therefore on the underside of the point cloud in Figure 11(i), which makes it invisible in the picture. It is also barely visible in the image since there are few very dark patches. It is important to note that in the

(a) The original image.

(b) Segmentation into 6 clusters.







(c) Point cloud in Luv space and estimated density using the Gaussian estimator with $h = 25$.

(d) Persistence barcode obtained from the point cloud of (c).

(e) The point cloud in Luv space segmented into 6 clusters.

Figure 12: Results of our approach on the Street dataset.

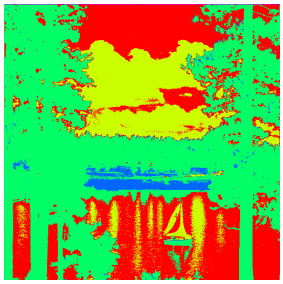
Gaussian case, black is the third most prominent cluster, whereas in the case of the distance to a measure it is the fourth one.

Our experiments on subsequent images followed the same procedure. However, to save space we only report the results obtained using the Gaussian estimator with bandwith parameter $h = 25$.

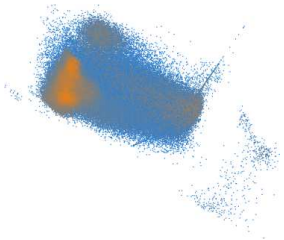
Street dataset. The results are shown in Figure 12. In this picture of a San Francisco street, we see 4 different colored buildings, the road, the sky, and a few cars. Figure 12(c) shows the result of the Gaussian estimator, while Figures 12(e) and 12(b) show the six clusters output by the algorithm, both in Luv color space and mapped back onto the image. As can be seen, we recover the sky, the 4 houses and the street. Note that in the barcode there is a small gap between 6 and 7 clusters, the 7th cluster being texture resulting from shadows on some of the houses. By contrast, the gap between 7 and 8 clusters is clear. Alternatively, if we choose 5 clusters, then the street merges with the left-most house.

Landscape dataset. The results are shown in Figure 13. This data set is interesting because in color space, an obvious cluster appears which is invisible in the image. This is the purple cluster in Figure 13(e). Although it is seeminlgy absent in the image, it is in fact present in the outline of the trees, which makes it hardly visible. In the barcode, we can clearly see this cluster as the second infinitely persistent bar, since it is sufficiently far from the rest of the data points. There are 4 other clusters in the image. The trees form one cluster, the sky and water another cluster and the reflections, clouds and sails a third

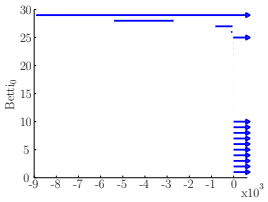
(a) The original image.



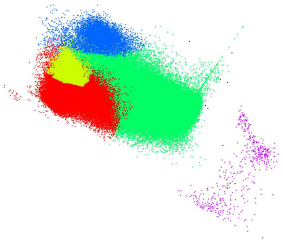
(b) Segmentation into 5 clusters.



(c) Point cloud in Luv space and estimated density using the Gaussian estimator with $h = 25$.



(d) Persistence barcode obtained from the point cloud of (c).



(e) The point cloud in Luv space segmented into 5 clusters.

Figure 13: Results of our approach on the Landscape dataset.

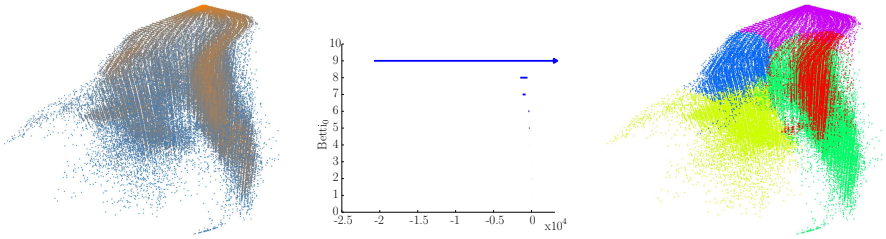
cluster. Finally, we have the grassy area as our fourth cluster. These 4 clusters are clearly seen in the barcode, which may be due to the fact that the image is a painting, not a photograph.

Koala dataset. The results are shown in Figure 14. Our segmentation separates the koala, the tree, the background and the other plants. The 5th cluster (shown in blue) is again the shadows which are significantly darker than the other parts of the image. The interesting thing to note about this example is that there is a clear quantization effect in the point cloud, due to the fact that this image was converted to JPEG format. Note nevertheless that this effect did not adversely affect our algorithm.

Taking spatial information into account. Clustering in Luv space is oblivious to proximity relations between pixels in the image. Thus, it allows pixels that are far apart in the image to end up in a same cluster. Depending on the context, this property can be viewed either as a feature or as a drawback. Removing it requires to take spatial information into account during the clustering phase. The most naive way of doing so consists in appending the two pixel coordinates to the three color channels, thus yielding point cloud data in a 5-dimensional space. The big disadvantage is that the contributions of color and spatial coordinates must be balanced appropriately in the computation of distances, because the scales of the color channels and spatial coordinates are unrelated. This is a whole issue in its own right.

(a) The original image.

(b) Segmentation into 5 clusters.



(c) Point cloud in Luv space and estimated density using the Gaussian estimator with $h = 25$.

(d) Persistence barcode obtained from the point cloud of (c).

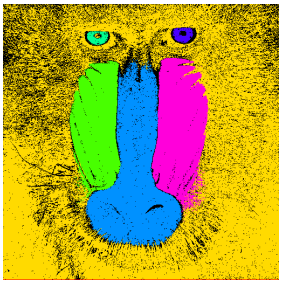
(e) The point cloud in Luv space segmented into 5 clusters.

Figure 14: Results of our approach on the Koala dataset.

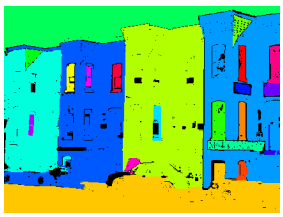
To avoid this pitfall, we still consider our point clouds in 3-dimensional Luv space and compute our density estimates as above. However, for two data points to be connected in the Rips graph we now require that they be close both in Luv space and in the image domain. Basically, this boils down to pruning the previously computed Rips graph using a spatial filter. In practice, we proceed in the opposite order, that is: we first connect points that are close in the image domain, then we prune out the edges whose vertices are far apart in Luv space. The reason for doing so is obvious: thanks to the grid structure of the image, the number of neighbors of a pixel is constant and therefore the algorithm runs much faster (less pruning occurs as well). In practice, we typically we chose patches of size 5×5 around each pixel, to ensure a mostly connected graph[11].

The results obtained with this approach are shown in Figure 15. We use the same images as before and notice some interesting results. In the case of the mandrill (Figures 15(a) and 15(d)), we are able to discriminate the left cheek from the right cheek and the left eye from the right eye. The dark pixels correspond to very small clusters due to the texture of the fur: these were discarded in a post-processing step. In the persistence histogram we see a clear gap, within which we chose the value of the merging parameter τ. In the case of the street, there is significantly more noise and less of a gap, as can be seen in from the histogram (Figure 15(e)). In the image (Figure 15(b)), we

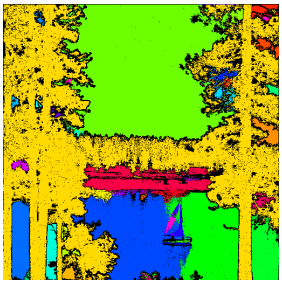
[11] In practice, natural images have textures that create very small clusters independent from the rest of the data. These are simply ignored in practice. More precisely, once the clustering is done, we discard all clusters of cardinality less than a threshold, taken to be 100 in our experiments. These discarded clusters are shown in black in our results.



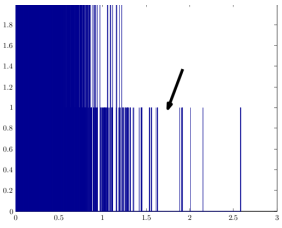
(a) Mandrill segmented with color and spatial information.



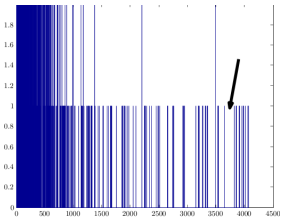
(b) Street image segmented with color and spatial information.



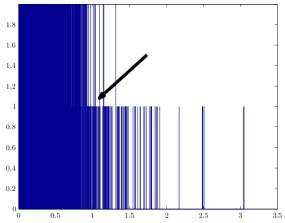
(c) Landscape image segmented with color and spatial information.



(d) Persistence histogram for the mandrill. The arrow indicates the merging parameter.



(e) Persistence histogram for the street image.



(f) Persistence histogram for the landscape image.

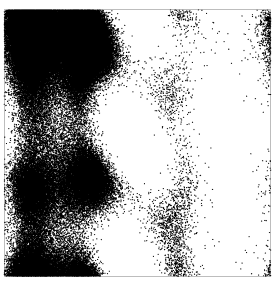
Figure 15: Results of our approach, taking both color and spatial information into account.

recover not only the buildings, sky and road, but also the windows as individual clusters. Finally, on the landscape image (Figure 15(b)), trees appear as the major cluster. The sky is merged with the clouds and the water is split into two parts by the sailboat. The outline of the trees is spatially a small cluster. The histogram shows many small clusters that were filtered out in the post-processing step.

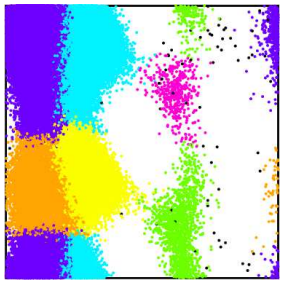
6.3 Alanine-dipeptide dynamics

For the study of a biological system, it is often desirable to obtain accurate simulations of its dynamics. In the case of a protein, this requires the simulations to be done at the level of atoms, a computer-intensive task that limits the duration of the simulations to picoseconds even for small molecules. One proposed approach around this is to use *metastable* states of the protein. These are states between which transitions are infrequent and independent. This allows for accurate simulations using Markovian models [7, 8, 9], which are computationally much easier to simulate. A key issue is to discover the metastable states. The *metastability* of a clustering is defined as the sum of the self-transition probabilities for a given lag time [21]. In other words, a good clustering should produce states that are stable with high probability under random transitions of the protein within the specified lag time.

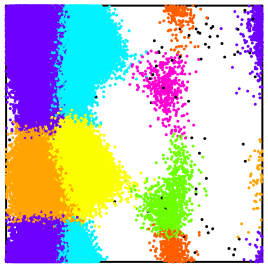
As an illustration, we now look at the conformational dynamics of the terminally blocked alanine-dipeptide molecule. The choice of this paricular protein



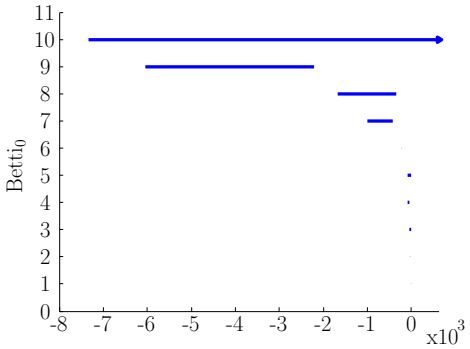
(a) Input point cloud after projection down to the (ϕ, ψ) domain.



(b) Segmentation with 6 clusters.



(c) Segmentation with 7 clusters.



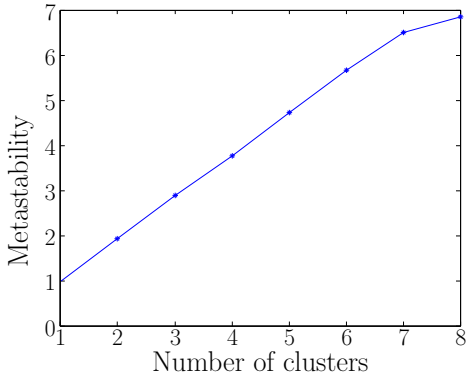
(d) Barcode computed using the Gaussian estimator.

Rank	(Unnormalized) Prominence
1	∞
2	5677
3	3828
4	1335
5	850
6	316
7	258
8	72
9	30
10	22

(e) Intervals sorted by decreasing prominence.

Figure 16: Results of our clustering method, visualized in the (ϕ, ψ) domain.

is motivated by the fact that this is a small molecule whose dynamics is relatively well-understood. It is known that there are only two relevant degrees of freedom, parametrized by two angles ϕ and ψ. This makes it possible to visualize the conformation data by projecting the points from the 22-d conformation space down to the 2-d (ϕ, ψ) domain, as shown in Figure 16(a). It is in this 2-d domain that the previous *gold standard* for metastable clustering was done, resulting in 6 clusters.



Number of clusters	Normalized Metastability
1	0.9998
2	0.9594
3	0.9603
4	0.9404
5	0.9467
6	0.9427
7	0.9295
8	0.8549

Figure 17: Left: metastability versus number of clusters. Right: normalized metastability versus number of clusters. Normalization is done with respect to the number of clusters.

Our input consists of a set of conformations obtained from simulations at 400 Kelvins with 192 trajectories of 1000 configurations each. The spacing between samples in the time domain is 0.1 ps. In the simulations, the peptide was modeled by the AMBER 96 forcefield [29] and solvated in TIP3P water [22]. The simulations themselves where done using the parallel tempering method of [8]. We refer the reader to these texts for a more in-depth discussion. For our experiments, we took the 192,000 conformations along the trajectories and treated them as independent samples in 22-d space. The distance metric used was root-mean-squared deviation (RMSD) after the best rigid motion matching. This was done using the Theobald method [32]. The point cloud in 22 dimensions, together with the pairwise RMSD distances between the samples, were the only two inputs to the algorithm. Figure 16(a) shows the input point cloud after projection in the (ϕ, ψ) domain.

We tried our two density estimators on this data set, with similar outcomes. Figure 16(d) shows the persistence barcode computed using the Gaussian estimator, which suggests that there are 4 main clusters. In order to distinguish between the other intervals in the barcode, we have sorted them according to prominence in the table of Figure 16(e). In this representation it also appears that there could be anywhere from 5 to 7 clusters. Figure 17 shows the metastabilities achieved by our clusterings with different numbers of clusters. This data corroborates the observation made from our barcode that it sounds reasonable to consider 7 metastable clusters instead of 6. It also suggests that the quality of our clustering is slightly better than the one achieved by previous methods: for instance, the normalized metastability computed in [7] is about 0.94 for 6 clusters, against 0.9427 with our method. Of course, these observations need to be validated by further data inspection, which is beyond the scope of the paper.

In terms of running time, computing RMSD distances between conformations was by far the longest phase (about a day). For each data point, only the closest 15000 conformations where recorded. The clustering itself only took 10 minutes on a personal computer, most of which were spent on disk accesses. The main memory usage remained constant throughout the clustering phase, and as a result several runs could be done in parallel on a multi-core processor.

7 Conclusion

We have presented a novel clustering scheme that combines a mode-seeking phase with a cluster merging phase. While mode detection is done by a standard graph-based technique, the true novelty of our approach resides in its use of topological persistence to guide the merges between clusters. The outcome does not reduce to a mere set of clusters, but it also includes visual feedback in the form of a persistence barcode or diagram, which can be used to tune the parameters.

Taking advantage of recent advances in persistence theory, we have given a theoretically sound notion of what a *good* clustering is, and we have proved that our algorithm produces such clusterings. Furthermore, we have given probabilistic statements relating the probability of success and the number of samples drawn. We have also addressed several practical issues including the effect of density estimation and of uncertainty in distance measurements on the quality of the output. These theoretical guarantees hold in a very large setting, thereby making the approach quite general. They have been validated on practical data, both synthetic and real, coming from various fields of application.

This work has raised numerous questions that deserve further treatment. First, we only used two density estimators for illustration purposes. Density estimation is an area of research in its own right, and many other estimators could have been considered as well, leading to different results. An important goal for us would be to be able to reliably estimate densities over Riemannian manifolds. There has recently been some work in this direction [27, 28, 31], but what can be said when the manifold underlying the data remains unknown?

Another important aspect in our work is the metric. Here we mainly used Euclidean or geodesic distances in Riemannian manifolds, but other metrics could be considered as well. For instance, diffusion distances within the neighborhood graph are likely to affect the behaviour of our algorithm, making it less sensitive to the local connectivity of the graph. In some sense, this would be like using or algorithm in stead of *k-means* in spectral clustering.

In this paper we concentrated on 0-dimensional homology, but as mentioned in Section 4.1 our theoretical results can be extended to higher dimensions, where all the arguments follow through. With higher-dimensional persistence information at hand, we can detect more subtle phenomena such as the fact that one of the clusters in Figure 3 has an annulus shape. Unfortunately, the complexity of the algorithm increases significantly in higher dimensions, potentially taking $O(m^3)$ time and $O(m^2)$ space, where m is the size of the *Rips complex* (the generalization of the Rips graph in higher dimensions). Furthermore, m is known to grow exponentially with the dimension of the complex, thus making the approach tractable only when the intrinsic dimensionality of the data remains small.

Our theoretical guarantees on the spatial stability of the clusters opens the door to a more statistical approach to clustering: since we know some parts of the clusters computed by the algorithm are stable under small perturbations of the density, we can conduct multiple runs of the algorithm with random pertubations of the input data, and then find correspondences between clusters generated at different runs. Thus, a *soft* clustering of the dataset can be built, where the probability that a given data point p is assigned to a given cluster C

corresponds to the fraction of the runs that actually connected p to C. It would be interesting to study the stability properties of such a clustering.

References

[1] D. Burago, Y. Burago, and S. Ivanov. *A Course in Metric Geometry*, volume 33 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2001.

[2] F. Chazal and D. Cohen-Steiner. *Geometric Inference*. to appear as a book chapter, Springer, 2007.

[3] F. Chazal, D. Cohen-Steiner, L. J. Guibas, M. Glisse, and S. Y. Oudot. Proximity of persistence modules and their diagrams. In *Proc. 25th ACM Sympos. Comput. Geom.*, 2009.

[4] F. Chazal, D. Cohen-Steiner, and Q. Merigot. Geometric inference for measures based on distance functions. Technical report, INRIA, May 2009.

[5] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Analysis of scalar fields over point cloud data. Research Report 6576, INRIA, July 2008.

[6] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Analysis of scalar fields over point cloud data. In *Proc. 20th ACM-SIAM Sympos. Discrete Algorithms*, pages 1021–1030, 2009.

[7] John D. Chodera, Nina Singhal, Vijay S. Pande, Ken A. Dill, and William C. Swope. Automatic discovery of metastable states for the construction of markov models of macromolecular conformational dynamics. *The Journal of Chemical Physics*, 126(15):155101, 2007.

[8] John D. Chodera, William C. Swope, Jed W. Pitera, and Ken A. Dill. Long-time protein folding dynamics from short-time molecular dynamics simulations. *Multiscale Modeling & Simulation*, 5(4):1214–1226, 2006.

[9] John D. Chodera, William C. Swope, Jed W. Pitera, Chaok Seok, and Ken A. Dill. Use of the weighted histogram analysis method for the analysis of simulated and parallel tempering simulations. *Journal of Chemical Theory and Computation*, 3(1):26–41, 2007.

[10] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. In *Proc. 21st ACM Sympos. Comput. Geom.*, pages 263–271, 2005.

[11] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2nd edition, 2001.

[13] M. d'Amico, P. Frosini, and C. Landi. Using matching distance in size theory: a survey. *International Journal of Imaging Systems and Technology*, 16(5):154–161, 2006.

[14] L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer, 2001.

[15] Luc Devroye and Laszlo Gyorfi. *Nonparametric Density Estimation: The L1 View*. Wiley Series in Probability and Statistics, 1985.

[16] H. Edelsbrunner and J. Harer. Persistent homology — a survey. *In Twenty Years After, AMS*, 2007.

[17] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse complexes for piecewise linear 2-manifolds. In *Proc. 17th Annu. Sympos. Comput. Geom.*, pages 70–79, 2001.

[18] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.

[19] S. Gallot, D. Hulin, and J. Lafontaine. *Riemannian Geometry*. Universitext. Springer, 3 edition, 2004.

[20] A. Hatcher. *Algebraic Topology*. Cambridge Univ. Press, 2001.

[21] Wilhelm Huisinga and Bernd Schmidt. Advances in algorithms for macromolecular simulation, chapter metastability and dominant eigenvalues of transfer operators. *Lecture Notes in Computational Science and Engineering. Springer*, 2005.

[22] William L. Jorgensen, Jayaraman Chandrasekhar, Jeffry D. Madura, Roger W. Impey, and Michael L. Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, 79(2):926–935, 1983.

[23] A. Kolmogorov and V. Tikhomirov. ϵ-entropy and ϵ-capacity of sets of functions. *Translations of the AMS*, 17:277–364, 1961.

[24] W. L. Koontz, P. M. Narendra, and K. Fukunaga. A graph-theoretic approach to nonparametric cluster analysis. *IEEE Trans. on Computers*, 24:936–944, September 1976.

[25] John W. Milnor. *Morse Theory*. Princeton University Press, Princeton, NJ, 1963.

[26] F. Morgan. *Geometric Measure Theory: a Beginner's Guide*. Academic Press, 1988.

[27] Bruno Pelletier. Kernel density estimation on riemannian manifolds. *Statistics & Probability Letters*, 73(3):297 – 304, 2005.

[28] Bruno Pelletier. Non-parametric regression estimation on closed riemannian manifolds. *Journal of Nonparametric Statistics*, 18(1):57–67, 2006.

[29] K. Roterman, M. H. Lambert, K. D. Gibson, and H. A. Scheraga. A comparison of the charmm, amber and ecepp potentials for peptides. ii. ϕ-ψ maps for n-acetyl ala- nine n-methyl amide: Comparisons, contrasts and simple experimental tests. *Journal of Biomolecular Structural Dynamics*, 78:421–453, 1989.

[30] Y. A. Sheikh, E. Khan, and T. Kanade. Mode-seeking by medoidshifts. In *Proc. 11th IEEE Internat. Conf. on Computer Vision (ICCV 2007)*, October 2007.

[31] Raghav Subbarao and Peter Meer. Nonlinear mean shift for clustering over analytic manifolds. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1168–1175, Washington, DC, USA, 2006. IEEE Computer Society.

[32] D. L. Theobald. Rapid calculation of rmsds using a quaternion-based characteristic polynomial. *Acta Crystallographica Section A: Foundations of Crystallography*, 61(4):478–480, 2005.

[33] Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 2008.

[34] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *Proc. European Conf. on Computer Vision (ECCV)*, 2008.

[35] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete Comput. Geom.*, 33(2):249–274, 2005.

[36] `http://www.cs.umd.edu/~mount/ANN/`.

A Appendix — Proof of Lemma 4.6

Let $\mathcal{X} = \{X^\beta\}_{\beta\in\mathbb{R}}$ and $\mathcal{Y} = \{Y^\beta\}_{\beta\in\mathbb{R}}$ be two tame persistence modules that are (strongly) ε-interleaved above some given time α. Let $\{x_\beta^{\beta'} : X^{\beta'} \to X^\beta\}_{\beta'\geq\beta}$ be the family of homomorphisms associated with $\mathcal{X}$, and $\{y_\beta^{\beta'} : Y^{\beta'} \to Y^\beta\}_{\beta'\geq\beta}$ the family of homomorphisms associated with $\mathcal{Y}$. We define a new persistence module $\tilde{\mathcal{X}}$ from $\mathcal{X}$ as follows:

$$\left\{\begin{array}{l} \forall\beta \geq \alpha,\ \tilde{X}^\beta = X^\beta \\ \forall\beta < \alpha,\ \tilde{X}^\beta = 0 \end{array}\right. \qquad \left\{\begin{array}{l} \forall\beta \geq \alpha,\ \forall\beta' \geq \beta,\ \tilde{x}_\beta^{\beta'} = x_\beta^{\beta'} \\ \forall\beta < \alpha,\ \forall\beta' \geq \beta,\ \tilde{x}_\beta^{\beta'} = 0 \end{array}\right. \tag{16}$$

Clearly, $\tilde{x}_\beta^{\beta'} \circ \tilde{x}_{\beta'}^{\beta''} = x_\beta^{\beta'} \circ x_{\beta'}^{\beta''} = x_\beta^{\beta''} = \tilde{x}_\beta^{\beta''}$ when $\beta \geq \alpha$, whereas $\tilde{x}_\beta^{\beta'} \circ \tilde{x}_{\beta'}^{\beta''} = 0 = \tilde{x}_\beta^{\beta''}$ when $\beta < \alpha$. Thus, $\tilde{\mathcal{X}}$ is indeed a persistence module. Its relationship with $\mathcal{X}$ is encoded in the following commutative diagram, where i_β is the identity over $X^\beta = \tilde{X}^\beta$ when $\beta \geq \alpha$ and the constant zero map otherwise:

$$\begin{array}{ccc} X^{\beta'} & \stackrel{x_\beta^{\beta'}}{\to} & X^\beta \\ {\scriptstyle i_{\beta'}}\downarrow & & {\scriptstyle i_\beta}\downarrow \\ \tilde{X}^{\beta'} & \stackrel{\tilde{x}_\beta^{\beta'}}{\to} & \tilde{X}^\beta \end{array} \tag{17}$$

Since $i_{\beta'}$ and i_β are isomorphisms whenever $\beta' \geq \beta \geq \alpha$, the commutativity of (17) implies that

$$\forall\beta' \geq \beta \geq \alpha,\ \operatorname{rank} x_\beta^{\beta'} = \operatorname{rank} \tilde{x}_\beta^{\beta'}. \tag{18}$$

Then, using the terminology of [3], for any discrete set $B \subset \mathbb{R}$ containing α and no accumulation point, the B-discretizations $\mathcal{X}_B$ and $\tilde{\mathcal{X}}_B$ of the persistence

modules $\mathcal{X}$ and $\tilde{\mathcal{X}}$ satisfy $\mathrm{D}\mathcal{X}_B \cap Q_\alpha^{\mathrm{NE}} = \mathrm{D}\tilde{\mathcal{X}}_B \cap Q_\alpha^{\mathrm{NE}}$. It follows then from the definition of persistence diagram[12] that $\mathrm{D}\mathcal{X} \cap Q_\alpha^{\mathrm{NE}} = \mathrm{D}\tilde{\mathcal{X}} \cap Q_\alpha^{\mathrm{NE}}$. Let $\gamma_X : \mathrm{D}\mathcal{X} \to \mathrm{D}\tilde{\mathcal{X}}$ be a multi-bijection such that γ_X and γ_X^{-1} leave the points within Q_α^{NE} fixed. We will now show that the total multiplicities of $\mathrm{D}\mathcal{X}$ and $\mathrm{D}\tilde{\mathcal{X}}$ are equal within any given vertical half-line $\{\beta'\} \times [-\infty, \beta]$ where $\beta' > \beta \geq \alpha$, which will enable us to further assume that γ_X and γ_X^{-1} only move the points vertically within the lower-right quadrant Q_α^{SE}, as illustrated in Figure 7 (right).

Given any $\eta > 0$, we discretize $\mathcal{X}$ and $\tilde{\mathcal{X}}$ over the integer scale $\alpha + \eta\mathbb{Z}$, to get respectively $\mathcal{X}_{\alpha+\eta\mathbb{Z}}$ and $\tilde{\mathcal{X}}_{\alpha+\eta\mathbb{Z}}$. Their persistence diagrams are then snapped onto the regular grid $(\alpha+\eta\mathbb{Z})\times(\alpha+\eta\mathbb{Z})$, as per Theorem 3.7 of [3] (the snapping directions are in fact reversed here, since time flows from $+\infty$ to $-\infty$). For any integers $m > n \in \mathbb{Z}$, the total multiplicity of $\mathrm{D}\mathcal{X}_{\alpha+\eta\mathbb{Z}}$ within the vertical half-line $\{\alpha + m\eta\} \times [-\infty, \alpha + n\eta]$ is given by the sum of the multiplicities of the points $(\alpha + m\eta,\ \alpha + (n-l)\eta)$ for l ranging over $\mathbb{N} \cup \{+\infty\}$:

$$\mu^{\mathrm{tot}}_{\eta,m,n}(\mathrm{D}\mathcal{X}_{\alpha+\eta\mathbb{Z}}) = \mu(\alpha + m\eta, -\infty) + \sum_{l\in\mathbb{N}} \mu(\alpha + m\eta,\ \alpha + (n-l)\eta). \tag{19}$$

By definition[13], the multiplicity of each point $(\alpha + m\eta,\ \alpha + (n-l)\eta)$, $l \in \mathbb{N}$, is given by:

$$\begin{aligned} \mu(\alpha + m\eta,\ \alpha + (n-l)\eta) \quad &= \quad \left(\operatorname{rank} x^{\alpha+(n-l+1)\eta}_{\alpha+m\eta} - \operatorname{rank} x^{\alpha+(n-l+1)\eta}_{\alpha+(m+1)\eta}\right) \\ &\quad - \left(\operatorname{rank} x^{\alpha+(n-l)\eta}_{\alpha+m\eta} - \operatorname{rank} x^{\alpha+(n-l)\eta}_{\alpha+(m+1)\eta}\right). \end{aligned} \tag{20}$$

Since the persistence module $\mathcal{X}$ is tame, its persistence diagram (and thus the one of its discretization $\mathcal{X}_{\alpha+\eta\mathbb{Z}}$) contains only finitely many points off the diagonal. This implies that the sum in Eq. (19) is a finite sum, where the terms of Eq. (20) pairwise cancel out:

$$\begin{aligned} \textstyle\sum_{l\in\mathbb{N}} \mu(\alpha + m\eta,\ \alpha + (n-l)\eta) \quad &= \quad \textstyle\sum_{l=0}^{l_{\max}} \mu(\alpha + m\eta,\ \alpha + (n-l)\eta) \\ &= \quad \left(\operatorname{rank} x^{\alpha+(n+1)\eta}_{\alpha+m\eta} - \operatorname{rank} x^{\alpha+(n+1)\eta}_{\alpha+(m+1)\eta}\right) \\ &\quad - \left(\operatorname{rank} x^{\alpha+(n-l_{\max})\eta}_{\alpha+m\eta} - \operatorname{rank} x^{\alpha+(n-l_{\max})\eta}_{\alpha+(m+1)\eta}\right). \end{aligned}$$

The second term of the subtraction is precisely the multiplicity of $(\alpha+m\eta, -\infty)$, which cancels out with $\mu(\alpha+m\eta, -\infty)$ in Eq. (19). Hence, the total multiplicity of $\mathrm{D}\mathcal{X}_{\alpha+\eta\mathbb{Z}}$ within the vertical half-line $\{\alpha + m\eta\} \times [-\infty, \alpha + n\eta]$ is:

$$\mu^{\mathrm{tot}}_{\eta,m,n}(\mathrm{D}\mathcal{X}_{\alpha+\eta\mathbb{Z}}) = \operatorname{rank} x^{\alpha+(n+1)\eta}_{\alpha+m\eta} - \operatorname{rank} x^{\alpha+(n+1)\eta}_{\alpha+(m+1)\eta}. \tag{21}$$

The same is true for $\tilde{\mathcal{X}}_{\alpha+\eta\mathbb{Z}}$ (which is tame since $\mathcal{X}_{\alpha+\eta\mathbb{Z}}$ is), namely:

$$\mu^{\mathrm{tot}}_{\eta,m,n}(\mathrm{D}\tilde{\mathcal{X}}_{\alpha+\eta\mathbb{Z}}) = \operatorname{rank} \tilde{x}^{\alpha+(n+1)\eta}_{\alpha+m\eta} - \operatorname{rank} \tilde{x}^{\alpha+(n+1)\eta}_{\alpha+(m+1)\eta}. \tag{22}$$

Assume that $m > n \geq 0$, *i.e.* that the endpoint of the vertical half-line lies on or above the horizontal line $y = \alpha$. Under this asssumption, Eqs. (18) and

[12] See Definition 3.6 of [3].

[13] See Definition 3.2 of [3] and recall that coordinates are reversed here because time flows from $+\infty$ to $-\infty$.

(21)-(22) imply that $\mu^{\mathrm{tot}}_{\eta,m,n}(\mathrm{D}\mathcal{X}_{\alpha+\eta\mathbb{Z}}) = \mu^{\mathrm{tot}}_{\eta,m,n}(\mathrm{D}\tilde{\mathcal{X}}_{\alpha+\eta\mathbb{Z}})$. Since this is true for all $\eta > 0$, we deduce that the total multiplicities of the diagrams $\mathrm{D}\mathcal{X}$ and $\mathrm{D}\tilde{\mathcal{X}}$ in any vertical half-line $\{\beta'\} \times [-\infty, \beta]$ with $\beta' > \beta \geq \alpha$ are the same. We may thus further assume that the multi-bijection $\gamma_X : \mathrm{D}\mathcal{X} \to \mathrm{D}\tilde{\mathcal{X}}$ defined above is such that γ_X and γ_X^{-1} move the points within the lower-right quadrant Q^{SE}_α vertically, in addition to keeping the points within the upper-right quadrant Q^{NE}_α fixed.

The same construction as in Eq. (16) can be applied to the tame persistence module $\mathcal{Y}$, thus yielding another tame persistence module $\tilde{\mathcal{Y}}$. By the same sequence of arguments as above, we know that there is a multi-bijection $\gamma_Y : \mathrm{D}\mathcal{Y} \to \mathrm{D}\tilde{\mathcal{Y}}$ such that γ_Y and γ_Y^{-1} move the points within Q^{SE}_α vertically while keeping the points within Q^{NE}_α fixed.

Observe now that the newly-introduced persistence modules $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ are (strongly) ε-interleaved. Indeed, let $\{\phi_\beta : X^\beta \to Y^{\beta-\varepsilon}\}_{\beta\geq\alpha}$ and $\{\psi_\beta : Y^\beta \to X^{\beta-\varepsilon}\}_{\beta\geq\alpha}$ be two families of homomorphisms that make $\mathcal{X}$ and $\mathcal{Y}$ (strongly) ε-interleaved above time α. We define two new families of homomorphisms between $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$, indexed over $\mathbb{R}$, as follows:

$$\begin{cases} \forall \beta \geq \alpha,\ \tilde{\phi}_\beta = \phi_\beta \text{ and } \tilde{\psi}_\beta = \psi_\beta, \\ \forall \beta < \alpha,\ \tilde{\phi}_\beta = 0 \text{ and } \tilde{\psi}_\beta = 0. \end{cases}$$

The fact that these two families of homomorphisms make the diagrams of Eq. (6) commute for all $\beta' \geq \beta \geq \alpha$ comes from the fact that $\{\phi_\beta\}_{\beta\geq\alpha}$ and $\{\psi_\beta\}_{\beta\geq\alpha}$ themselves make the diagrams commute. The fact that the families $\{\tilde{\phi}_\beta\}_{\beta\in\mathbb{R}}$ and $\{\tilde{\psi}_\beta\}_{\beta\in\mathbb{R}}$ make the diagrams commute across and below time α comes from the fact that they are identically zero below time α. Thus, $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ are (strongly) ε-interleaved over whole $\mathbb{R}$, which implies by the Extended Stability Theorem[14] that there is a multi-bijection $\tilde{\gamma} : \mathrm{D}\tilde{\mathcal{X}} \to \mathrm{D}\tilde{\mathcal{Y}}$ that moves the points by at most ε in the l^∞-distance. The map $\gamma_Y^{-1} \circ \tilde{\gamma} \circ \gamma_X$ is then a multi-bijection $\mathrm{D}\mathcal{X} \to \mathrm{D}\mathcal{Y}$ satisfying assertions (i) through (iv) of Theorem 4.5. This concludes the proof of Lemma 4.6.

[14] See Theorem 4.4 of [3].

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399