Answer to reviewer 1

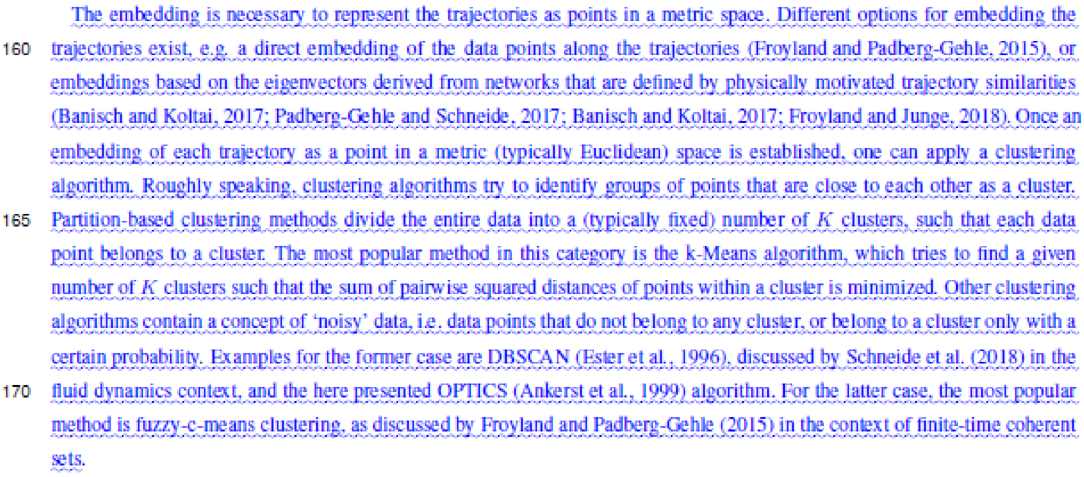
General answer:
We thank the reviewer for the detailed comments on the paper. They have helped us to significantly improve on the readability and clarity in the revised version. We have implemented changes for every comment raised by the reviewer.

Please note:
The images in this file are excerpts of the revised version in latexdiff. Please apologize the formatting problems of latexdiff that cuts off references at line breaks. This is not the case in the revised version.

Comment 1
I find there to be room for improvement on a few presentational issues. (i). There seems to be an assumption of familiarity with other clustering methods. The paper would be more accessible, and therefore useful, if the authors took just slightly more time in defining new terms and in providing the intuitive content of mathematical concepts.

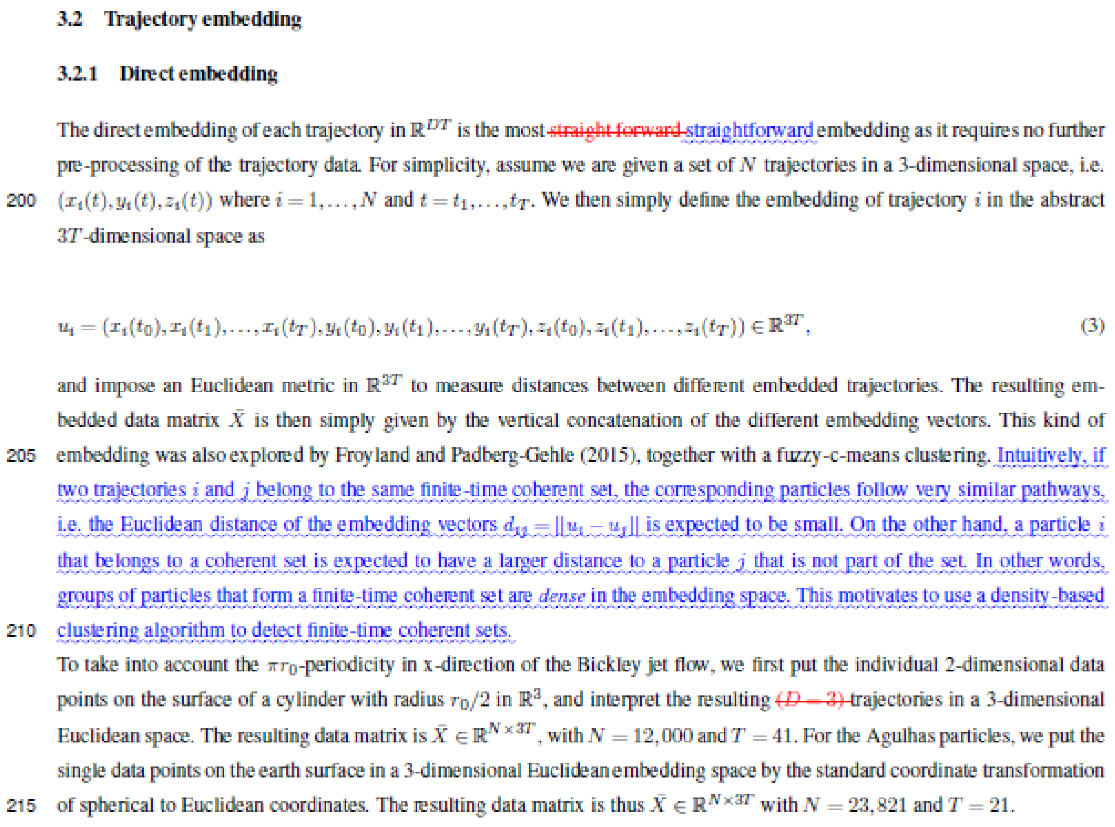
Answer to comment 1
Thank you for your comment. We have made the following changes in the revised version:

1. Additional paragraph in the methods section that briefly describes why embedding / clustering is necessary, and also explains in one sentence what k-Means does



The embedding is necessary to represent the trajectories as points in a metric space. Different options for embedding the trajectories exist, e.g. a direct embedding of the data points along the trajectories (Froyland and Padberg-Gehle, 2015), or embeddings based on the eigenvectors derived from networks that are defined by physically motivated trajectory similarities (Banisch and Koltai, 2017; Padberg-Gehle and Schneide, 2017; Banisch and Koltai, 2017; Froyland and Junge, 2018). Once an embedding of each trajectory as a point in a metric (typically Euclidean) space is established, one can apply a clustering algorithm. Roughly speaking, clustering algorithms try to identify groups of points that are close to each other as a cluster. Partition-based clustering methods divide the entire data into a (typically fixed) number of K clusters, such that each data point belongs to a cluster. The most popular method in this category is the k-Means algorithm, which tries to find a given number of K clusters such that the sum of pairwise squared distances of points within a cluster is minimized. Other clustering algorithms contain a concept of 'noisy' data, i.e. data points that do not belong to any cluster, or belong to a cluster only with a certain probability. Examples for the former case are DBSCAN (Ester et al., 1996), discussed by Schneide et al. (2018) in the fluid dynamics context, and the here presented OPTICS (Ankerst et al., 1999) algorithm. For the latter case, the most popular method is fuzzy-c-means clustering, as discussed by Froyland and Padberg-Gehle (2015) in the context of finite-time coherent sets.

2. Additional explanation in the methods section that describes why the embedding we choose is expected to create a detectable signal for OPTICS.



3.2 Trajectory embedding

3.2.1 Direct embedding

The direct embedding of each trajectory in $\mathbb{R}^{DT}$ is the most ~~straight forward~~ straightforward embedding as it requires no further pre-processing of the trajectory data. For simplicity, assume we are given a set of N trajectories in a 3-dimensional space, i.e. $(x_i(t), y_i(t), z_i(t))$ where $i = 1, \ldots, N$ and $t = t_1, \ldots, t_T$. We then simply define the embedding of trajectory i in the abstract $3T$-dimensional space as

$$u_i = (x_i(t_0), x_i(t_1), \ldots, x_i(t_T), y_i(t_0), y_i(t_1), \ldots, y_i(t_T), z_i(t_0), z_i(t_1), \ldots, z_i(t_T)) \in \mathbb{R}^{3T}, \tag{3}$$

and impose an Euclidean metric in $\mathbb{R}^{3T}$ to measure distances between different embedded trajectories. The resulting embedded data matrix $\bar{X}$ is then simply given by the vertical concatenation of the different embedding vectors. This kind of embedding was also explored by Froyland and Padberg-Gehle (2015), together with a fuzzy-c-means clustering. Intuitively, if two trajectories i and j belong to the same finite-time coherent set, the corresponding particles follow very similar pathways, i.e. the Euclidean distance of the embedding vectors $d_{ij} = ||u_i - u_j||$ is expected to be small. On the other hand, a particle i that belongs to a coherent set is expected to have a larger distance to a particle j that is not part of the set. In other words, groups of particles that form a finite-time coherent set are *dense* in the embedding space. This motivates to use a density-based clustering algorithm to detect finite-time coherent sets.

To take into account the πr_0-periodicity in x-direction of the Bickley jet flow, we first put the individual 2-dimensional data points on the surface of a cylinder with radius $r_0/2$ in $\mathbb{R}^3$, and interpret the resulting ~~$(D=3)$~~ trajectories in a 3-dimensional Euclidean space. The resulting data matrix is $\bar{X} \in \mathbb{R}^{N \times 3T}$, with $N = 12,000$ and $T = 41$. For the Agulhas particles, we put the single data points on the earth surface in a 3-dimensional Euclidean embedding space by the standard coordinate transformation of spherical to Euclidean coordinates. The resulting data matrix is thus $\bar{X} \in \mathbb{R}^{N \times 3T}$ with $N = 23,821$ and $T = 21$.

Comment 2

(ii) I find it a little strange that some figures are presented in the appendix, but discussed only in the main text. Some of these make good illustrations of the performance of the method with respect to others, e.g. D1&D2. I feel this tends to negatively impact the narrative. If figures are discussed in the main text, I would present them there also.

Answer to comment 2

Thank you for this comment. We agree with the reviewer and have now included the clustering results of the classical MDS method in the main text. In the revised version, we provide the results of OPTICS together with its comparison to k-Means for both of the model flows. We have decided to leave the discussion of the embedded network of Padberg-Gehle and Schneide (2017) together with the previous figures D1-D3 in the appendix. This is because the major focus of the paper is the OPTICS clustering on the direct embedding of the trajectories, as this removes the need of several parameters compared to Padberg-Gehle and Schneide (2017), such as the cut-off parameter *d*, and the embedding dimensions. A reader that is interested in the application of OPTICS to the spectral embedding of Padberg-Gehle and Schneide (2017) gets a full account on that topic in the appendix. We do not discuss these results in the main text, but only mention them quickly. The actual discussion is contained in appendix C of the revised version.

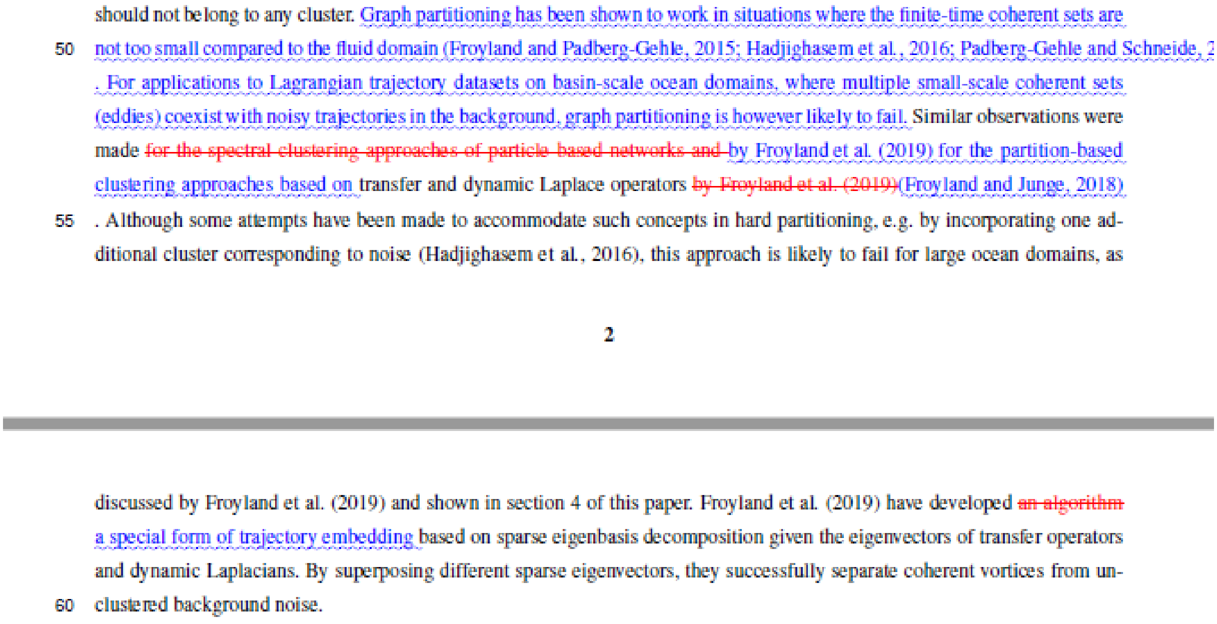
Comment 3

(iii) The paper has a highly technical focus throughout. More framing of the import of this problem at the start and end would have been appreciated.

Answer to comment 3

Thank you for this suggestion. We have now added more content on the problem itself, i.e. the detection of many small coherent structures in a large, noisy ocean domain.

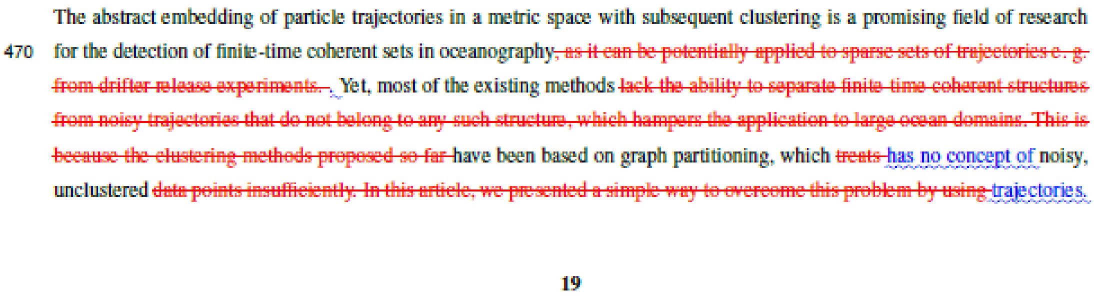
1. Introduction



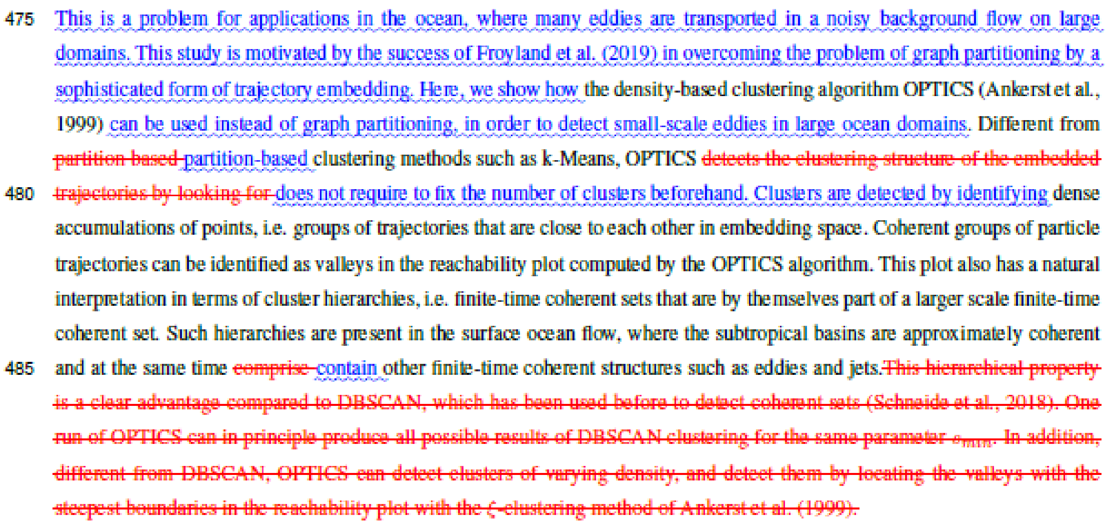
should not belong to any cluster. Graph partitioning has been shown to work in situations where the finite-time coherent sets are not too small compared to the fluid domain (Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2 . For applications to Lagrangian trajectory datasets on basin-scale ocean domains, where multiple small-scale coherent sets (eddies) coexist with noisy trajectories in the background, graph partitioning is however likely to fail. Similar observations were made ~~for the spectral clustering approaches of particle based networks and~~ by Froyland et al. (2019) for the partition-based clustering approaches based on transfer and dynamic Laplace operators ~~by Froyland et al. (2019)~~(Froyland and Junge, 2018) . Although some attempts have been made to accommodate such concepts in hard partitioning, e.g. by incorporating one additional cluster corresponding to noise (Hadjighasem et al., 2016), this approach is likely to fail for large ocean domains, as

discussed by Froyland et al. (2019) and shown in section 4 of this paper. Froyland et al. (2019) have developed ~~an algorithm~~ a special form of trajectory embedding based on sparse eigenbasis decomposition given the eigenvectors of transfer operators and dynamic Laplacians. By superposing different sparse eigenvectors, they successfully separate coherent vortices from unclustered background noise.

2. Conclusion

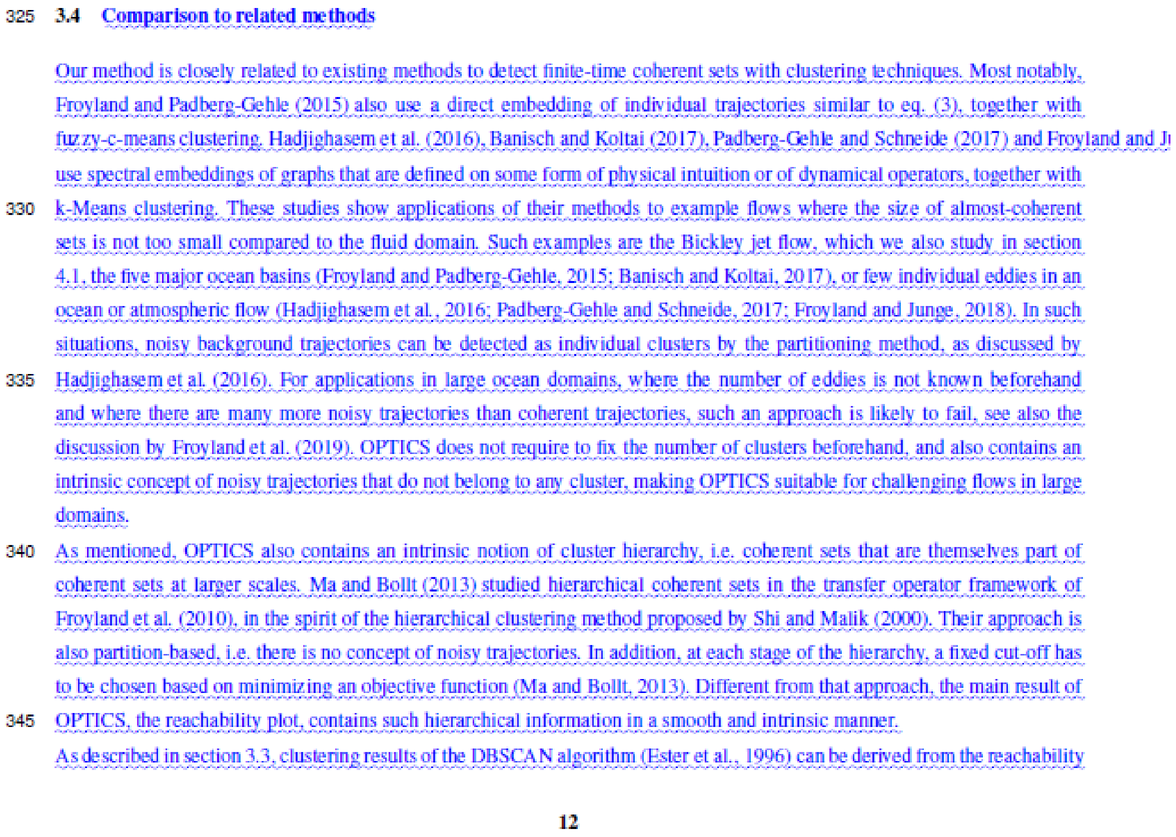


The abstract embedding of particle trajectories in a metric space with subsequent clustering is a promising field of research for the detection of finite-time coherent sets in oceanography~~, as it can be potentially applied to sparse sets of trajectories e. g. from drifter release experiments.~~ . Yet, most of the existing methods ~~lack the ability to separate finite-time coherent structures from noisy trajectories that do not belong to any such structure, which hampers the application to large ocean domains. This is because the clustering methods proposed so far~~ have been based on graph partitioning, which ~~treats~~ has no concept of noisy, unclustered ~~data points insufficiently. In this article, we presented a simple way to overcome this problem by using~~ trajectories.



This is a problem for applications in the ocean, where many eddies are transported in a noisy background flow on large domains. This study is motivated by the success of Froyland et al. (2019) in overcoming the problem of graph partitioning by a sophisticated form of trajectory embedding. Here, we show how the density-based clustering algorithm OPTICS (Ankerst et al., 1999) can be used instead of graph partitioning, in order to detect small-scale eddies in large ocean domains. Different from ~~partition based~~ partition-based clustering methods such as k-Means, OPTICS ~~detects the clustering structure of the embedded trajectories by looking for~~ does not require to fix the number of clusters beforehand. Clusters are detected by identifying dense accumulations of points, i.e. groups of trajectories that are close to each other in embedding space. Coherent groups of particle trajectories can be identified as valleys in the reachability plot computed by the OPTICS algorithm. This plot also has a natural interpretation in terms of cluster hierarchies, i.e. finite-time coherent sets that are by themselves part of a larger scale finite-time coherent set. Such hierarchies are present in the surface ocean flow, where the subtropical basins are approximately coherent and at the same time ~~comprise~~ contain other finite-time coherent structures such as eddies and jets.~~This hierarchical property is a clear advantage compared to DBSCAN, which has been used before to detect coherent sets (Schneide et al., 2018). One run of OPTICS can in principle produce all possible results of DBSCAN clustering for the same parameter ϵ_{min}. In addition, different from DBSCAN, OPTICS can detect clusters of varying density, and detect them by locating the valleys with the steepest boundaries in the reachability plot with the ξ-clustering method of Ankerst et al. (1999).~~

3. We have now also discussed the relation of our method to existing methods. In particular, we stress that we focus on a new clustering algorithm instead of a new form of embedding, as e.g. done by Froyland et al. (2018).

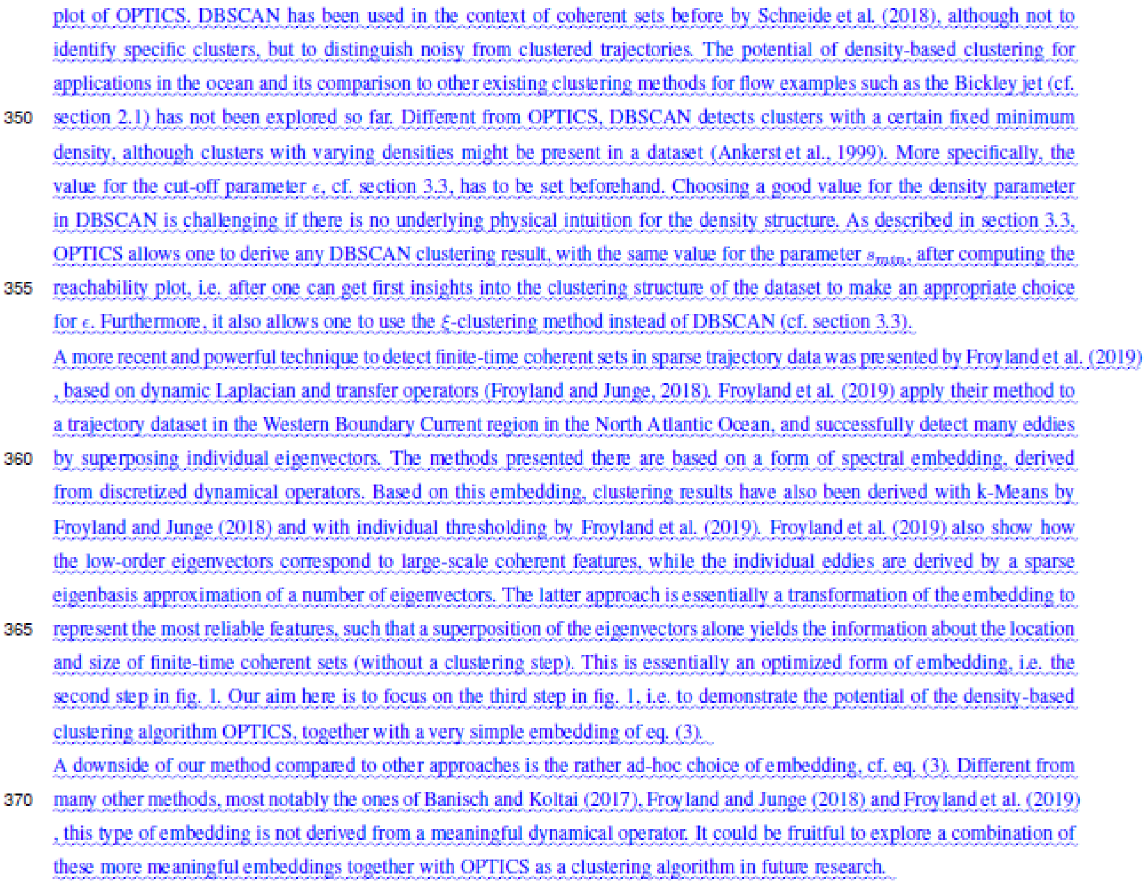


3.4 Comparison to related methods

Our method is closely related to existing methods to detect finite-time coherent sets with clustering techniques. Most notably, Froyland and Padberg-Gehle (2015) also use a direct embedding of individual trajectories similar to eq. (3), together with fuzzy-c-means clustering. Hadjighasem et al. (2016), Banisch and Koltai (2017), Padberg-Gehle and Schneide (2017) and Froyland and J use spectral embeddings of graphs that are defined on some form of physical intuition or of dynamical operators, together with k-Means clustering. These studies show applications of their methods to example flows where the size of almost-coherent sets is not too small compared to the fluid domain. Such examples are the Bickley jet flow, which we also study in section 4.1, the five major ocean basins (Froyland and Padberg-Gehle, 2015; Banisch and Koltai, 2017), or few individual eddies in an ocean or atmospheric flow (Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2017; Froyland and Junge, 2018). In such situations, noisy background trajectories can be detected as individual clusters by the partitioning method, as discussed by Hadjighasem et al. (2016). For applications in large ocean domains, where the number of eddies is not known beforehand and where there are many more noisy trajectories than coherent trajectories, such an approach is likely to fail, see also the discussion by Froyland et al. (2019). OPTICS does not require to fix the number of clusters beforehand, and also contains an intrinsic concept of noisy trajectories that do not belong to any cluster, making OPTICS suitable for challenging flows in large domains.

As mentioned, OPTICS also contains an intrinsic notion of cluster hierarchy, i.e. coherent sets that are themselves part of coherent sets at larger scales. Ma and Bollt (2013) studied hierarchical coherent sets in the transfer operator framework of Froyland et al. (2010), in the spirit of the hierarchical clustering method proposed by Shi and Malik (2000). Their approach is also partition-based, i.e. there is no concept of noisy trajectories. In addition, at each stage of the hierarchy, a fixed cut-off has to be chosen based on minimizing an objective function (Ma and Bollt, 2013). Different from that approach, the main result of OPTICS, the reachability plot, contains such hierarchical information in a smooth and intrinsic manner.

As described in section 3.3, clustering results of the DBSCAN algorithm (Ester et al., 1996) can be derived from the reachability



plot of OPTICS. DBSCAN has been used in the context of coherent sets before by Schneide et al. (2018), although not to identify specific clusters, but to distinguish noisy from clustered trajectories. The potential of density-based clustering for applications in the ocean and its comparison to other existing clustering methods for flow examples such as the Bickley jet (cf. section 2.1) has not been explored so far. Different from OPTICS, DBSCAN detects clusters with a certain fixed minimum density, although clusters with varying densities might be present in a dataset (Ankerst et al., 1999). More specifically, the value for the cut-off parameter ϵ, cf. section 3.3, has to be set beforehand. Choosing a good value for the density parameter in DBSCAN is challenging if there is no underlying physical intuition for the density structure. As described in section 3.3, OPTICS allows one to derive any DBSCAN clustering result, with the same value for the parameter s_{min}, after computing the reachability plot, i.e. after one can get first insights into the clustering structure of the dataset to make an appropriate choice for ϵ. Furthermore, it also allows one to use the ξ-clustering method instead of DBSCAN (cf. section 3.3).

A more recent and powerful technique to detect finite-time coherent sets in sparse trajectory data was presented by Froyland et al. (2019) , based on dynamic Laplacian and transfer operators (Froyland and Junge, 2018). Froyland et al. (2019) apply their method to a trajectory dataset in the Western Boundary Current region in the North Atlantic Ocean, and successfully detect many eddies by superposing individual eigenvectors. The methods presented there are based on a form of spectral embedding, derived from discretized dynamical operators. Based on this embedding, clustering results have also been derived with k-Means by Froyland and Junge (2018) and with individual thresholding by Froyland et al. (2019). Froyland et al. (2019) also show how the low-order eigenvectors correspond to large-scale coherent features, while the individual eddies are derived by a sparse eigenbasis approximation of a number of eigenvectors. The latter approach is essentially a transformation of the embedding to represent the most reliable features, such that a superposition of the eigenvectors alone yields the information about the location and size of finite-time coherent sets (without a clustering step). This is essentially an optimized form of embedding, i.e. the second step in fig. 1. Our aim here is to focus on the third step in fig. 1, i.e. to demonstrate the potential of the density-based clustering algorithm OPTICS, together with a very simple embedding of eq. (3).

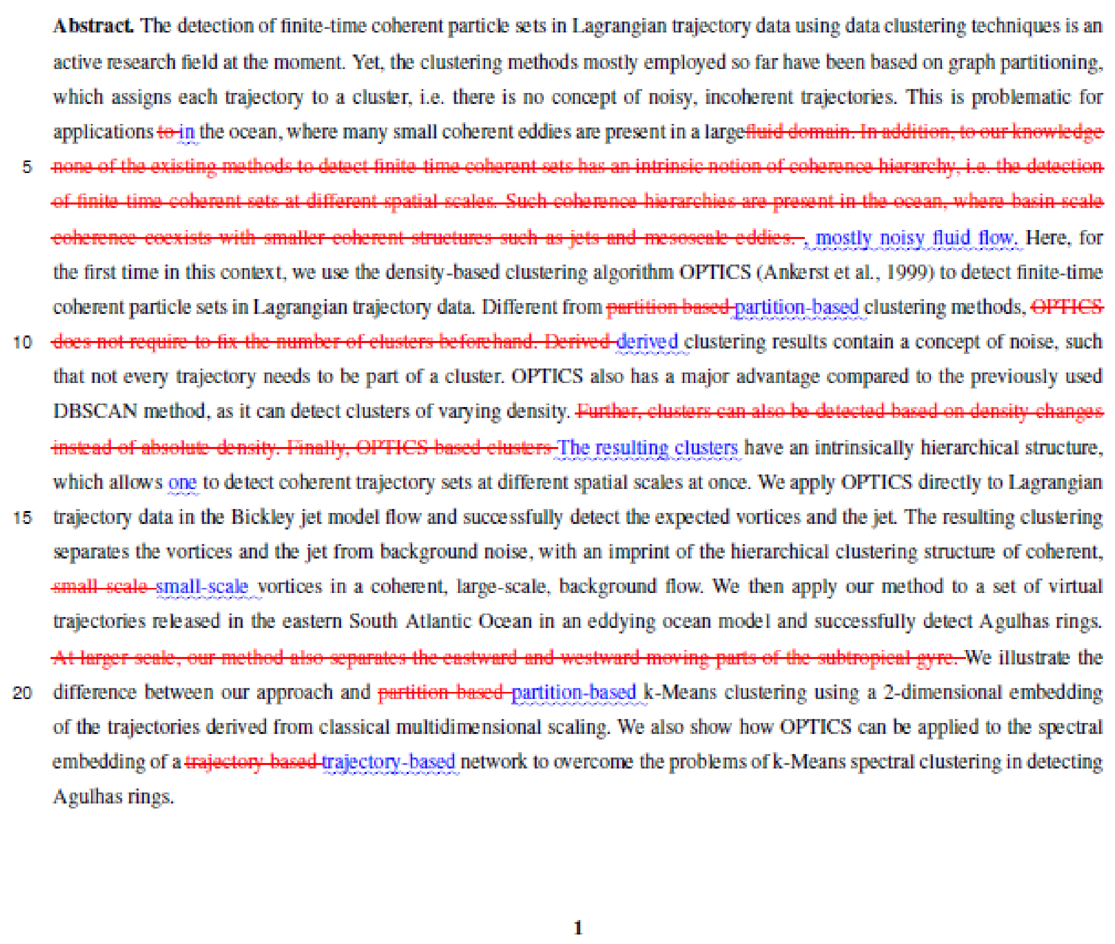
A downside of our method compared to other approaches is the rather ad-hoc choice of embedding, cf. eq. (3). Different from many other methods, most notably the ones of Banisch and Koltai (2017), Froyland and Junge (2018) and Froyland et al. (2019) , this type of embedding is not derived from a meaningful dynamical operator. It could be fruitful to explore a combination of these more meaningful embeddings together with OPTICS as a clustering algorithm in future research.

Comment 4

(iv) For a short paper, the abstract is perhaps disproportionately long.

Answer to comment 4

Thanks for noting. We have shortened the abstract a bit in the new version.



Abstract. The detection of finite-time coherent particle sets in Lagrangian trajectory data using data clustering techniques is an active research field at the moment. Yet, the clustering methods mostly employed so far have been based on graph partitioning, which assigns each trajectory to a cluster, i.e. there is no concept of noisy, incoherent trajectories. This is problematic for applications ~~to~~ in the ocean, where many small coherent eddies are present in a large~~fluid domain. In addition, to our knowledge none of the existing methods to detect finite-time coherent sets has an intrinsic notion of coherence hierarchy, i.e. the detection of finite-time coherent sets at different spatial scales. Such coherence hierarchies are present in the ocean, where basin-scale coherence coexists with smaller coherent structures such as jets and mesoscale eddies.~~ , mostly noisy fluid flow. Here, for the first time in this context, we use the density-based clustering algorithm OPTICS (Ankerst et al., 1999) to detect finite-time coherent particle sets in Lagrangian trajectory data. Different from ~~partition based~~ partition-based clustering methods, ~~OPTICS does not require to fix the number of clusters beforehand. Derived~~ derived clustering results contain a concept of noise, such that not every trajectory needs to be part of a cluster. OPTICS also has a major advantage compared to the previously used DBSCAN method, as it can detect clusters of varying density. ~~Further, clusters can also be detected based on density changes instead of absolute density. Finally, OPTICS-based clusters~~ The resulting clusters have an intrinsically hierarchical structure, which allows one to detect coherent trajectory sets at different spatial scales at once. We apply OPTICS directly to Lagrangian trajectory data in the Bickley jet model flow and successfully detect the expected vortices and the jet. The resulting clustering separates the vortices and the jet from background noise, with an imprint of the hierarchical clustering structure of coherent, ~~small scale~~ small-scale vortices in a coherent, large-scale, background flow. We then apply our method to a set of virtual trajectories released in the eastern South Atlantic Ocean in an eddying ocean model and successfully detect Agulhas rings. ~~At larger scale, our method also separates the eastward and westward moving parts of the subtropical gyre.~~ We illustrate the difference between our approach and ~~partition based~~ partition-based k-Means clustering using a 2-dimensional embedding of the trajectories derived from classical multidimensional scaling. We also show how OPTICS can be applied to the spectral embedding of a ~~trajectory based~~ trajectory-based network to overcome the problems of k-Means spectral clustering in detecting Agulhas rings.

1

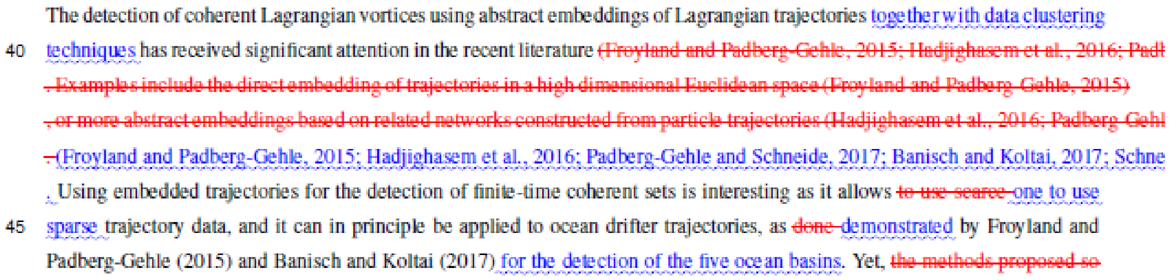
Comment 5

There is one point raised in the paper that I felt required more elaboration. A selling point the authors bring up for this method is that it can in principle be applied to real-world trajectory data, see line 86 and also line 306. This is true but incomplete. Real-world Lagrangian instruments are sufficiently sparse that it is rare to find more than one in the same eddy at the same time. Thus, the application presented herein—finding eddies is idealized configurations—is not really relevant for how one would apply this method to real-world trajectories. The data density used here is orders of magnitude greater than for real-world instruments. Since the authors bring this up as an advantage of the method, a more fair and nuanced discussion of its potential and limitations with respect to real-world data is called for. I would say, rather, that the method seems more suitable in application to model data or virtual trajectories from altimetry, where it benefits from a simplicity with respect to some other proposed methods.

Answer to comment 5

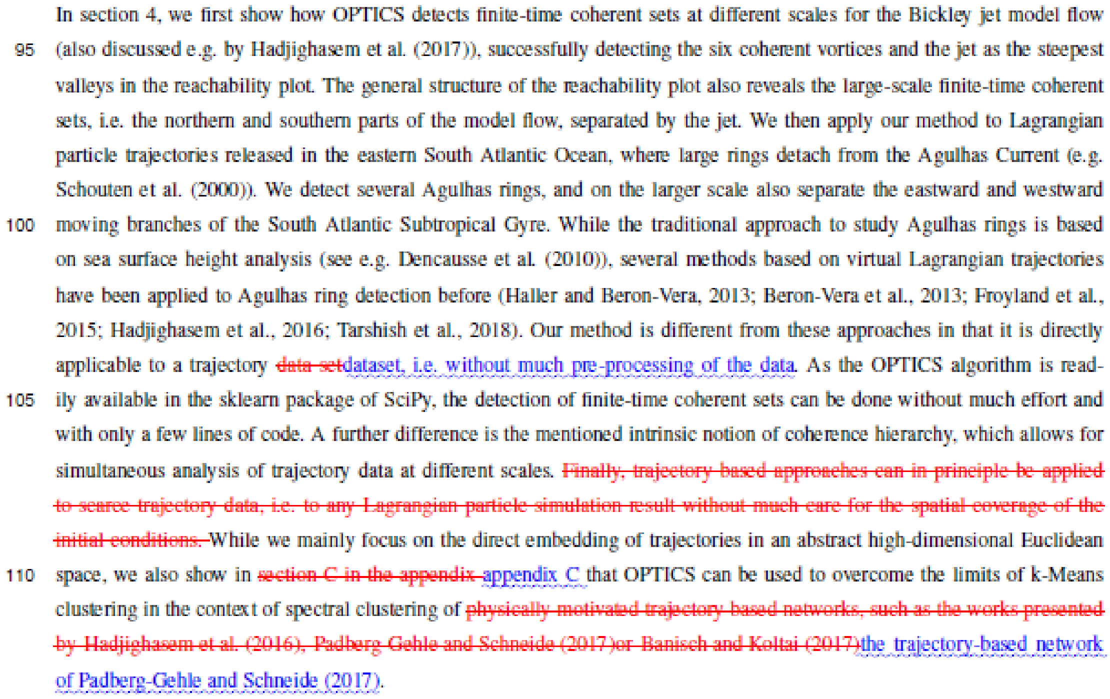
The reviewer is correct that an application of our method to real drifters to detect eddies is not possible due to the limited coverage of drifter data. Note that two studies applied their methods to real drifters, as we mentioned in the introduction (Froyland and Padberg-Gehle (2015) and Banisch and Koltai (2017)), however to detect the five major ocean basins and not eddies. In the new version, we omit the reference to real ocean drifters at other places but the introduction, where we now explicitly mention the application to ocean basins (and not eddies).

1. Changes in introduction to clarify that trajectory-based clustering has been applied to real drifter data only in the context of detecting the ocean basins, not individual eddies.



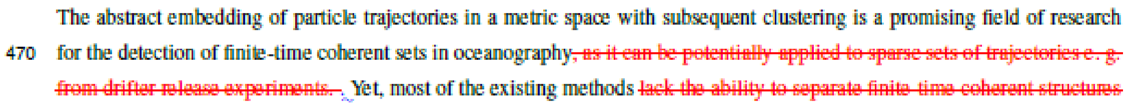
The detection of coherent Lagrangian vortices using abstract embeddings of Lagrangian trajectories together with data clustering techniques has received significant attention in the recent literature ~~(Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padl~~ ~~. Examples include the direct embedding of trajectories in a high dimensional Euclidean space (Froyland and Padberg-Gehle, 2015)~~ ~~, or more abstract embeddings based on related networks constructed from particle trajectories (Hadjighasem et al., 2016; Padberg-Gehl~~ ~~.~~(Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2017; Banisch and Koltai, 2017; Schne . Using embedded trajectories for the detection of finite-time coherent sets is interesting as it allows ~~to use scarce~~ one to use sparse trajectory data, and it can in principle be applied to ocean drifter trajectories, as ~~done~~ demonstrated by Froyland and Padberg-Gehle (2015) and Banisch and Koltai (2017) for the detection of the five ocean basins. Yet, ~~the methods proposed so~~

2. End of the introduction



In section 4, we first show how OPTICS detects finite-time coherent sets at different scales for the Bickley jet model flow (also discussed e.g. by Hadjighasem et al. (2017)), successfully detecting the six coherent vortices and the jet as the steepest valleys in the reachability plot. The general structure of the reachability plot also reveals the large-scale finite-time coherent sets, i.e. the northern and southern parts of the model flow, separated by the jet. We then apply our method to Lagrangian particle trajectories released in the eastern South Atlantic Ocean, where large rings detach from the Agulhas Current (e.g. Schouten et al. (2000)). We detect several Agulhas rings, and on the larger scale also separate the eastward and westward moving branches of the South Atlantic Subtropical Gyre. While the traditional approach to study Agulhas rings is based on sea surface height analysis (see e.g. Dencausse et al. (2010)), several methods based on virtual Lagrangian trajectories have been applied to Agulhas ring detection before (Haller and Beron-Vera, 2013; Beron-Vera et al., 2013; Froyland et al., 2015; Hadjighasem et al., 2016; Tarshish et al., 2018). Our method is different from these approaches in that it is directly applicable to a trajectory ~~data set~~dataset, i.e. without much pre-processing of the data. As the OPTICS algorithm is readily available in the sklearn package of SciPy, the detection of finite-time coherent sets can be done without much effort and with only a few lines of code. A further difference is the mentioned intrinsic notion of coherence hierarchy, which allows for simultaneous analysis of trajectory data at different scales. ~~Finally, trajectory based approaches can in principle be applied to scarce trajectory data, i.e. to any Lagrangian particle simulation result without much care for the spatial coverage of the initial conditions.~~ While we mainly focus on the direct embedding of trajectories in an abstract high-dimensional Euclidean space, we also show in ~~section C in the appendix~~ appendix C that OPTICS can be used to overcome the limits of k-Means clustering in the context of spectral clustering of ~~physically motivated trajectory based networks, such as the works presented by Hadjighasem et al. (2016), Padberg-Gehle and Schneide (2017)or Banisch and Koltai (2017)~~the trajectory-based network of Padberg-Gehle and Schneide (2017).

2. First sentence in conclusion



The abstract embedding of particle trajectories in a metric space with subsequent clustering is a promising field of research for the detection of finite-time coherent sets in oceanography~~, as it can be potentially applied to sparse sets of trajectories e. g. from drifter release experiments.~~ . Yet, most of the existing methods ~~lack the ability to separate finite-time coherent structures~~

Comment 6
Line 99. Do you not want to cite Bickley? My understanding is that the term “Bickley jet” itself is used to refer to a steady solution with a sechˆ2 u-velocity, see e.g. Swaters (1999). The authors’ Eq. (2) is an added perturbation. As read, it sounds like the whole thing is the Bickley jet.

Answer to comment 6
Thank you for this comment and the careful check of our references of the flow. You are indeed right that the Bickley Jet is a steady, sechˆ2 velocity profile. We have added the reference to Bickley now,

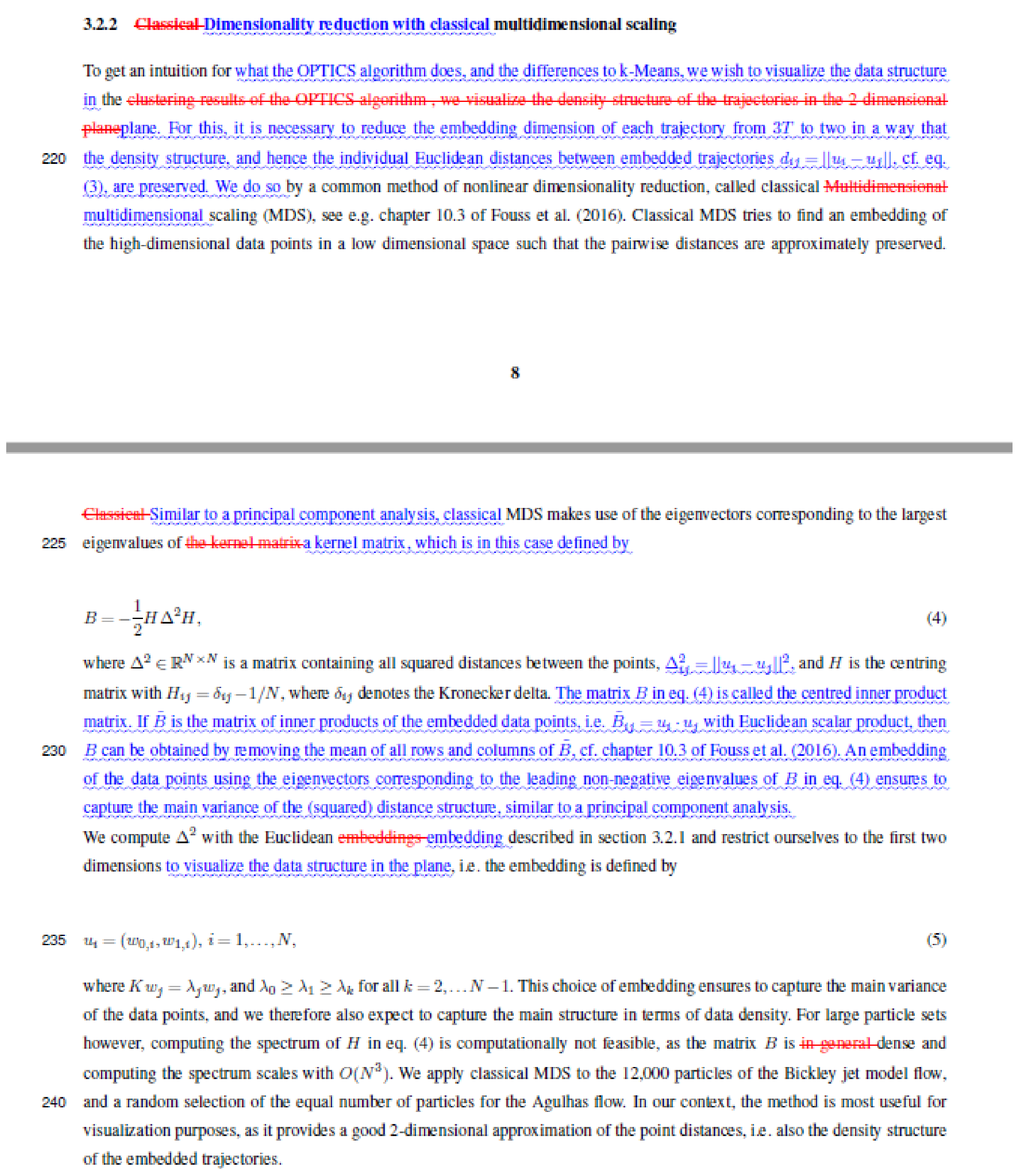
together with a reference to the paper of del Castillo-Negrete and Morrison (1993), where the perturbed form of the jet is motivated.

Comment 7

Section 3.2.2. I didn’t really understand this section, or what B is encoding in Eq. (4). A more intuitive description would be helpful. When you say, “pairwise distances are approximately preserved”, this is with respect to what? Also, why are two dimensions chosen?

Answer to comment 7

Thank you for this comment. In the new version, we elaborate more on the intuitive goal of classical MDS in this section. We choose two dimensions because we wish to visualize the data in the plane. We have made this more clear in the new version.



3.2.2 ~~Classical~~ Dimensionality reduction with classical multidimensional scaling

To get an intuition for what the OPTICS algorithm does, and the differences to k-Means, we wish to visualize the data structure in the ~~clustering results of the OPTICS algorithm , we visualize the density structure of the trajectories in the 2 dimensional plane~~plane. For this, it is necessary to reduce the embedding dimension of each trajectory from $3T$ to two in a way that the density structure, and hence the individual Euclidean distances between embedded trajectories $d_{ij} = ||u_i - u_j||$, cf. eq. (3), are preserved. We do so by a common method of nonlinear dimensionality reduction, called classical ~~Multidimensional~~ multidimensional scaling (MDS), see e.g. chapter 10.3 of Fouss et al. (2016). Classical MDS tries to find an embedding of the high-dimensional data points in a low dimensional space such that the pairwise distances are approximately preserved.

~~Classical~~ Similar to a principal component analysis, classical MDS makes use of the eigenvectors corresponding to the largest eigenvalues of ~~the kernel matrix~~ a kernel matrix , which is in this case defined by

$$B = -\frac{1}{2} H \Delta^2 H, \tag{4}$$

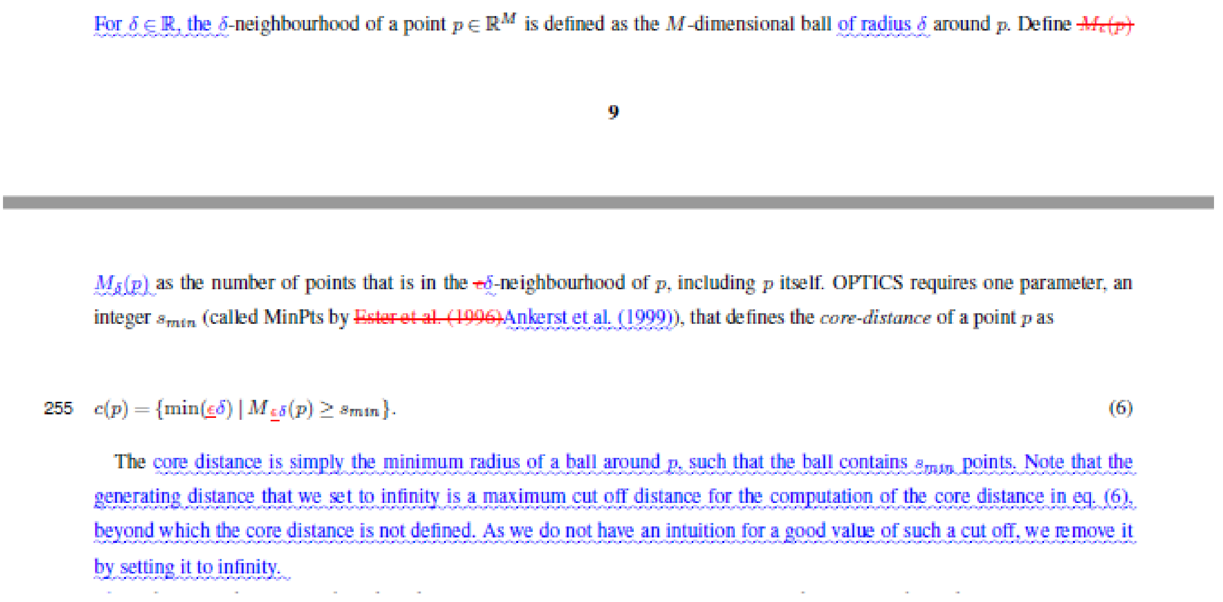
where $\Delta^2 \in \mathbb{R}^{N \times N}$ is a matrix containing all squared distances between the points, $\Delta^2_{ij} = ||u_i - u_j||^2$, and H is the centring matrix with $H_{ij} = \delta_{ij} - 1/N$, where δ_{ij} denotes the Kronecker delta. The matrix B in eq. (4) is called the centred inner product matrix. If $\bar{B}$ is the matrix of inner products of the embedded data points, i.e. $\bar{B}_{ij} = u_i \cdot u_j$ with Euclidean scalar product, then B can be obtained by removing the mean of all rows and columns of $\bar{B}$, cf. chapter 10.3 of Fouss et al. (2016). An embedding of the data points using the eigenvectors corresponding to the leading non-negative eigenvalues of B in eq. (4) ensures to capture the main variance of the (squared) distance structure, similar to a principal component analysis.
We compute Δ^2 with the Euclidean ~~embeddings~~ embedding described in section 3.2.1 and restrict ourselves to the first two dimensions to visualize the data structure in the plane, i.e. the embedding is defined by

$$u_i = (w_{0,i}, w_{1,i}),\ i = 1, \ldots, N, \tag{5}$$

where $K w_j = \lambda_j w_j$, and $\lambda_0 \geq \lambda_1 \geq \lambda_k$ for all $k = 2, \ldots N-1$. This choice of embedding ensures to capture the main variance of the data points, and we therefore also expect to capture the main structure in terms of data density. For large particle sets however, computing the spectrum of H in eq. (4) is computationally not feasible, as the matrix B is ~~in general~~ dense and computing the spectrum scales with $O(N^3)$. We apply classical MDS to the 12,000 particles of the Bickley jet model flow, and a random selection of the equal number of particles for the Agulhas flow. In our context, the method is most useful for visualization purposes, as it provides a good 2-dimensional approximation of the point distances, i.e. also the density structure of the embedded trajectories.

Comment 8
Line 193. The intuitive meaning of the ‘generating distances’ that are not being used here should be mentioned

Answer to comment 8
Than you for the comment. In the new version, we briefly mention what a finite generating distance would mean.



For $\delta \in \mathbb{R}$, the δ-neighbourhood of a point $p \in \mathbb{R}^M$ is defined as the M-dimensional ball of radius δ around p. Define ~~$M_\epsilon(p)$~~

9

$M_\delta(p)$ as the number of points that is in the ~~ϵ~~δ-neighbourhood of p, including p itself. OPTICS requires one parameter, an integer s_{min} (called MinPts by ~~Ester et al. (1996)~~Ankerst et al. (1999)), that defines the *core-distance* of a point p as

$$c(p) = \{\min(\epsilon\delta) \mid M_{\epsilon\delta}(p) \geq s_{min}\}. \tag{6}$$

The core distance is simply the minimum radius of a ball around p, such that the ball contains s_{min} points. Note that the generating distance that we set to infinity is a maximum cut off distance for the computation of the core distance in eq. (6), beyond which the core distance is not defined. As we do not have an intuition for a good value of such a cut off, we remove it by setting it to infinity.

Comment 9

Line 196. The definition of the epsilon neighborhood appears incomplete. Is it not the M-dimensional sphere of radius epsilon? Otherwise, what is the epsilon?

Answer to comment 9

Indeed the epsilon-neighborhood of p is just the M-dimensional ball around the point p, and the previous version was incomplete. We have changed this in the new version, together with renaming epsilon to delta, see our answer to comment 8.

Comment 10

Line 200. It would be very helpful to write out in words the meaning of Eq. (6). My understanding is that c(p) is minimum distance epsilon such that the number of points in an epsilon neighborhood is greater than a specified number.

Answer to comment 10

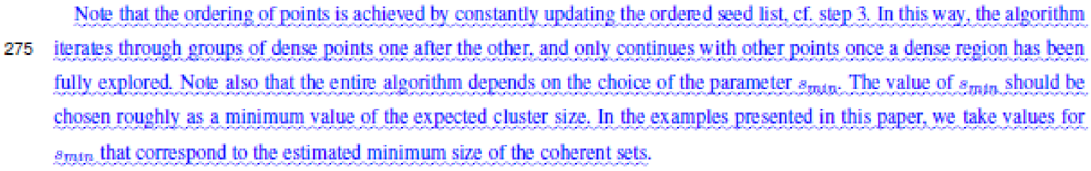
Thank you for your comment. Your interpretation was correct. We have made it more clear in the new version, see the answer to comment 8.

Comment 11

Line 213. I did not immediately understand how it arises that there are valleys in the reachability if you have sorted iteratively on the reachability. You might explain that this happens as you encounter groups of points that are all near to each other, thus replacing earlier high values of reachability with lower values.

Answer to comment 11

Thank you for the comment. Indeed, it is the sorting that is the most important step in the algorithm. We added some more explanation in the new version.



Note that the ordering of points is achieved by constantly updating the ordered seed list, cf. step 3. In this way, the algorithm iterates through groups of dense points one after the other, and only continues with other points once a dense region has been fully explored. Note also that the entire algorithm depends on the choice of the parameter s_{min}. The value of s_{min} should be chosen roughly as a minimum value of the expected cluster size. In the examples presented in this paper, we take values for s_{min} that correspond to the estimated minimum size of the coherent sets.

Comment 12

Line 216. The phrasing here made me wonder if this was a second, different epsilon. It would be clearer to say that you choose a value for the parameter epsilon. Also, it appears this is conditional on a choice of s_min which should then be emphasized.

Answer to comment 12

Thank you for very much for pointing this out. Indeed, this was a second epsilon, and the presentation in the first version was confusing. We have made the appropriate changes in the new version by re-naming one of the epsilons into delta. See our answer to comment 8.

Comment 13

Line 228. What are the permissible values of k in condition (a)?

Answer to comment 13

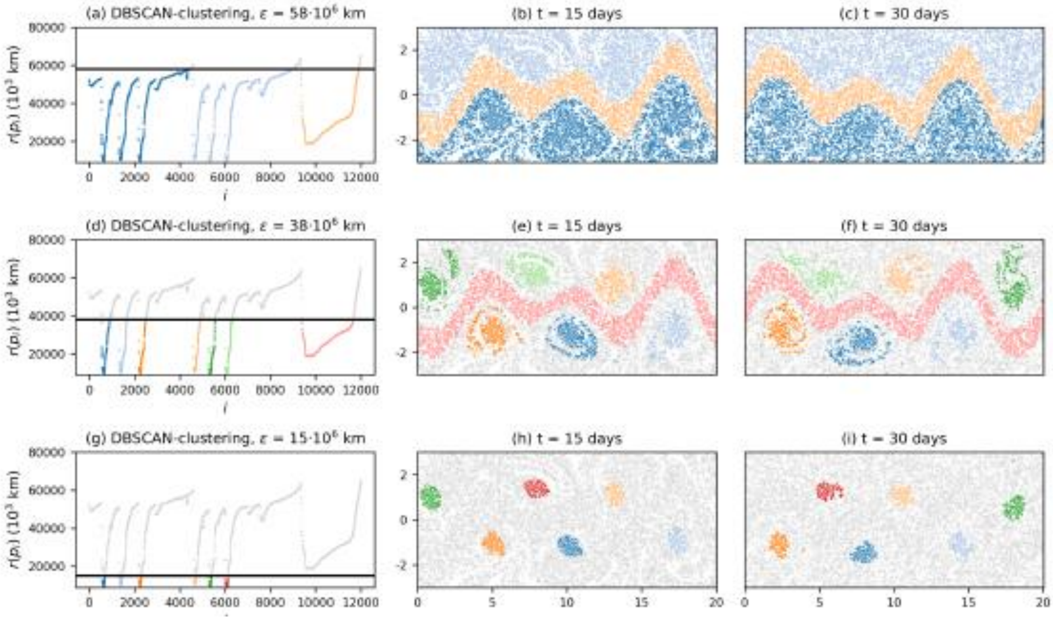
We have made this more precise in the new version. It can be any integer larger than zero and smaller than N - l.

Comment 14

Figure 2, what are the units of the y-axis in the left column of plots?

Answer to comment 14

Thank you for this comment. Indeed, we missed to specify the units of all reachability values. We do so in all figures in the new version (apart from the network embedding case in the appendix, where quantities are dimensionless), see the example below.

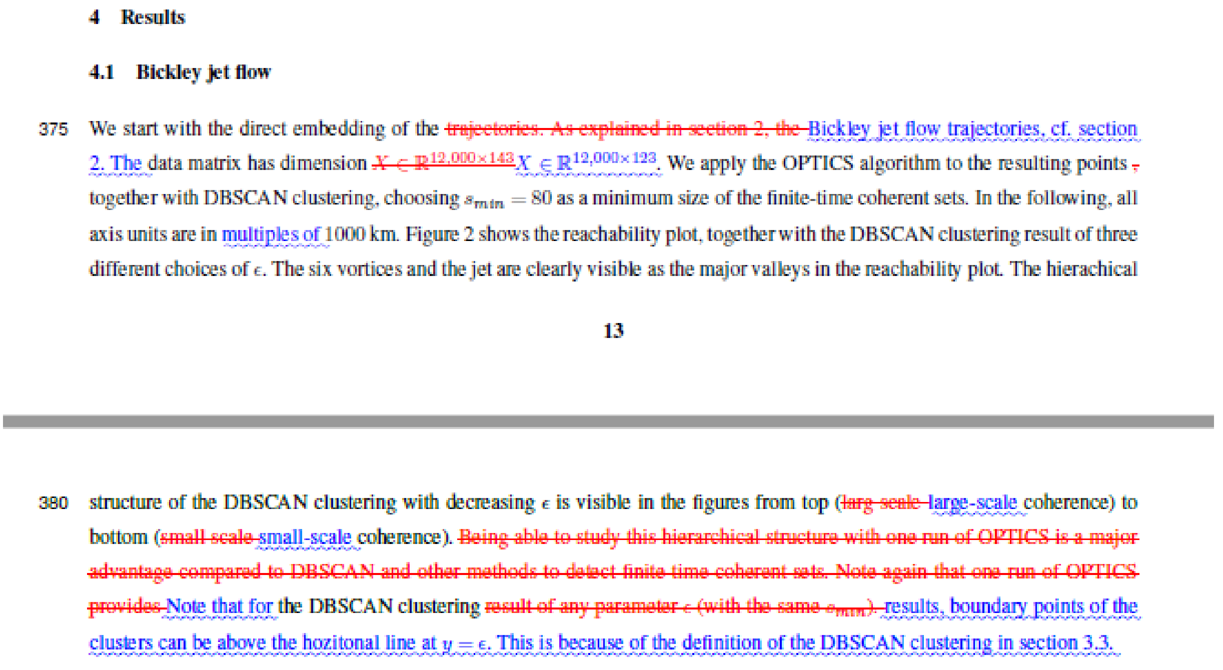


Comment 15

Figs 2 and 3, some of the colored dots lie above the epsilon threshold.

Answer to comment 15

This is correct, DBSCAN classifies the points below the line only up to boundary points, i.e. there can be points at the cluster boundary that belong to the cluster. We have made this more clear in the new version.



4 Results

4.1 Bickley jet flow

We start with the direct embedding of the ~~trajectories. As explained in section 2, the~~ Bickley jet flow trajectories, cf. section 2. The data matrix has dimension ~~$X \in \mathbb{R}^{12,000\times 143}$~~ $X \in \mathbb{R}^{12,000\times 123}$. We apply the OPTICS algorithm to the resulting points ~~,~~ together with DBSCAN clustering, choosing $s_{min} = 80$ as a minimum size of the finite-time coherent sets. In the following, all axis units are in multiples of 1000 km. Figure 2 shows the reachability plot, together with the DBSCAN clustering result of three different choices of ϵ. The six vortices and the jet are clearly visible as the major valleys in the reachability plot. The hierachical

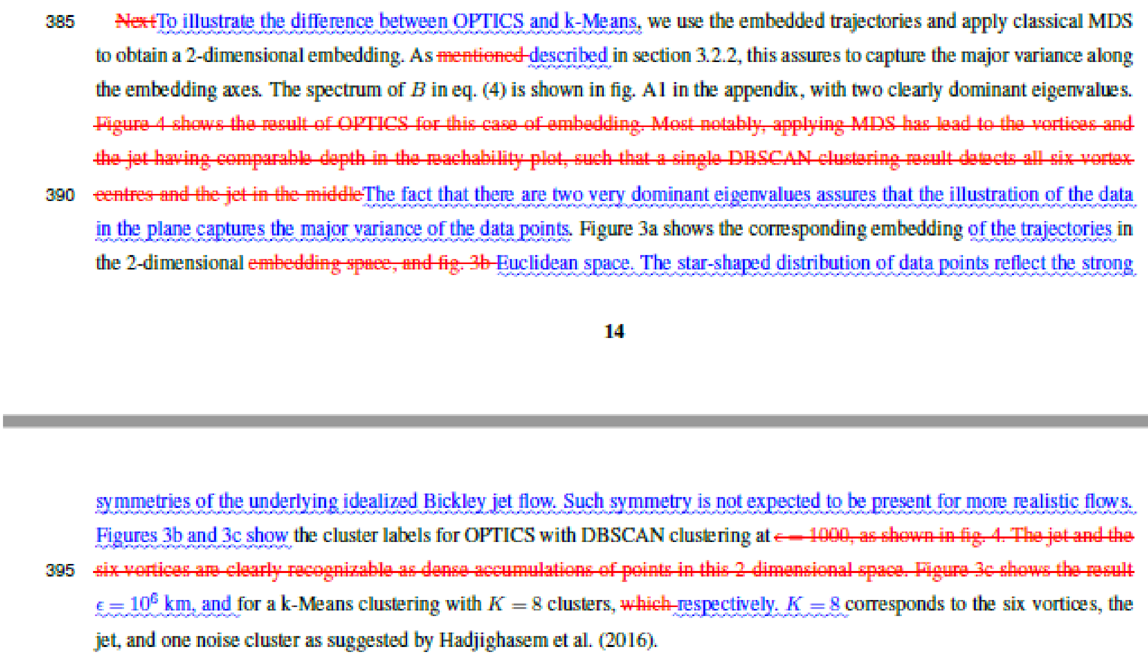
structure of the DBSCAN clustering with decreasing ϵ is visible in the figures from top (~~larg scale~~ large-scale coherence) to bottom (~~small scale~~ small-scale coherence). ~~Being able to study this hierarchical structure with one run of OPTICS is a major advantage compared to DBSCAN and other methods to detect finite time coherent sets. Note again that one run of OPTICS provides~~ Note that for the DBSCAN clustering ~~result of any parameter ϵ (with the same s_{min}).~~ results, boundary points of the clusters can be above the hozitonal line at $y = \epsilon$. This is because of the definition of the DBSCAN clustering in section 3.3.

Comment 16

Figure 4. I really don't understand the two dimensions of these plots, nor the star-shaped patterns, could you explain these more?

Answer to comment 16

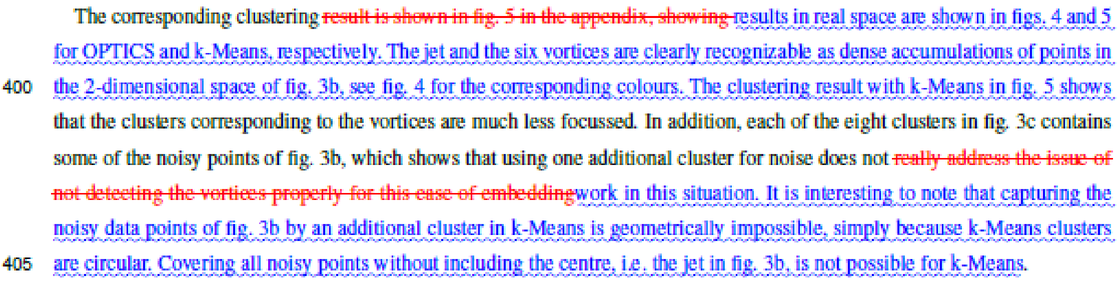
We have now made the presentation of the methods regarding classical MDS more clear, also relating it to principal component analysis, see the answer to your comment 7. In addition, we have provided more explanation on the star-shaped structure in the results section.



~~Next~~To illustrate the difference between OPTICS and k-Means, we use the embedded trajectories and apply classical MDS to obtain a 2-dimensional embedding. As ~~mentioned~~ described in section 3.2.2, this assures to capture the major variance along the embedding axes. The spectrum of B in eq. (4) is shown in fig. A1 in the appendix, with two clearly dominant eigenvalues. ~~Figure 4 shows the result of OPTICS for this case of embedding. Most notably, applying MDS has lead to the vortices and the jet having comparable depth in the reachability plot, such that a single DBSCAN clustering result detects all six vortex centres and the jet in the middle~~The fact that there are two very dominant eigenvalues assures that the illustration of the data in the plane captures the major variance of the data points. Figure 3a shows the corresponding embedding of the trajectories in the 2-dimensional ~~embedding space, and fig. 3b~~ Euclidean space. The star-shaped distribution of data points reflect the strong

symmetries of the underlying idealized Bickley jet flow. Such symmetry is not expected to be present for more realistic flows. Figures 3b and 3c show the cluster labels for OPTICS with DBSCAN clustering at ~~$\epsilon = 1000$, as shown in fig. 4. The jet and the six vortices are clearly recognizable as dense accumulations of points in this 2-dimensional space. Figure 3c shows the result~~ $\epsilon = 10^6$ km, and for a k-Means clustering with $K = 8$ clusters, ~~which~~ respectively. $K = 8$ corresponds to the six vortices, the jet, and one noise cluster as suggested by Hadjighasem et al. (2016).

In addition, we have further discussed the failure of k-Means in relation to the star-shaped structure of the embedding.



The corresponding clustering ~~result is shown in fig. 5 in the appendix, showing~~ results in real space are shown in figs. 4 and 5 for OPTICS and k-Means, respectively. The jet and the six vortices are clearly recognizable as dense accumulations of points in the 2-dimensional space of fig. 3b, see fig. 4 for the corresponding colours. The clustering result with k-Means in fig. 5 shows that the clusters corresponding to the vortices are much less focussed. In addition, each of the eight clusters in fig. 3c contains some of the noisy points of fig. 3b, which shows that using one additional cluster for noise does not ~~really address the issue of not detecting the vortices properly for this case of embedding~~work in this situation. It is interesting to note that capturing the noisy data points of fig. 3b by an additional cluster in k-Means is geometrically impossible, simply because k-Means clusters are circular. Covering all noisy points without including the centre, i.e. the jet in fig. 3b, is not possible for k-Means.

Comment 17

Data locations at Zenodo should be cited, not only the papers referring to them.

Answer to comment 17

The reference is actually a Zenodo link, not a paper. Note that there were two references Wichmann 2020 (Zenodo link) and Wichmann et al. (2020) (previous paper). In the new version, there is now also a Zenodo link to an animation for the Agulhas flow.

Comment 18
Throughout the paper, the authors consistently omit the subject ahead of an infinitive, e.g. “which allows to detect”. I believe this is grammatically incorrect (in US usage anyway). “allows one to detect” or “allowing the detection of” sound better

Answer to comment 18
Thank you, we have made appropriate changes in the new version.

Comment 19
l 42 and 90. “sparse” should probably be used instead of “scarce”. The former means thinly distributed while the latter means hard to come by.

Answer to comment 19
We have made appropriate changes in the new version.

Comment 20
l 128. NumPy and Zenodo are the standard capitalizations

Answer to comment 20
We made the suggested changes in the new version. Thank you for noting.

Comment 21
l 141. “method” should be “methods”

Answer to comment 21
Thank you for noting, we corrected it in the new version.

Comment 22
l 156. Straightforward

Answer to comment 22
Thank you for noting, we corrected it in the new version.

Comment 23
l 191. “and as will become clear”

Answer to comment 23
Thank you for noting, we corrected it in the new version.

Comment 24
l 217. “is equal to” should be “set of points is equivalent to”.

Answer to comment 24
Thank you for noting, we corrected it in the new version.

Comment 25
l 243. “a priory” should be “a priori”

Answer to comment 25
We corrected it in the new version.

Comment 26
l 279. “large- and small-scale”

Answer to comment 26
Thank you for noting, we corrected it in the new version.

Comment 27
l 354. GitHub

Answer to comment 27
Thank you for noting, we corrected it in the new version.

Comment 28

l 359. There is a title of an appendix with no appendix.

Answer to comment 28
The content of appendix C consisted of only two figures, C1 and C2. It appeared as without content due to the page break. In the new version, we have removed one appendix as we include the figures in the main text, such that the formatting looks better.

Comment 29
l 360 & 361. “particle-based”

Answer to comment 29

Thank you for noting, we corrected it in the new version.

Comment 30
l 383. “ot”

Answer to comment 30
Thanks for the careful read, we made the changes in the revised manuscript.

Comment 31
l 389. There should be a period at the end of this sentence

Answer to comment 31
Done. Thanks for noting.

Comment 32
Figure C1, “three” eigenvalues should be “two”, correct?

Answer to comment 32
Yes, indeed. Thanks for reading also the appendix figure captions so carefully! We corrected this in the new version.

Answer to reviewer 2

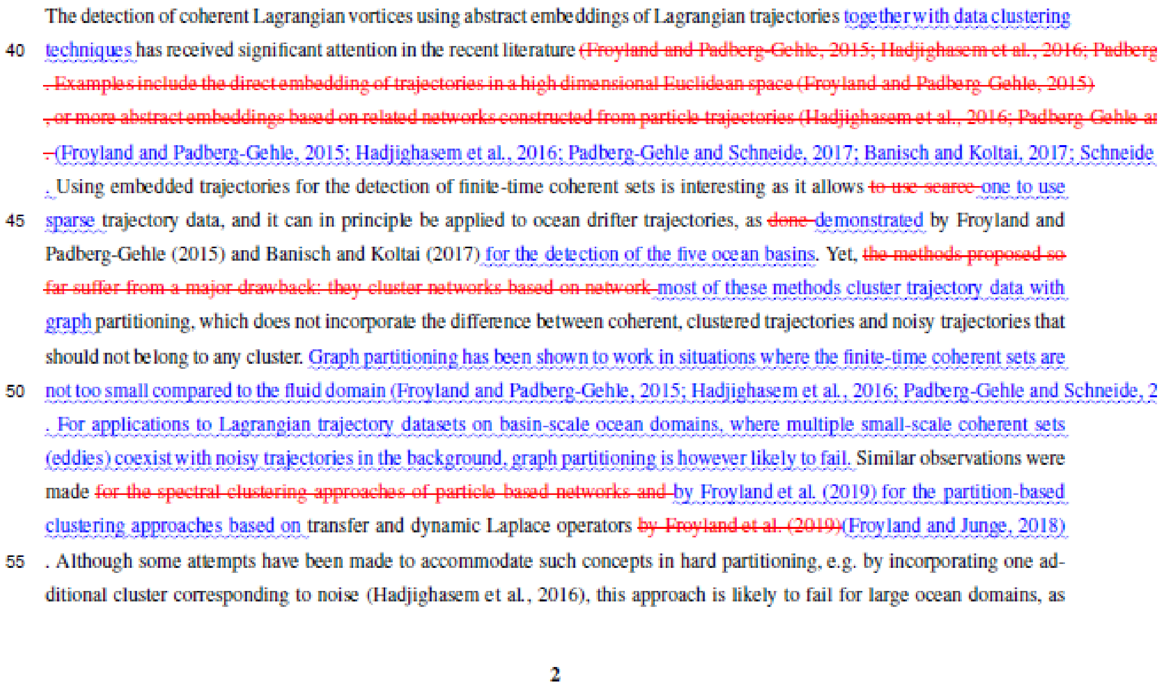
General answer:
We thank the reviewer for the critical comments, and in particular for the detailed analysis of other methods and their comparison to our approach. We agree with most points raised by the reviewer. We have made major adaptations to the formulations in the revised version, and explain the relation of our method to existing studies in more detail.

Please note:
The images in this file are excerpts of the revised version in latexdiff. Please apologize the formatting problems of latexdiff that cuts off references at line breaks. This is not the case in the revised version.

Comment 1
There are already several clustering methods in the literature for finding finite-time coherent sets, including a density-based clustering DBSCAN by Schneide-etal’18, which is a special case of the OPTICS approach in the manuscript. The idea of a hierarchy of finite-time coherent sets has been considered by Ma/Bollt’13. The paper Fr/Sa/Ro’19 develops a robust method to classify only those sets are that coherent, not fully partitioning the domain. In Fr/Sa/Ro’19, coherent sets at different spatial scales are also considered, similar to a hierarchy. Fr/Sa/Ro’19 also considers the Bickley jet and ocean eddies, with ocean eddies listed as a motivation in Fr/Sa/Ro’19 for developing a non-partitioning approach. Not limited to the work above, I would say there is some "upselling" of the novelty in the manuscript, and that prior work is occasionally omitted, mischaracterized, or overly criticized.

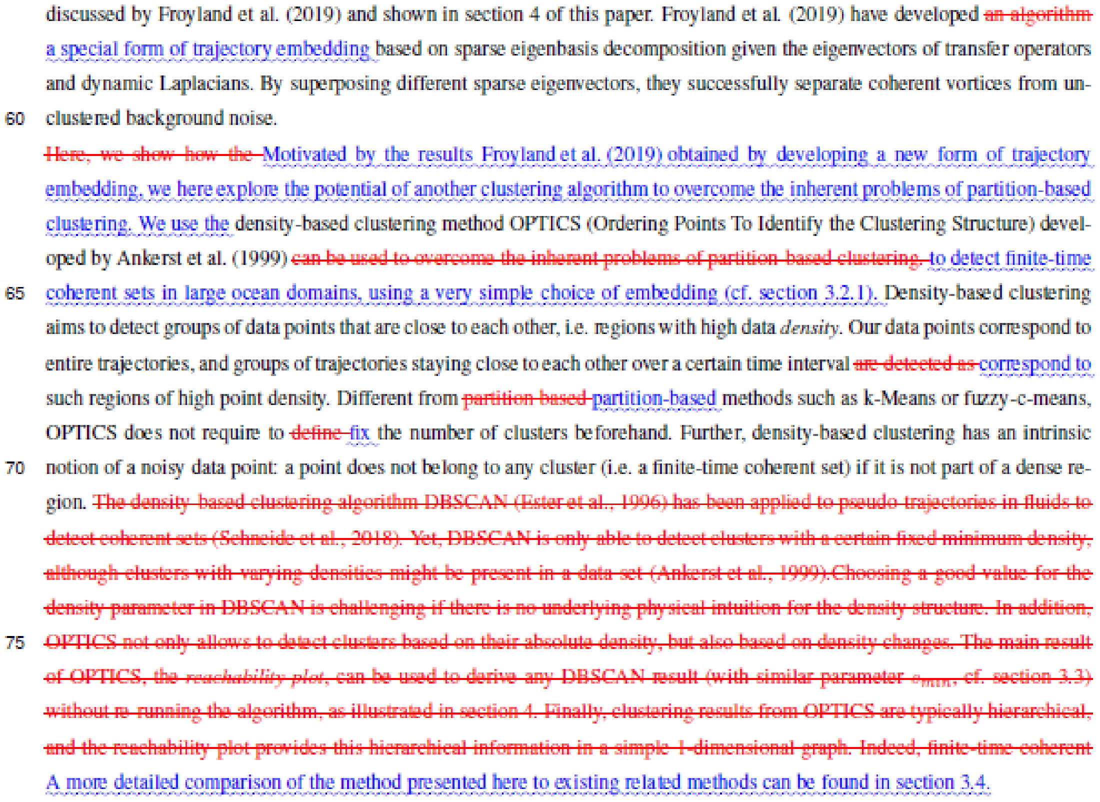
Answer to comment 1
Thank you for this comment. We did not intend to upsell our work, or omit, mischaracterize or overly criticize existing work. In fact, our work has been majorly motivated by the paper of Froyland et al. 2019. But we understand that the original manuscript appeared to do so, and we thank the reviewer to making this clear to us. We have made the following changes in the new version.

1. We mainly removed the discussion of other methods in the introduction and moved it to a separate section. In the introduction, we emphasize that our work is majorly inspired by Froyland et al. 2019. We are also more specific about the actual problem at hand, i.e. the detection of many small scale coherent sets in large-scale, noisy ocean flows.



The detection of coherent Lagrangian vortices using abstract embeddings of Lagrangian trajectories together with data clustering techniques has received significant attention in the recent literature ~~(Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg~~ ~~. Examples include the direct embedding of trajectories in a high dimensional Euclidean space (Froyland and Padberg-Gehle, 2015) , or more abstract embeddings based on related networks constructed from particle trajectories (Hadjighasem et al., 2016; Padberg-Gehle ar~~ ~~.~~ (Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2017; Banisch and Koltai, 2017; Schneide . Using embedded trajectories for the detection of finite-time coherent sets is interesting as it allows ~~to use scarce~~ one to use sparse trajectory data, and it can in principle be applied to ocean drifter trajectories, as ~~done~~ demonstrated by Froyland and Padberg-Gehle (2015) and Banisch and Koltai (2017) for the detection of the five ocean basins. Yet, ~~the methods proposed so far suffer from a major drawback: they cluster networks based on network~~ most of these methods cluster trajectory data with graph partitioning, which does not incorporate the difference between coherent, clustered trajectories and noisy trajectories that should not belong to any cluster. Graph partitioning has been shown to work in situations where the finite-time coherent sets are not too small compared to the fluid domain (Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2 . For applications to Lagrangian trajectory datasets on basin-scale ocean domains, where multiple small-scale coherent sets (eddies) coexist with noisy trajectories in the background, graph partitioning is however likely to fail. Similar observations were made ~~for the spectral clustering approaches of particle-based networks and~~ by Froyland et al. (2019) for the partition-based clustering approaches based on transfer and dynamic Laplace operators ~~by Froyland et al. (2019)~~(Froyland and Junge, 2018) . Although some attempts have been made to accommodate such concepts in hard partitioning, e.g. by incorporating one additional cluster corresponding to noise (Hadjighasem et al., 2016), this approach is likely to fail for large ocean domains, as

2

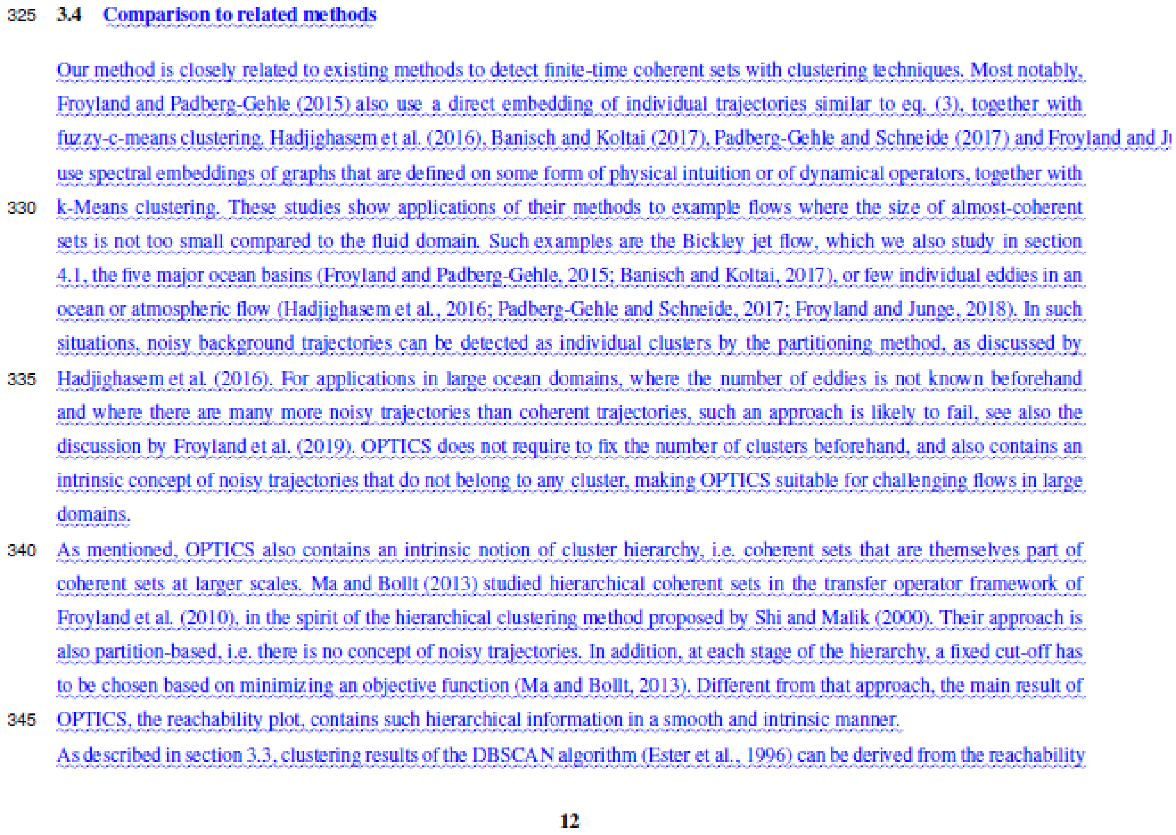


discussed by Froyland et al. (2019) and shown in section 4 of this paper. Froyland et al. (2019) have developed ~~an algorithm~~ a special form of trajectory embedding based on sparse eigenbasis decomposition given the eigenvectors of transfer operators and dynamic Laplacians. By superposing different sparse eigenvectors, they successfully separate coherent vortices from unclustered background noise.

~~Here, we show how the~~ Motivated by the results Froyland et al. (2019) obtained by developing a new form of trajectory embedding, we here explore the potential of another clustering algorithm to overcome the inherent problems of partition-based clustering. We use the density-based clustering method OPTICS (Ordering Points To Identify the Clustering Structure) developed by Ankerst et al. (1999) ~~can be used to overcome the inherent problems of partition-based clustering.~~ to detect finite-time coherent sets in large ocean domains, using a very simple choice of embedding (cf. section 3.2.1). Density-based clustering aims to detect groups of data points that are close to each other, i.e. regions with high data *density*. Our data points correspond to entire trajectories, and groups of trajectories staying close to each other over a certain time interval ~~are detected as~~ correspond to such regions of high point density. Different from ~~partition-based~~ partition-based methods such as k-Means or fuzzy-c-means, OPTICS does not require to ~~define~~ fix the number of clusters beforehand. Further, density-based clustering has an intrinsic notion of a noisy data point: a point does not belong to any cluster (i.e. a finite-time coherent set) if it is not part of a dense region. ~~The density-based clustering algorithm DBSCAN (Ester et al., 1996) has been applied to pseudo trajectories in fluids to detect coherent sets (Schneide et al., 2018). Yet, DBSCAN is only able to detect clusters with a certain fixed minimum density, although clusters with varying densities might be present in a data set (Ankerst et al., 1999).Choosing a good value for the density parameter in DBSCAN is challenging if there is no underlying physical intuition for the density structure. In addition, OPTICS not only allows to detect clusters based on their absolute density, but also based on density changes. The main result of OPTICS, the *reachability plot*, can be used to derive any DBSCAN result (with similar parameter ε_{min}, cf. section 3.3) without re-running the algorithm, as illustrated in section 4. Finally, clustering results from OPTICS are typically hierarchical, and the reachability plot provides this hierarchical information in a simple 1-dimensional graph. Indeed, finite-time coherent~~ A more detailed comparison of the method presented here to existing related methods can be found in section 3.4.

2. We have added an additional section to compare our method to existing approaches. There, we stress what our contribution is compared to Froyland et al. 2019: the study of an improved clustering step, instead of an improved embedding step. We also mention the downside of our method compared to Froyland and Junge (2018) and Froyland et al. (2019). Note that the hierarchical method of Ma and Bollt (2013) is powerful, but it is partition-

based and not intrinsic to the clustering algorithm, as there is a cut-off chosen at each step of the hierarchical clustering. Also, note that many of the existing methods that use k-Means did work for examples where the coherent sets are not very small compared to the fluid domain. Finally, DBSCAN has been used by Schneide et al. (2018), but not to derive explicit clustering results, and also not in the ocean context. We explain this in this new section.

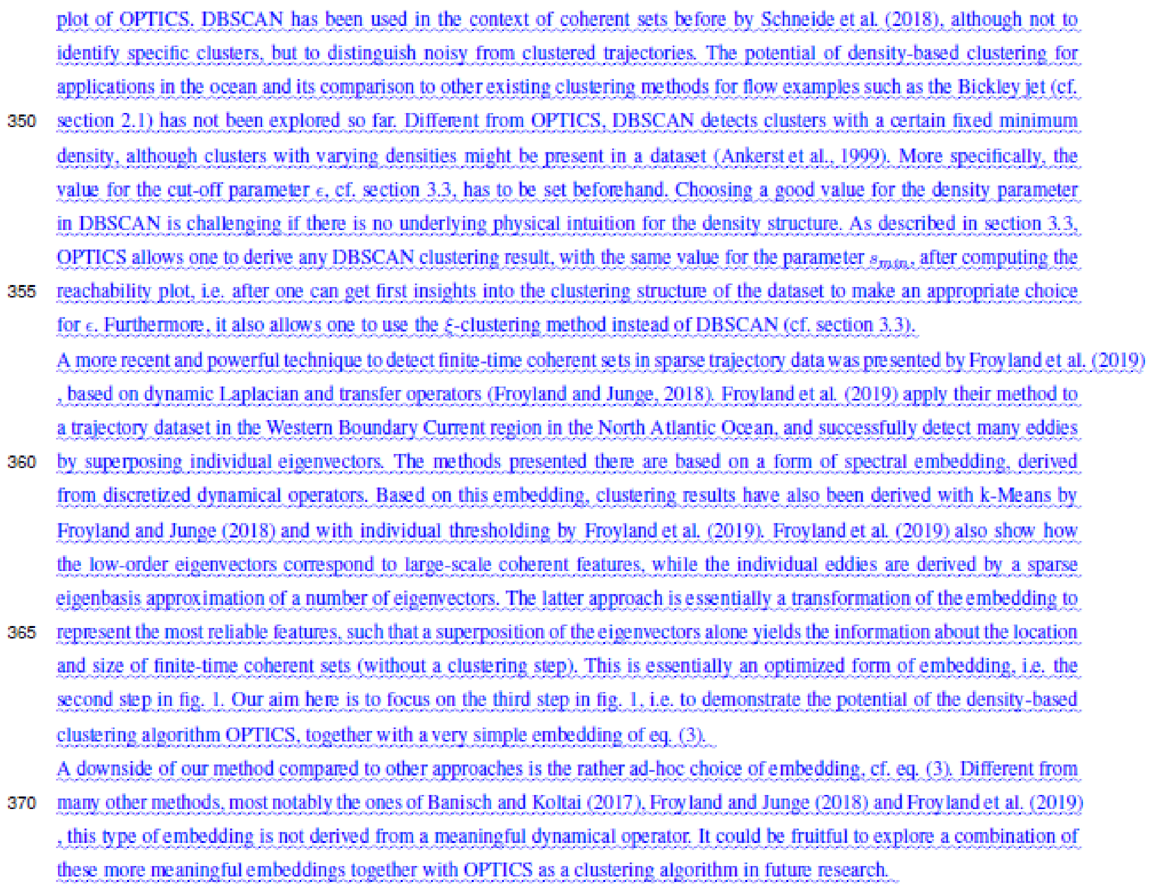


3.4 Comparison to related methods

Our method is closely related to existing methods to detect finite-time coherent sets with clustering techniques. Most notably, Froyland and Padberg-Gehle (2015) also use a direct embedding of individual trajectories similar to eq. (3), together with fuzzy-c-means clustering. Hadjighasem et al. (2016), Banisch and Koltai (2017), Padberg-Gehle and Schneide (2017) and Froyland and J
use spectral embeddings of graphs that are defined on some form of physical intuition or of dynamical operators, together with k-Means clustering. These studies show applications of their methods to example flows where the size of almost-coherent sets is not too small compared to the fluid domain. Such examples are the Bickley jet flow, which we also study in section 4.1, the five major ocean basins (Froyland and Padberg-Gehle, 2015; Banisch and Koltai, 2017), or few individual eddies in an ocean or atmospheric flow (Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2017; Froyland and Junge, 2018). In such situations, noisy background trajectories can be detected as individual clusters by the partitioning method, as discussed by Hadjighasem et al. (2016). For applications in large ocean domains, where the number of eddies is not known beforehand and where there are many more noisy trajectories than coherent trajectories, such an approach is likely to fail, see also the discussion by Froyland et al. (2019). OPTICS does not require to fix the number of clusters beforehand, and also contains an intrinsic concept of noisy trajectories that do not belong to any cluster, making OPTICS suitable for challenging flows in large domains.

As mentioned, OPTICS also contains an intrinsic notion of cluster hierarchy, i.e. coherent sets that are themselves part of coherent sets at larger scales. Ma and Bollt (2013) studied hierarchical coherent sets in the transfer operator framework of Froyland et al. (2010), in the spirit of the hierarchical clustering method proposed by Shi and Malik (2000). Their approach is also partition-based, i.e. there is no concept of noisy trajectories. In addition, at each stage of the hierarchy, a fixed cut-off has to be chosen based on minimizing an objective function (Ma and Bollt, 2013). Different from that approach, the main result of OPTICS, the reachability plot, contains such hierarchical information in a smooth and intrinsic manner.

As described in section 3.3, clustering results of the DBSCAN algorithm (Ester et al., 1996) can be derived from the reachability



plot of OPTICS. DBSCAN has been used in the context of coherent sets before by Schneide et al. (2018), although not to identify specific clusters, but to distinguish noisy from clustered trajectories. The potential of density-based clustering for applications in the ocean and its comparison to other existing clustering methods for flow examples such as the Bickley jet (cf. section 2.1) has not been explored so far. Different from OPTICS, DBSCAN detects clusters with a certain fixed minimum density, although clusters with varying densities might be present in a dataset (Ankerst et al., 1999). More specifically, the value for the cut-off parameter ϵ, cf. section 3.3, has to be set beforehand. Choosing a good value for the density parameter in DBSCAN is challenging if there is no underlying physical intuition for the density structure. As described in section 3.3, OPTICS allows one to derive any DBSCAN clustering result, with the same value for the parameter s_{min}, after computing the reachability plot, i.e. after one can get first insights into the clustering structure of the dataset to make an appropriate choice for ϵ. Furthermore, it also allows one to use the ξ-clustering method instead of DBSCAN (cf. section 3.3).

A more recent and powerful technique to detect finite-time coherent sets in sparse trajectory data was presented by Froyland et al. (2019) , based on dynamic Laplacian and transfer operators (Froyland and Junge, 2018). Froyland et al. (2019) apply their method to a trajectory dataset in the Western Boundary Current region in the North Atlantic Ocean, and successfully detect many eddies by superposing individual eigenvectors. The methods presented there are based on a form of spectral embedding, derived from discretized dynamical operators. Based on this embedding, clustering results have also been derived with k-Means by Froyland and Junge (2018) and with individual thresholding by Froyland et al. (2019). Froyland et al. (2019) also show how the low-order eigenvectors correspond to large-scale coherent features, while the individual eddies are derived by a sparse eigenbasis approximation of a number of eigenvectors. The latter approach is essentially a transformation of the embedding to represent the most reliable features, such that a superposition of the eigenvectors alone yields the information about the location and size of finite-time coherent sets (without a clustering step). This is essentially an optimized form of embedding, i.e. the second step in fig. 1. Our aim here is to focus on the third step in fig. 1, i.e. to demonstrate the potential of the density-based clustering algorithm OPTICS, together with a very simple embedding of eq. (3).

A downside of our method compared to other approaches is the rather ad-hoc choice of embedding, cf. eq. (3). Different from many other methods, most notably the ones of Banisch and Koltai (2017), Froyland and Junge (2018) and Froyland et al. (2019) , this type of embedding is not derived from a meaningful dynamical operator. It could be fruitful to explore a combination of these more meaningful embeddings together with OPTICS as a clustering algorithm in future research.

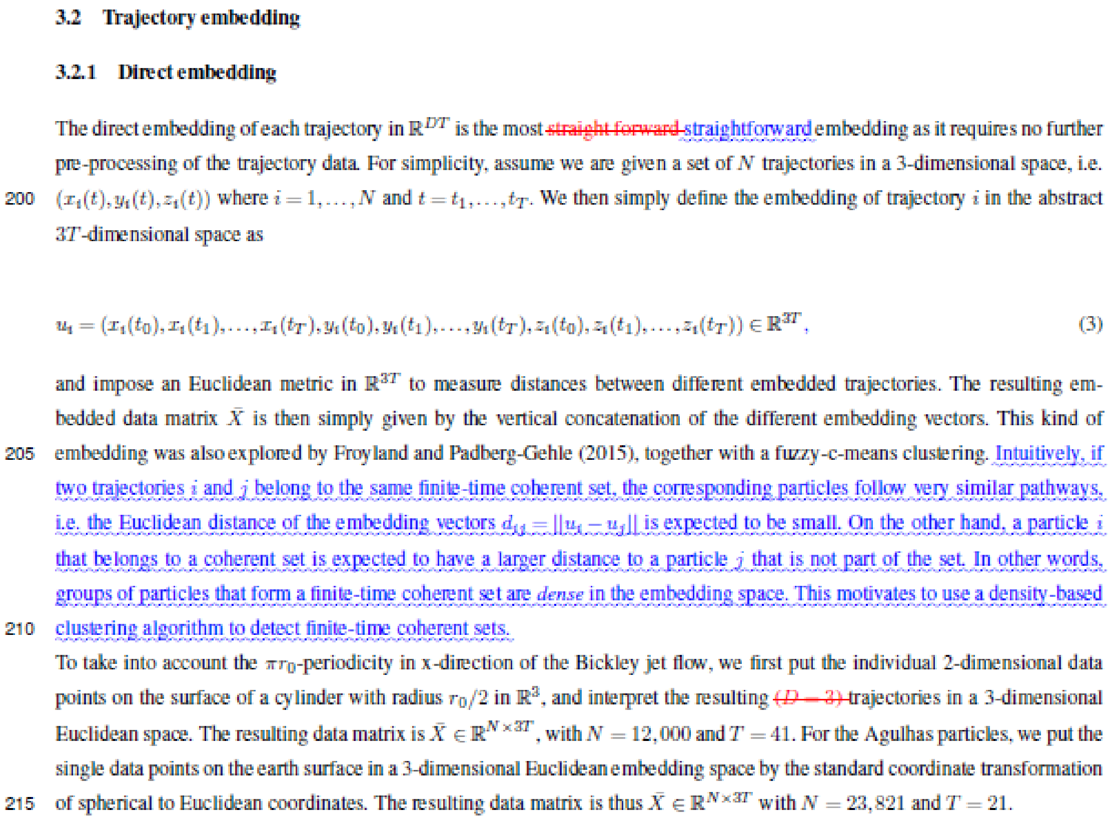
Comment 2

A positive aspect is that the (standard) "DBSCAN" and "\xi" clustering outputs of the OPTICS clustering could provide potentially useful hierarchical information, and to my knowledge this is a new way of analyzing the dynamics. Unfortunately, this is not explored much, and the authors do not provide an intuitive explanation of what the "DBSCAN" and "\xi" clustering algorithms are actually doing in their dynamical context. It would be beneficial for the authors to link the algorithms more with the dynamical inputs (trajectories) and the dynamical problem being solved. As this is the main contribution of the paper, I think this needs to be expanded much more. The reasons behind the choices of which clustering algorithm is applied to the different datasets should also be explained.

Answer to comment 2

Thank you for this comment. We were indeed lacking some form of intuition behind the two clustering methods and their application. We have made the following changes.

1. More explanation about the embedding and why the embedded trajectories create a signal in terms of data density.



3.2 Trajectory embedding

3.2.1 Direct embedding

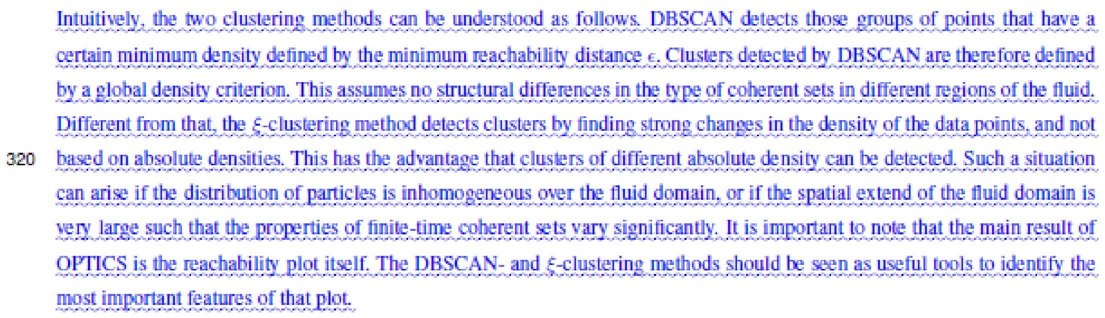
The direct embedding of each trajectory in $\mathbb{R}^{DT}$ is the most ~~straight forward~~ straightforward embedding as it requires no further pre-processing of the trajectory data. For simplicity, assume we are given a set of N trajectories in a 3-dimensional space, i.e. $(x_i(t), y_i(t), z_i(t))$ where $i = 1, \ldots, N$ and $t = t_1, \ldots, t_T$. We then simply define the embedding of trajectory i in the abstract $3T$-dimensional space as

$$u_i = (x_i(t_0), x_i(t_1), \ldots, x_i(t_T), y_i(t_0), y_i(t_1), \ldots, y_i(t_T), z_i(t_0), z_i(t_1), \ldots, z_i(t_T)) \in \mathbb{R}^{3T}, \tag{3}$$

and impose an Euclidean metric in $\mathbb{R}^{3T}$ to measure distances between different embedded trajectories. The resulting embedded data matrix $\bar{X}$ is then simply given by the vertical concatenation of the different embedding vectors. This kind of embedding was also explored by Froyland and Padberg-Gehle (2015), together with a fuzzy-c-means clustering. Intuitively, if two trajectories i and j belong to the same finite-time coherent set, the corresponding particles follow very similar pathways, i.e. the Euclidean distance of the embedding vectors $d_{ij} = ||u_i - u_j||$ is expected to be small. On the other hand, a particle i that belongs to a coherent set is expected to have a larger distance to a particle j that is not part of the set. In other words, groups of particles that form a finite-time coherent set are *dense* in the embedding space. This motivates to use a density-based clustering algorithm to detect finite-time coherent sets.

To take into account the πr_0-periodicity in x-direction of the Bickley jet flow, we first put the individual 2-dimensional data points on the surface of a cylinder with radius $r_0/2$ in $\mathbb{R}^3$, and interpret the resulting ~~$(D-3)$~~ trajectories in a 3-dimensional Euclidean space. The resulting data matrix is $\bar{X} \in \mathbb{R}^{N \times 3T}$, with $N = 12,000$ and $T = 41$. For the Agulhas particles, we put the single data points on the earth surface in a 3-dimensional Euclidean embedding space by the standard coordinate transformation of spherical to Euclidean coordinates. The resulting data matrix is thus $\bar{X} \in \mathbb{R}^{N \times 3T}$ with $N = 23,821$ and $T = 21$.

2. An intuitive explanation of the two clustering methods and their major properties.



Intuitively, the two clustering methods can be understood as follows. DBSCAN detects those groups of points that have a certain minimum density defined by the minimum reachability distance ϵ. Clusters detected by DBSCAN are therefore defined by a global density criterion. This assumes no structural differences in the type of coherent sets in different regions of the fluid. Different from that, the ξ-clustering method detects clusters by finding strong changes in the density of the data points, and not based on absolute densities. This has the advantage that clusters of different absolute density can be detected. Such a situation can arise if the distribution of particles is inhomogeneous over the fluid domain, or if the spatial extend of the fluid domain is very large such that the properties of finite-time coherent sets vary significantly. It is important to note that the main result of OPTICS is the reachability plot itself. The DBSCAN- and ξ-clustering methods should be seen as useful tools to identify the most important features of that plot.

3. We have included a DBSCAN clustering result in the main figure of the Agulhas flow example, and discuss the differences between xi and DBSCAN clustering.

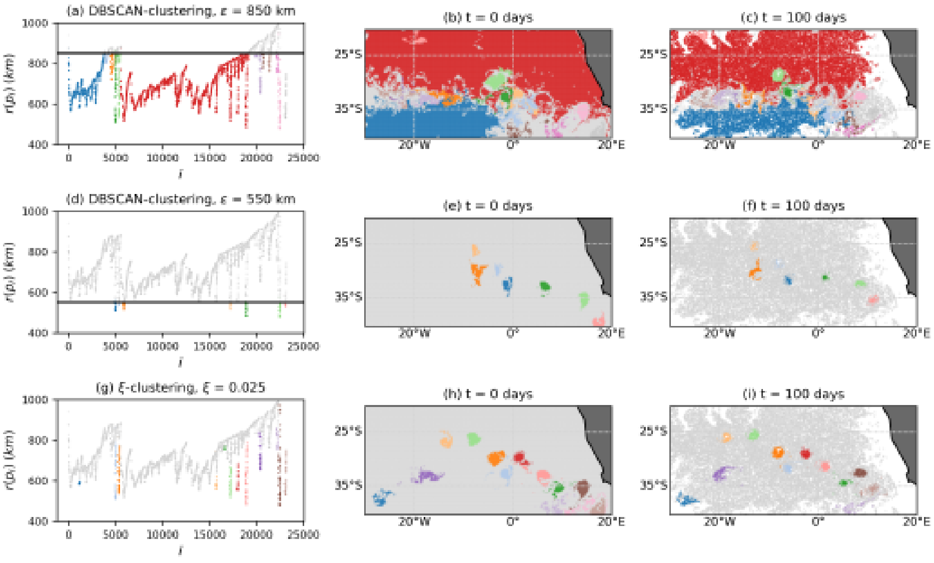


Figure 6. Result of the OPTICS algorithm applied to the direct embedding of the trajectories, with different clustering methods. Grey particles correspond to noise.

~~We~~ Figure 6 shows that for this situation, the ξ-clustering method detects more Agulhas rings than DBSCAN. While the clustering results shown in the figure all depends on the parameter values for ξ and ϵ, it is visible in the reachability plot of fig. 6g that the definition of some eddies includes the entire boundary of the valleys, i.e. up to very high reachability values. At the same time, the detection of the large-scale clusters as in 6a-c is not possible with the ξ-clustering method. These findings are in fact expected, cf. the discussion of the two clustering methods at the end of section 3.3. DBSCAN is best to detect global density structures, i.e. when the reachability values of all points are compared to the same cut-off ϵ. Regions that are dense locally but not necessarily globally are better detected with the ξ-clustering method. Despite these differences between the two clustering methods, we again emphasize that the main result of OPTICS is the reachability plot itself. Fig. 7 shows a colour

map at initial time of the reachability values. We clearly see Agulhas rings as the dark regions corresponding to lowest values of reachability. The regions of large reachability correspond to trajectories that are relatively noisy compared to all the other trajectories.

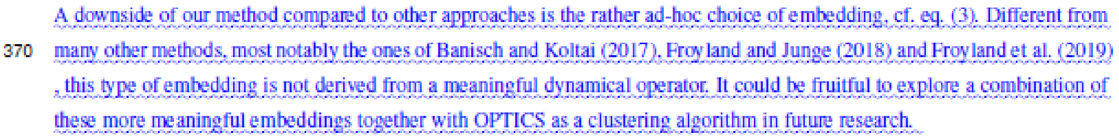
Comment 3

The (uncited) paper Froyland/Junge'18 develops a finite-element approximation of the dynamic Laplacian, which is a very accurate and robust method of finite-time coherent set extraction for low-dimensional systems of the type treated in the Wichmann manuscript. In Froyland/Junge'18 there are no free parameters, the method is unaffected by the density of the data points, and estimates are produced on the whole domain. A comparison can be made for the Bickley example in the Wichmann manuscript because the setup is identical. Wichmann et al uses a 200x60 grid of points and particle positions at times t=0, 1, 2, 3,..., 39, 40. Froyland/Junge'18 studied the same Bickley flow as in Wichmann, except that Froyland/Junge'18 used a coarser 100x30 grid of points and only particle positions at time 0 and time 40. Figure 15 in Froyland/Junge'18 shows much clearer images with fewer trajectory inputs. Thus, I think there is not a strong case for the approach in the manuscript being a better performer.

Answer to comment 3

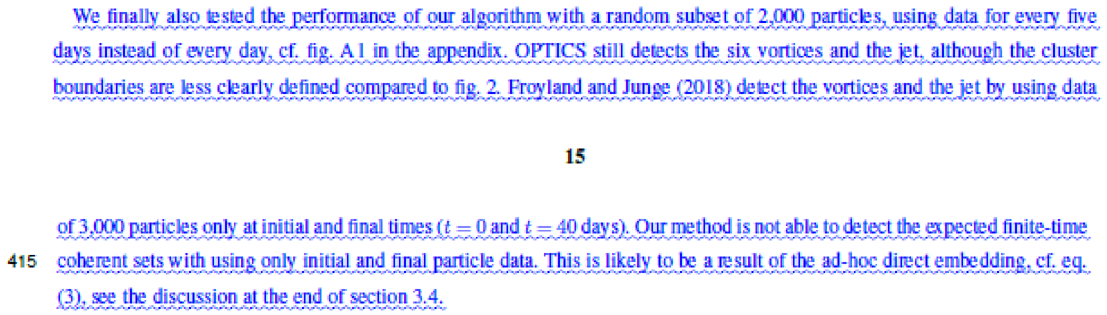
Thank you for this comment, and we apologize for not having cited that paper. Note however that the clustering results presented there are also based on k-Means clustering, and there are no free parameters only up to the choice of embedding dimension and the number of clusters. The paper also shows that the approach with k-Means works for situations where the coherent sets are not very small compared to the fluid domain, see the problems of k-Means in this context in the paper by Froyland et al. 2019. Nevertheless, the concepts presented there are powerful, as they provide a type of embedding that has a clear dynamical motivation, which is an advantage compared to our heuristic embedding. We refer to the paper at many places in the new version in different contexts:

1. End of the new section on comparison to other methods



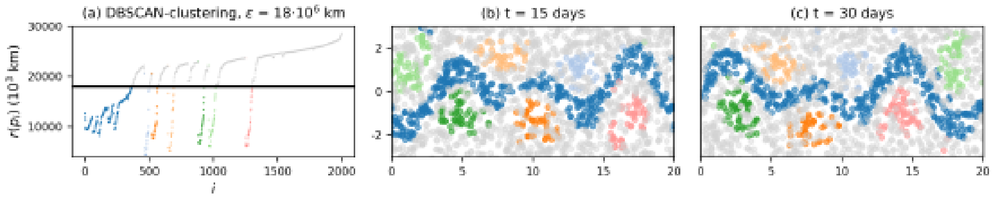
A downside of our method compared to other approaches is the rather ad-hoc choice of embedding, cf. eq. (3). Different from many other methods, most notably the ones of Banisch and Koltai (2017), Froyland and Junge (2018) and Froyland et al. (2019), this type of embedding is not derived from a meaningful dynamical operator. It could be fruitful to explore a combination of these more meaningful embeddings together with OPTICS as a clustering algorithm in future research.

2. We now also tested our method with the Bickley jet using less particles and less data points for each trajectory. Our method does indeed not perform as well as the method of Froyland and Junge (2018), and we want to thank the reviewer for explicitly mentioning this possible comparison.



We finally also tested the performance of our algorithm with a random subset of 2,000 particles, using data for every five days instead of every day, cf. fig. A1 in the appendix. OPTICS still detects the six vortices and the jet, although the cluster boundaries are less clearly defined compared to fig. 2. Froyland and Junge (2018) detect the vortices and the jet by using data

of 3,000 particles only at initial and final times ($t = 0$ and $t = 40$ days). Our method is not able to detect the expected finite-time coherent sets with using only initial and final particle data. This is likely to be a result of the ad-hoc direct embedding, cf. eq. (3), see the discussion at the end of section 3.4.



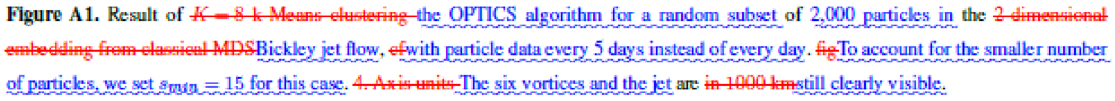
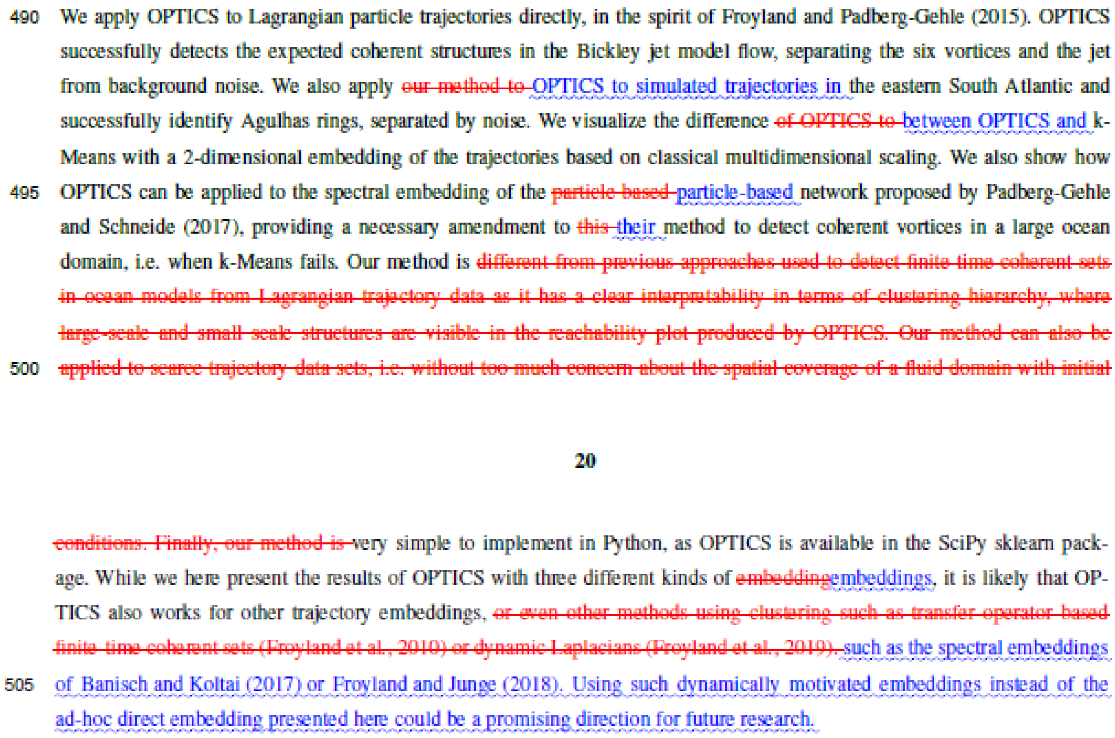


Figure A1. Result of ~~$K = 8$ k-Means clustering~~ the OPTICS algorithm for a random subset of 2,000 particles in the ~~2-dimensional embedding from classical MDS~~Bickley jet flow, ~~cf~~with particle data every 5 days instead of every day. ~~fig~~To account for the smaller number of particles, we set $s_{min} = 15$ for this case. ~~4. Axis units~~ The six vortices and the jet are ~~in 1000 km~~still clearly visible.

3. In the conclusion, we come back to the problems of our form of embedding and mention again that a combination of the embedding of Froyland and Junge (2018) together with OPTICS could yield better results.



We apply OPTICS to Lagrangian particle trajectories directly, in the spirit of Froyland and Padberg-Gehle (2015). OPTICS successfully detects the expected coherent structures in the Bickley jet model flow, separating the six vortices and the jet from background noise. We also apply ~~our method to~~ OPTICS to simulated trajectories in the eastern South Atlantic and successfully identify Agulhas rings, separated by noise. We visualize the difference ~~of OPTICS to~~ between OPTICS and k-Means with a 2-dimensional embedding of the trajectories based on classical multidimensional scaling. We also show how OPTICS can be applied to the spectral embedding of the ~~particle based~~ particle-based network proposed by Padberg-Gehle and Schneide (2017), providing a necessary amendment to ~~this~~ their method to detect coherent vortices in a large ocean domain, i.e. when k-Means fails. Our method is ~~different from previous approaches used to detect finite-time coherent sets in ocean models from Lagrangian trajectory data as it has a clear interpretability in terms of clustering hierarchy, where large-scale and small-scale structures are visible in the reachability plot produced by OPTICS. Our method can also be applied to scarce trajectory data sets, i.e. without too much concern about the spatial coverage of a fluid domain with initial conditions. Finally, our method is~~ very simple to implement in Python, as OPTICS is available in the SciPy sklearn package. While we here present the results of OPTICS with three different kinds of ~~embedding~~embeddings, it is likely that OPTICS also works for other trajectory embeddings, ~~or even other methods using clustering such as transfer operator based finite-time coherent sets (Froyland et al., 2010) or dynamic Laplacians (Froyland et al., 2019).~~ such as the spectral embeddings of Banisch and Koltai (2017) or Froyland and Junge (2018). Using such dynamically motivated embeddings instead of the ad-hoc direct embedding presented here could be a promising direction for future research.

Comment 4

The idea to not fully partition the domain has already been treated in Fr/Sa/Ro'19. Regarding the ocean eddy example in the manuscript, Fr/Sa/Ro'19 also applied the method of Froyland/Junge'18 to ocean flow and successfully extracted a greater number of eddies than Wichmann at a higher quality. On the other hand, Fr/Sa/Ro'19 used AVISO-derived trajectories rather than model output, so it could be that Wichmann is using a rougher velocity field. Wichmann also used lower trajectory density than Fr/Sa/Ro'19 by a factor of about 4; both of these items could make Wichmann's task more difficult, compared to Fr/Sa/Ro'19.

Answer to comment 4

Thank you for pointing this out. For a detailed comparison of the both methods, it would indeed be necessary to choose exactly the same flows. Detecting a greater number of eddies in a specific ocean domain does not necessarily have an implication for the usefulness of a method. We would like to note again that the results of Froyland et al. (2019) were a major motivation for our paper, and we do not aim to compete with their method any aspects. We would rather like to show how a change of clustering algorithm, instead of a change of embedding, can also yield better results compared to partition-based clustering, see the paragraph below in the revised paper on the comparison to other methods. We believe that a combination of the embedding of Froyland and Junge 2018 together with OPTICS could be a useful extension of our method. See our answer to your comments 1 and 3 for more content relating to their method.

Ordering of trajectories reveals hierarchical finite-time coherent sets in Lagrangian particle data: detecting Agulhas rings in the South Atlantic Ocean

David Wichmann[1,2], Christian Kehl[1], Henk A. Dijkstra[1,2], and Erik van Sebille[1,2]

[1]Institute for Marine and Atmospheric Research Utrecht, Utrecht University
[2]Centre for Complex Systems Studies, Utrecht University

Correspondence: David Wichmann (d.wichmann@uu.nl)

Abstract. The detection of finite-time coherent particle sets in Lagrangian trajectory data using data clustering techniques is an active research field at the moment. Yet, the clustering methods mostly employed so far have been based on graph partitioning, which assigns each trajectory to a cluster, i.e. there is no concept of noisy, incoherent trajectories. This is problematic for applications ~~to~~ in the ocean, where many small coherent eddies are present in a large~~fluid domain. In addition, to our knowledge none of the existing methods to detect finite-time coherent sets has an intrinsic notion of coherence hierarchy, i.e. the detection of finite-time coherent sets at different spatial scales. Such coherence hierarchies are present in the ocean, where basin scale coherence coexists with smaller coherent structures such as jets and mesoscale eddies.~~ , mostly noisy fluid flow. Here, for the first time in this context, we use the density-based clustering algorithm OPTICS (Ankerst et al., 1999) to detect finite-time coherent particle sets in Lagrangian trajectory data. Different from ~~partition based~~ partition-based clustering methods, ~~OPTICS does not require to fix the number of clusters beforehand. Derived~~ derived clustering results contain a concept of noise, such that not every trajectory needs to be part of a cluster. OPTICS also has a major advantage compared to the previously used DBSCAN method, as it can detect clusters of varying density. ~~Further, clusters can also be detected based on density changes instead of absolute density. Finally, OPTICS based clusters~~ The resulting clusters have an intrinsically hierarchical structure, which allows one to detect coherent trajectory sets at different spatial scales at once. We apply OPTICS directly to Lagrangian trajectory data in the Bickley jet model flow and successfully detect the expected vortices and the jet. The resulting clustering separates the vortices and the jet from background noise, with an imprint of the hierarchical clustering structure of coherent, ~~small scale~~ small-scale vortices in a coherent, large-scale, background flow. We then apply our method to a set of virtual trajectories released in the eastern South Atlantic Ocean in an eddying ocean model and successfully detect Agulhas rings. ~~At larger scale, our method also separates the eastward and westward moving parts of the subtropical gyre.~~ We illustrate the difference between our approach and ~~partition based~~ partition-based k-Means clustering using a 2-dimensional embedding of the trajectories derived from classical multidimensional scaling. We also show how OPTICS can be applied to the spectral embedding of a ~~trajectory based~~ trajectory-based network to overcome the problems of k-Means spectral clustering in detecting Agulhas rings.

Copyright statement. TEXT

1 Introduction

Understanding the transport of tracers in the ocean is an important topic in oceanography. Despite large-scale transport features of the mean flow, on smaller scales, mesoscale eddies and jets play an important role for tracer transport (Van Sebille et al., 2020). Such eddies can capture large amounts of a tracer, and, while transported in a background flow, redistribute them in the ocean. Eddies have been shown to play an important role for the accumulation of plastic (Brach et al., 2018) and the transport of heat and salt (Dong et al., 2014). To quantify the effects of eddies ~~for~~ on tracer transport in the ocean, it is necessary to develop methods that are able to detect and track them. Many methods exist to detect such *finite-time coherent sets* of fluid parcels based on different mathematical or heuristic principles (Hadjighasem et al., 2017). The term 'finite-time coherent set' is based on the work of Froyland et al. (2010), and is in our context defined as a set of particles that stay, in a sense to be made more specific, close to each other along their entire trajectories. ~~In this article, we propose a new way to identify finite-time coherent sets in Lagrangian trajectory data. For~~ Here, for the first time in this context, we make use of the density-based clustering algorithm OPTICS (Ankerst et al., 1999) ~~which allows to detect coherent trajectories at different spatial scales at once, introducing a powerful computational tool to the geophysical fluid dynamics community~~to detect finite-time coherent sets in Lagrangian trajectory data.

The detection of coherent Lagrangian vortices using abstract embeddings of Lagrangian trajectories together with data clustering techniques has received significant attention in the recent literature ~~(Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg-~~ ~~. Examples include the direct embedding of trajectories in a high dimensional Euclidean space (Froyland and Padberg-Gehle, 2015)~~ ~~, or more abstract embeddings based on related networks constructed from particle trajectories (Hadjighasem et al., 2016; Padberg-Gehle an~~ ~~.~~ (Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2017; Banisch and Koltai, 2017; Schneide . Using embedded trajectories for the detection of finite-time coherent sets is interesting as it allows ~~to use scarce~~ one to use sparse trajectory data, and it can in principle be applied to ocean drifter trajectories, as ~~done~~ demonstrated by Froyland and Padberg-Gehle (2015) and Banisch and Koltai (2017) for the detection of the five ocean basins. Yet, ~~the methods proposed so far suffer from a major drawback: they cluster networks based on network~~ most of these methods cluster trajectory data with graph partitioning, which does not incorporate the difference between coherent, clustered trajectories and noisy trajectories that should not belong to any cluster. Graph partitioning has been shown to work in situations where the finite-time coherent sets are not too small compared to the fluid domain (Froyland and Padberg-Gehle, 2015; Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2 . For applications to Lagrangian trajectory datasets on basin-scale ocean domains, where multiple small-scale coherent sets (eddies) coexist with noisy trajectories in the background, graph partitioning is however likely to fail. Similar observations were made ~~for the spectral clustering approaches of particle-based networks and~~ by Froyland et al. (2019) for the partition-based clustering approaches based on transfer and dynamic Laplace operators ~~by Froyland et al. (2019)~~(Froyland and Junge, 2018) . Although some attempts have been made to accommodate such concepts in hard partitioning, e.g. by incorporating one additional cluster corresponding to noise (Hadjighasem et al., 2016), this approach is likely to fail for large ocean domains, as discussed by Froyland et al. (2019) and shown in section 4 of this paper. Froyland et al. (2019) have developed ~~an algorithm~~ a special form of trajectory embedding based on sparse eigenbasis decomposition given the eigenvectors of transfer operators and dynamic Laplacians. By superposing different sparse eigenvectors, they successfully separate coherent vortices from unclustered background noise.

~~Here, we show how the~~ Motivated by the results Froyland et al. (2019) obtained by developing a new form of trajectory embedding, we here explore the potential of another clustering algorithm to overcome the inherent problems of partition-based clustering. We use the density-based clustering method OPTICS (Ordering Points To Identify the Clustering Structure) developed by Ankerst et al. (1999) ~~can be used to overcome the inherent problems of partition-based clustering.~~ to detect finite-time coherent sets in large ocean domains, using a very simple choice of embedding (cf. section 3.2.1). Density-based clustering aims to detect groups of data points that are close to each other, i.e. regions with high data *density*. Our data points correspond to entire trajectories, and groups of trajectories staying close to each other over a certain time interval ~~are detected as~~ correspond to such regions of high point density. Different from ~~partition based~~ partition-based methods such as k-Means or fuzzy-c-means, OPTICS does not require to ~~define~~ fix the number of clusters beforehand. Further, density-based clustering has an intrinsic notion of a noisy data point: a point does not belong to any cluster (i.e. a finite-time coherent set) if it is not part of a dense region. ~~The density-based clustering algorithm DBSCAN (Ester et al., 1996) has been applied to pseudo-trajectories in fluids to detect coherent sets (Schneide et al., 2018). Yet, DBSCAN is only able to detect clusters with a certain fixed minimum density, although clusters with varying densities might be present in a data set (Ankerst et al., 1999).Choosing a good value for the density parameter in DBSCAN is challenging if there is no underlying physical intuition for the density structure. In addition, OPTICS not only allows to detect clusters based on their absolute density, but also based on density changes. The main result of OPTICS, the *reachability plot*, can be used to derive any DBSCAN result (with similar parameter s_{min}, cf. section 3.3) without re-running the algorithm, as illustrated in section 4. Finally, clustering results from OPTICS are typically hierarchical, and the reachability plot provides this hierarchical information in a simple 1-dimensional graph. Indeed, finite-time coherent~~ A more detailed comparison of the method presented here to existing related methods can be found in section 3.4.

Another desirable property of the OPTICS algorithm is its ability to capture coherence hierarchies. In the ocean, coherent sets of trajectories naturally come with a notion of such a hierarchy. For example, the surface flow in the North Atlantic Ocean can be seen as approximately coherent (Froyland et al., 2014), while mesoscale eddies and jets are also finite-time coherent sets of trajectories at smaller scales *within* the North Atlantic Ocean. ~~This is also reflected in previous studies that apply methods to detect finite-time coherent sets to individual vortices and also to global drifter data, identifying the five major ocean basins (Froyland and Padberg-Gehle, 2015; Banisch and Koltai, 2017). The hierachical property of finite-time coherent sets has been studied in the transfer operator framework of Froyland et al. (2010) by Ma and Bollt (2013). Different from this approach, however, the clustering result derived from OPTICS is intrinsically hierarchical. This means that it shows in a smooth manner how the coherent structures change when zooming in or out, and it does not require to fix a certain partition to detect sub-partitions, e.g. as is typical for hierarchical applications of spectral clustering in the spirit of Shi and Malik (2000)~~ Froyland et al. (2019) show how their leading eigenvectors resolve coherent sets at large scales, while small-scale results can be obtained with a sparse eigenbasis approximation of a set of eigenvectors. Similarly, clustering results obtained from OPTICS are typically hierarchical. The main result of OPTICS, the reachability plot, provides this hierarchical information in a simple 1-dimensional graph.

In section 4, we first show how OPTICS detects finite-time coherent sets at different scales for the Bickley jet model flow (also discussed e.g. by Hadjighasem et al. (2017)), successfully detecting the six coherent vortices and the jet as the steepest valleys in the reachability plot. The general structure of the reachability plot also reveals the large-scale finite-time coherent sets, i.e. the northern and southern parts of the model flow, separated by the jet. We then apply our method to Lagrangian particle trajectories released in the eastern South Atlantic Ocean, where large rings detach from the Agulhas Current (e.g. Schouten et al. (2000)). We detect several Agulhas rings, and on the larger scale also separate the eastward and westward moving branches of the South Atlantic Subtropical Gyre. While the traditional approach to study Agulhas rings is based on sea surface height analysis (see e.g. Dencausse et al. (2010)), several methods based on virtual Lagrangian trajectories have been applied to Agulhas ring detection before (Haller and Beron-Vera, 2013; Beron-Vera et al., 2013; Froyland et al., 2015; Hadjighasem et al., 2016; Tarshish et al., 2018). Our method is different from these approaches in that it is directly applicable to a trajectory ~~data set~~dataset, i.e. without much pre-processing of the data. As the OPTICS algorithm is readily available in the sklearn package of SciPy, the detection of finite-time coherent sets can be done without much effort and with only a few lines of code. A further difference is the mentioned intrinsic notion of coherence hierarchy, which allows for simultaneous analysis of trajectory data at different scales. ~~Finally, trajectory based approaches can in principle be applied to scarce trajectory data, i.e. to any Lagrangian particle simulation result without much care for the spatial coverage of the initial conditions.~~ While we mainly focus on the direct embedding of trajectories in an abstract high-dimensional Euclidean space, we also show in ~~section C in the appendix~~ appendix C that OPTICS can be used to overcome the limits of k-Means clustering in the context of spectral clustering of ~~physically motivated trajectory based networks, such as the works presented by Hadjighasem et al. (2016), Padberg-Gehle and Schneide (2017)or Banisch and Koltai (2017)~~the trajectory-based network of Padberg-Gehle and Schneide (2017).

2 Trajectory datasets

2.1 Quasi-periodically perturbed Bickley jet

We apply our method to a model system that has been used frequently in studies to detect finite-time coherent sets ~~(Hadjighasem et al., 2017;~~ (Hadjighasem et al., 2017; Padberg-Gehle and Schneide, 2017; Hadjighasem et al., 2016; Banisch and Koltai, 2017; Froyland and Junge,
. The velocity field of the ~~Bickley jet~~ quasi-periodically perturbed Bickley jet (Bickley, 1937; del Castillo-Negrete and Morrison, 1993) is defined by a stream function $\psi(x,y,t)$, i.e. $\dot{x} = -\frac{\partial\psi}{\partial y}$ and $\dot{y} = \frac{\partial\psi}{\partial x}$, with $\psi(x,y,t) = \psi_0(y) + \psi_1(x,y,t)$ consisting of a stationary eastward background flow

$$\psi_0(y) = -UL\tanh(y/L), \tag{1}$$

and a time-dependent perturbation

$$\psi_1(x,y,t) = UL\,\mathrm{sech}^2(y/L)\ \mathrm{Re}\left[\sum_{n=1}^{3} f_n(t)\exp(ik_nx)\right], \tag{2}$$

where $\mathrm{Re}(z)$ denotes the real part of the complex number z. We use the same parameter values as Hadjighasem et al. (2017), with $U = 62.66$ m/s the characteristic velocity of the zonal background flow, and $L = 1770$ km. The parameters in eq. (2) are given by $k_n = 2n/r_0$, $f_n(t) = \epsilon_n \exp(-ik_nc_nt)$ with $\epsilon_1 = 0.075$, $\epsilon_2 = 0.4$, $\epsilon_3 = 0.3$, ~~$c_3 = 0.461U$~~ $c_1 = 0.1446U$, $c_2 = 0.205U$, ~~$c_1 = 0.1446U$~~ $c_3 = 0.461U$. The domain of interest is $\Omega = [0, \pi r_0] \times [-3000\ \mathrm{km}, 3000\ \mathrm{km}]$, where $r_0 = 6371$ km is the radius of the Earth, and the left and right edges of Ω are identified, i.e. the flow is periodic in x-direction with period πr_0. Similar to Banisch and Koltai (2017), we seed the domain with an initial number of 12,000 particles on a uniform 200×60 grid. For this choice, the initial particle spacing is slightly above 100 km in both directions. We compute the trajectories for 40 days with a time step of one second using the SciPy integrate package. We output the trajectories every day, i.e. we have $T = 41$ data points in time for each trajectory.

2.2 Agulhas rings in the South Atlantic

To test ~~our method~~ the OPTICS algorithm with a more realistic ocean flow, we simulate surface particle trajectories in a strongly eddying ocean model. Surface velocities are derived from a NEMO ORCA-N006 run (Madec, 2008), which has a horizontal resolution of $1/12°$ and velocity output for every five days. The model is forced by reanalysis and observed data of wind, heat and fresh water fluxes (Dussin et al., 2016), i.e. the currents do not only contain the geostrophic component, as is the case in altimetry-derived currents (Beron-Vera et al., 2013; Froyland et al., 2019). For the advection of virtual particles, we use version 1.11 of the open source Parcels framework (Lange and van Sebille, 2017), see oceanparcels.org. The 2-dimensional surface current velocity is interpolated in space and time with the C-grid interpolation scheme of Delandmeter and van Sebille (2019), using a 4th order Runge-Kutta method with a time step of 10 minutes. We initially distribute particles uniformly in the ocean on the vertices of a $0.2° \times 0.2°$ grid in the domain $[30°W, 20°E] \times [40°S, 20°S]$, which corresponds to a total number of 23,821 particles. At 30°S, a spacing of 0.2° corresponds to roughly 20 km. The particles start at January 5, 2000 and are advected for two years. We output the trajectories with a time interval of five days. We only use the first 100 days as data to detect the finite-time coherent sets, i.e. we have $T = 21$ data points for each trajectory, but also look at later times to see how long the rings need to disperse. ~~The data is a subset of the trajectory output of our previous paper (Wichmann et al., 2019), and we refer to that paper and the code references in there for the details of the particle simulation.~~ We provide the used trajectory data for the Agulhas flow as ~~numpy file on zenodo~~ NumPy file on Zenodo (Wichmann, 2020b).

3 Methods

3.1 Detecting coherent structures in Lagrangian trajectory data~~: an overview~~

For N trajectories of dimension D and length T, the trajectory information can be stored in a *data matrix* $X \in \mathbb{R}^{N \times DT}$, where each row results from a particle trajectory by concatenating the different spatial dimensions. The analysis of trajectory data to detect finite-time coherent sets of trajectories ~~(Froyland and Padberg-Gehle, 2015; Banisch and Koltai, 2017; Hadjighasem et al., 2016; Pad~~ (Froyland and Padberg-Gehle, 2015; Banisch and Koltai, 2017; Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2017; Schneide et can be split into two essential steps:

Step 1 **Embedding** of the trajectories in an abstract (metric) space, i.e. $X \to \bar{X} \in \mathbb{R}^{N \times M}$, where $M \leq DT$. If one uses a dimensionality reduction method, $M < DT$.

Step 2 **Clustering** of the embedded data with a clustering algorithm.

The embedding is necessary to represent the trajectories as points in a metric space. Different options for embedding the trajectories exist, e.g. a direct embedding of the data points along the trajectories (Froyland and Padberg-Gehle, 2015), or embeddings based on the eigenvectors derived from networks that are defined by physically motivated trajectory similarities (Banisch and Koltai, 2017; Padberg-Gehle and Schneide, 2017; Banisch and Koltai, 2017; Froyland and Junge, 2018). Once an embedding of each trajectory as a point in a metric (typically Euclidean) space is established, one can apply a clustering algorithm. Roughly speaking, clustering algorithms try to identify groups of points that are close to each other as a cluster. Partition-based clustering methods divide the entire data into a (typically fixed) number of K clusters, such that each data point belongs to a cluster. The most popular method in this category is the k-Means algorithm, which tries to find a given number of K clusters such that the sum of pairwise squared distances of points within a cluster is minimized. Other clustering algorithms contain a concept of ‘noisy’ data, i.e. data points that do not belong to any cluster, or belong to a cluster only with a certain probability. Examples for the former case are DBSCAN (Ester et al., 1996), discussed by Schneide et al. (2018) in the fluid dynamics context, and the here presented OPTICS (Ankerst et al., 1999) algorithm. For the latter case, the most popular method is fuzzy-c-means clustering, as discussed by Froyland and Padberg-Gehle (2015) in the context of finite-time coherent sets.

Figure 1 shows a few possible options for ~~these two steps~~ trajectory embedding and clustering that have partially been explored before (see the footnotes in the figure for the combinations used in related studies). For a given trajectory dataset, one can in principle apply an arbitrary combination of embedding and clustering ~~method~~methods. Only a few of the different combinations have been explored so far, and many more options for embedding and clustering as those shown in fig. 1 exist. It is important to note that a good choice of embedding and clustering might well depend on the specific problem at hand, and there might be no combination that performs well for all possible situations.

Most of the studies that use clustering techniques to detect finite-time coherent sets have focused on developing new forms of trajectory embeddings. For example, Hadjighasem et al. (2016), Padberg-Gehle and Schneide (2017), Banisch and Koltai (2017) and Froyland and Junge (2018) all use different forms of spectral embeddings, together with k-Means clustering. Froyland et al. (2019)

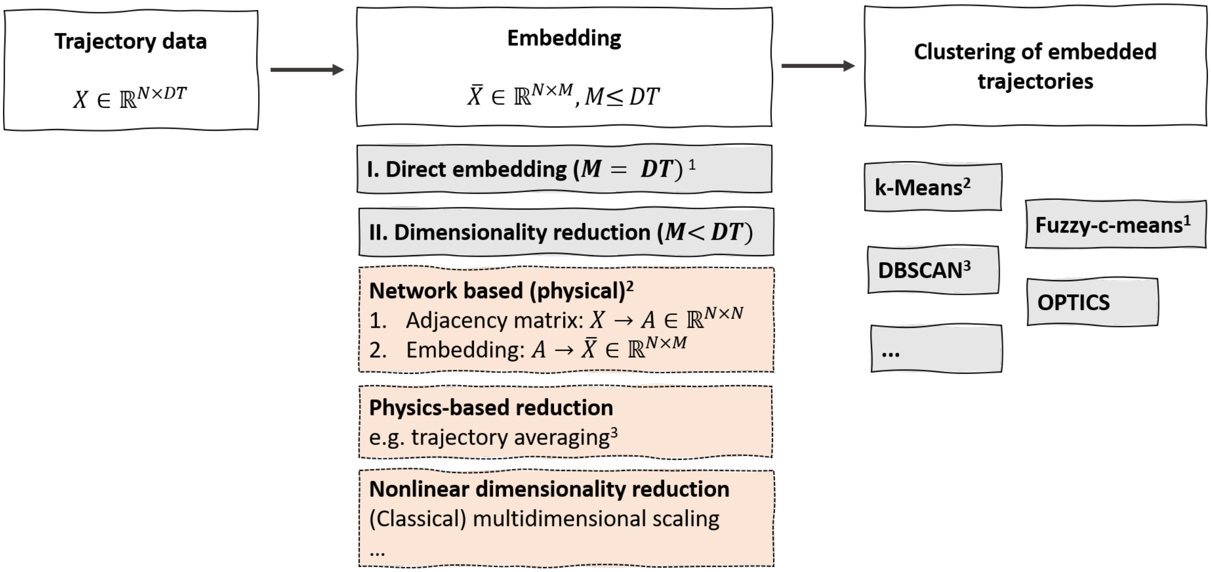

Trajectory data
$X \in \mathbb{R}^{N\times DT}$
Embedding
$\bar{X} \in \mathbb{R}^{N\times M}, M \leq DT$
Clustering of embedded trajectories
I. Direct embedding ($M = DT$) 1
II. Dimensionality reduction ($M < DT$)
Network based (physical)2
1. Adjacency matrix: $X \rightarrow A \in \mathbb{R}^{N\times N}$
2. Embedding: $A \rightarrow \bar{X} \in \mathbb{R}^{N\times M}$
Physics-based reduction
e.g. trajectory averaging3
Nonlinear dimensionality reduction
(Classical) multidimensional scaling
...
k-Means2
Fuzzy-c-means1
DBSCAN3
OPTICS
...

Figure 1. Different steps to detect coherent trajectories in Lagrangian data with trajectory clustering. The figure is non-exhaustive, and many more options for embedding and clustering exist. Footnotes: [1] Froyland and Padberg-Gehle (2015). [2] Hadjighasem et al. (2016), Padberg-Gehle and Schneide (2017) and Banisch and Koltai (2017) all define networks with spectral embedding and subsequent k-Means clustering. Froyland et al. (2019) define spectral embeddings defined on dynamic Laplacian and transfer operators. [3] Schneide et al. (2018).

have developed a powerful form of embedding, based on a sparse eigenbasis approximation. Here, we ~~explore the~~ focus on the clustering step in fig. 1, and propose the OPTICS clustering algorithm ~~for the first time in the context of finite-time coherent sets~~in the fluid dynamics context. We test ~~it~~ the algorithm for three different kinds of embeddings:

E1 A direct embedding of the trajectory data in a high dimensional Euclidean space, i.e. $M = DT$ (cf. section 3.2.1).

E2 A reduction of the trajectory data to a 2-dimensional embedding space using classical multidimensional scaling (MDS, cf. section 3.2.2). This is mainly to visualize the difference to ~~partition based~~ partition-based k-Means clustering.

E3 ~~An~~ A spectral embedding of the network proposed by Padberg-Gehle and Schneide (2017)~~, which is in section C in the appendix for the sake of brevity~~.

In the following sections, we explain in detail the embeddings E1 and E2 and the OPTICS algorithm. We introduce the network embedding E3 together with the corresponding results in ~~section C in the appendix~~ appendix C.

~~Different steps to detect coherent trajectories in Lagrangian data with trajectory clustering. The figure is non-exhaustive, and many more options for embedding and clustering exist. Footnotes: [1] Froyland and Padberg-Gehle (2015). [2] Hadjighasem et al. (2016), Padberg-Gehle and Schneide (2017) and Banisch and Koltai (2017) all define networks with spectral embedding and subsequent k-Means clustering. [3] Schneide et al. (2018).~~

3.2 Trajectory embedding

3.2.1 Direct embedding

The direct embedding of each trajectory in $\mathbb{R}^{DT}$ is the most ~~straight forward~~ straightforward embedding as it requires no further pre-processing of the trajectory data. For simplicity, assume we are given a set of N trajectories in a 3-dimensional space, i.e. $(x_i(t), y_i(t), z_i(t))$ where $i = 1, \ldots, N$ and $t = t_1, \ldots, t_T$. We then simply define the embedding of trajectory i in the abstract $3T$-dimensional space as

$$u_i = (x_i(t_0), x_i(t_1), \ldots, x_i(t_T), y_i(t_0), y_i(t_1), \ldots, y_i(t_T), z_i(t_0), z_i(t_1), \ldots, z_i(t_T)) \in \mathbb{R}^{3T}, \tag{3}$$

and impose an Euclidean metric in $\mathbb{R}^{3T}$ to measure distances between different embedded trajectories. The resulting embedded data matrix $\bar{X}$ is then simply given by the vertical concatenation of the different embedding vectors. This kind of embedding was also explored by Froyland and Padberg-Gehle (2015), together with a fuzzy-c-means clustering. Intuitively, if two trajectories i and j belong to the same finite-time coherent set, the corresponding particles follow very similar pathways, i.e. the Euclidean distance of the embedding vectors $d_{ij} = ||u_i - u_j||$ is expected to be small. On the other hand, a particle i that belongs to a coherent set is expected to have a larger distance to a particle j that is not part of the set. In other words, groups of particles that form a finite-time coherent set are *dense* in the embedding space. This motivates to use a density-based clustering algorithm to detect finite-time coherent sets.

To take into account the πr_0-periodicity in x-direction of the Bickley jet flow, we first put the individual 2-dimensional data points on the surface of a cylinder with radius $r_0/2$ in $\mathbb{R}^3$, and interpret the resulting ~~$(D = 3)$~~ trajectories in a 3-dimensional Euclidean space. The resulting data matrix is $\bar{X} \in \mathbb{R}^{N \times 3T}$, with $N = 12,000$ and $T = 41$. For the Agulhas particles, we put the single data points on the earth surface in a 3-dimensional Euclidean embedding space by the standard coordinate transformation of spherical to Euclidean coordinates. The resulting data matrix is thus $\bar{X} \in \mathbb{R}^{N \times 3T}$ with $N = 23,821$ and $T = 21$.

3.2.2 ~~Classical~~ Dimensionality reduction with classical multidimensional scaling

To get an intuition for what the OPTICS algorithm does, and the differences to k-Means, we wish to visualize the data structure in the ~~clustering results of the OPTICS algorithm , we visualize the density structure of the trajectories in the 2-dimensional plane~~plane. For this, it is necessary to reduce the embedding dimension of each trajectory from $3T$ to two in a way that the density structure, and hence the individual Euclidean distances between embedded trajectories $d_{ij} = ||u_i - u_j||$, cf. eq. (3), are preserved. We do so by a common method of nonlinear dimensionality reduction, called classical ~~Multidimensional~~ multidimensional scaling (MDS), see e.g. chapter 10.3 of Fouss et al. (2016). Classical MDS tries to find an embedding of the high-dimensional data points in a low dimensional space such that the pairwise distances are approximately preserved.

~~Classical~~ Similar to a principal component analysis, classical MDS makes use of the eigenvectors corresponding to the largest eigenvalues of ~~the kernel matrix~~a kernel matrix, which is in this case defined by

$$B = -\frac{1}{2} H \Delta^2 H, \tag{4}$$

where $\Delta^2 \in \mathbb{R}^{N \times N}$ is a matrix containing all squared distances between the points, $\Delta^2_{ij} = ||u_i - u_j||^2$, and H is the centring matrix with $H_{ij} = \delta_{ij} - 1/N$, where δ_{ij} denotes the Kronecker delta. The matrix B in eq. (4) is called the centred inner product matrix. If $\tilde{B}$ is the matrix of inner products of the embedded data points, i.e. $\tilde{B}_{ij} = u_i \cdot u_j$ with Euclidean scalar product, then B can be obtained by removing the mean of all rows and columns of $\tilde{B}$, cf. chapter 10.3 of Fouss et al. (2016). An embedding of the data points using the eigenvectors corresponding to the leading non-negative eigenvalues of B in eq. (4) ensures to capture the main variance of the (squared) distance structure, similar to a principal component analysis.
We compute Δ^2 with the Euclidean ~~embeddings~~ embedding described in section 3.2.1 and restrict ourselves to the first two dimensions to visualize the data structure in the plane, i.e. the embedding is defined by

$$u_i = (w_{0,i}, w_{1,i}),\ i = 1, \ldots, N, \tag{5}$$

where $K w_j = \lambda_j w_j$, and $\lambda_0 \geq \lambda_1 \geq \lambda_k$ for all $k = 2, \ldots N-1$. This choice of embedding ensures to capture the main variance of the data points, and we therefore also expect to capture the main structure in terms of data density. For large particle sets however, computing the spectrum of H in eq. (4) is computationally not feasible, as the matrix B is ~~in general~~ dense and computing the spectrum scales with $O(N^3)$. We apply classical MDS to the 12,000 particles of the Bickley jet model flow, and a random selection of the equal number of particles for the Agulhas flow. In our context, the method is most useful for visualization purposes, as it provides a good 2-dimensional approximation of the point distances, i.e. also the density structure of the embedded trajectories.

3.3 Clustering with OPTICS

The detection of dense accumulations of points that are separated from each other by non-dense regions (noise) is the main goal of density-based clustering. We use the OPTICS (Ordering Points To Identify the Clustering Structure) algorithm by Ankerst et al. (1999) to detect these regions. The OPTICS algorithm can be seen as an extension of DBSCAN (Ester et al., 1996)~~, with important advantages for the detection of finite-time coherent sets, as discussed in the introduction and will become clear in section 4.~~. As we have no prior information on the density structure of the embedded nodes, we set the 'generating distance' of OPTICS to infinity and our presentation here is limited to this case. The general OPTICS algorithm with finite generating distance is computationally more efficient and slightly more complicated, and we refer to Ankerst et al. (1999) for more details.~~For $\epsilon \in \mathbb{R}$, the ϵ~~

For $\delta \in \mathbb{R}$, the δ-neighbourhood of a point $p \in \mathbb{R}^M$ is defined as the M-dimensional ball of radius δ around p. Define ~~$M_\epsilon(p)$~~

$M_{\delta}(p)$ as the number of points that is in the ~~ϵ~~δ-neighbourhood of p, including p itself. OPTICS requires one parameter, an integer s_{min} (called MinPts by ~~Ester et al. (1996)~~Ankerst et al. (1999)), that defines the *core-distance* of a point p as

$$c(p) = \{\min(\underline{\epsilon}\delta) \mid M_{\underline{\epsilon}\delta}(p) \geq s_{min}\}. \tag{6}$$

The core distance is simply the minimum radius of a ball around p, such that the ball contains s_{min} points. Note that the generating distance that we set to infinity is a maximum cut off distance for the computation of the core distance in eq. (6), beyond which the core distance is not defined. As we do not have an intuition for a good value of such a cut off, we remove it by setting it to infinity.

The ordering of the points is based on the *reachability distance* of a point p w.r.t. another point q, defined as

$$r(p|q) = \max(c(q), ||p-q||), \tag{7}$$

where $||p-q||$ in our case denotes the Euclidean distance between p and q. The ordering of points is then constructed with the following scheme:

Step 1 Pick a point p_1. This is the first point in the order, and is arbitrary.

Step 2 Compute the core-distance $c(p_1)$ of p_1.

Step 3 Define an ordered seed list containing all other points, p_l, ~~$k = 2, \ldots, N$~~$l = 2, \ldots, N$. For each point p_l, define the reachability value $r(p_l)$ as the reachability distance (eq. (7)) w.r.t. p_1, $r(p_l) = r(p_l|p_1)$. Order the list in ascending order of the $r(p_l)$.

Step 4 Pick the first point on the ordered seed list as p_2 and compute the core-distance $c(p_2)$. For all remaining points p_l, $l = 3, \ldots, N$, update the reachability value $r(p_l) \to \min(r(p_l), r(p_l|p_2))$.

Step 5 Update the ordered seed list according to the new reachability.

~~Step 5~~

Step 6 Repeat steps 4-5 to obtain p_3. Continue until all points are processed.

Note that the ordering of points is achieved by constantly updating the ordered seed list, cf. step 3. In this way, the algorithm iterates through groups of dense points one after the other, and only continues with other points once a dense region has been fully explored. Note also that the entire algorithm depends on the choice of the parameter s_{min}. The value of s_{min} should be chosen roughly as a minimum value of the expected cluster size. In the examples presented in this paper, we take values for s_{min} that correspond to the estimated minimum size of the coherent sets.

The main result of the OPTICS algorithm is a *reachability plot*. This plot is the graph defined by $(i, r(p_i))$, where $p_0 = \infty$ by definition. The reachability plot is a powerful presentation of the global and local distribution of a set of points at once. The valleys in this plot correspond to dense regions, which we ~~will~~ relate to finite-time coherent sets. We ~~will~~ show examples of reachability plots in section 4. Given the reachability plot $(i, r(p_i))$, we use two common ways to derive a clustering result:

1. **DBSCAN clustering**: Choose a cut-off parameter ϵ and define all points p_i with $c(p_i) \leq \epsilon$ as core points. All points that are not in the ϵ-neighbourhood of a core point are defined as noise. This ~~is equal~~ set of noisy data points is equivalent to all points ~~p~~ p_i that are not core points and have a reachability value ~~$r(p)$ with $r(p) > \epsilon$~~$r(p_i)$ with $r(p_i) > \epsilon$. A cluster of size L is then defined as a consecutive set (in the sense of the ordering) of non-noise points $(p_j, p_{j+1}, \ldots, p_{j+L-1})$, with adjacent points p_{j-1} and p_{j+L} being noise. This is similar to the clustering result of a DBSCAN run with equal values for s_{min} and ϵ. All possible realizations of DBSCAN clusters~~(~~, with the same value for s_{min}~~)~~, can therefore be derived from the reachability values, core distances and the ordering determined by OPTICS. Up to boundary points, a DBSCAN clustering result can be obtained by drawing horizontal lines in the reachability plot, cf. section 4.

2. ξ**-clustering**: While the DBSCAN clustering method looks for deep valleys in the reachability plot, this method looks for valleys with steep boundaries. In short, the larger a parameter ξ with $0 < \xi < 1$, the steeper the boundary of a valley has to be to be classified as a cluster. In more detail, a ξ-cluster is defined as a consecutive set of points $(p_j, p_{j+1}, \ldots, p_{j+L-1})$ that has steep boundaries in the sense that for a parameter ξ, $0 < \xi < 1$:

   (a) The start of the cluster p_j is in a ξ-steep downward area. A ξ-steep downward area is a maximal set of consecutive points $(p_l, p_{l+1}, \ldots, p_{l+k})$, $k \in \{1, \ldots, N-l\}$ where: 1. p_l and p_{l+k} are ξ-steep downward points, i.e. $r(p_l) \leq (1-\xi) r(p_{l-1})$ and $r(p_{l+k}) \leq (1-\xi) r(p_{l+k-1})$, 2. $p_{l+i} \leq p_l$ for all $i = 1, \ldots, k$ and 3. not more than s_{min} consecutive points in the set are no ξ-steep downward points.

   (b) The end of the cluster p_{j+L-1} is a ξ-steep upward area. The definitions are the reverse of the ξ-steep downward area, with the definition of a ξ-steep upward point ~~is changed to~~ as $r(p_j) \leq (1-\xi) r(p_{j+1})$.

   (c) The cluster contains at least s_{min} points, i.e. $L \geq s_{min}$.

   (d) Every point in the inside of the cluster is at least a factor of $(1-\xi)$ smaller than the boundary points p_j and p_{j+L-1}. All points that do not belong to a cluster are classified as noise.

We refer to Ankerst et al. (1999) for a more detailed discussion of the ξ-clustering method with illustrations for example data. Note that the full ξ-clustering method presented by Ankerst et al. (1999) does contain some more details related to the choice of the start and end points, which we did not mention here.

~~Functions~~ The OPTICS algorithm as well as functions to derive both clustering results from an OPTICS output are ~~also~~ available in the SciPy sklearn package. Note that the implementation in sklearn allows for a minimum cluster size different from s_{min} ~~(item (c)~~ for the ξ-clustering method (item 2 c above), but we will not make use of this additional freedom to reduce the number of parameters. Note that, different from k-Means, both clustering methods do not require an a ~~priory~~ priori determination of the number of clusters. For the ξ-clustering method, a larger ξ requires steeper boundaries to form a cluster, i.e. will typically lead to a reduction of the number of resulting clusters. For DBSCAN clustering with very large ϵ, one will detect one large global cluster. Making ϵ smaller leads then to consecutive splits of this cluster, forming (up to noise) a cluster hierarchy. We will demonstrate the properties for both clustering methods in section 4 for different situations. In the following applications, we use an estimation of the minimum number of particles per finite-time coherent set for the parameter s_{min}. Intuitively, the two clustering methods can be understood as follows. DBSCAN detects those groups of points that have a certain minimum density defined by the minimum reachability distance ϵ. Clusters detected by DBSCAN are therefore defined by a global density criterion. This assumes no structural differences in the type of coherent sets in different regions of the fluid. Different from that, the ξ-clustering method detects clusters by finding strong changes in the density of the data points, and not based on absolute densities. This has the advantage that clusters of different absolute density can be detected. Such a situation can arise if the distribution of particles is inhomogeneous over the fluid domain, or if the spatial extend of the fluid domain is very large such that the properties of finite-time coherent sets vary significantly. It is important to note that the main result of OPTICS is the reachability plot itself. The DBSCAN- and ξ-clustering methods should be seen as useful tools to identify the most important features of that plot.

3.4 Comparison to related methods

Our method is closely related to existing methods to detect finite-time coherent sets with clustering techniques. Most notably, Froyland and Padberg-Gehle (2015) also use a direct embedding of individual trajectories similar to eq. (3), together with fuzzy-c-means clustering. Hadjighasem et al. (2016), Banisch and Koltai (2017), Padberg-Gehle and Schneide (2017) and Froyland and Ju use spectral embeddings of graphs that are defined on some form of physical intuition or of dynamical operators, together with k-Means clustering. These studies show applications of their methods to example flows where the size of almost-coherent sets is not too small compared to the fluid domain. Such examples are the Bickley jet flow, which we also study in section 4.1, the five major ocean basins (Froyland and Padberg-Gehle, 2015; Banisch and Koltai, 2017), or few individual eddies in an ocean or atmospheric flow (Hadjighasem et al., 2016; Padberg-Gehle and Schneide, 2017; Froyland and Junge, 2018). In such situations, noisy background trajectories can be detected as individual clusters by the partitioning method, as discussed by Hadjighasem et al. (2016). For applications in large ocean domains, where the number of eddies is not known beforehand and where there are many more noisy trajectories than coherent trajectories, such an approach is likely to fail, see also the discussion by Froyland et al. (2019). OPTICS does not require to fix the number of clusters beforehand, and also contains an intrinsic concept of noisy trajectories that do not belong to any cluster, making OPTICS suitable for challenging flows in large domains.

As mentioned, OPTICS also contains an intrinsic notion of cluster hierarchy, i.e. coherent sets that are themselves part of coherent sets at larger scales. Ma and Bollt (2013) studied hierarchical coherent sets in the transfer operator framework of Froyland et al. (2010), in the spirit of the hierarchical clustering method proposed by Shi and Malik (2000). Their approach is also partition-based, i.e. there is no concept of noisy trajectories. In addition, at each stage of the hierarchy, a fixed cut-off has to be chosen based on minimizing an objective function (Ma and Bollt, 2013). Different from that approach, the main result of OPTICS, the reachability plot, contains such hierarchical information in a smooth and intrinsic manner.

As described in section 3.3, clustering results of the DBSCAN algorithm (Ester et al., 1996) can be derived from the reachability

plot of OPTICS. DBSCAN has been used in the context of coherent sets before by Schneide et al. (2018), although not to identify specific clusters, but to distinguish noisy from clustered trajectories. The potential of density-based clustering for applications in the ocean and its comparison to other existing clustering methods for flow examples such as the Bickley jet (cf. section 2.1) has not been explored so far. Different from OPTICS, DBSCAN detects clusters with a certain fixed minimum density, although clusters with varying densities might be present in a dataset (Ankerst et al., 1999). More specifically, the value for the cut-off parameter ϵ, cf. section 3.3, has to be set beforehand. Choosing a good value for the density parameter in DBSCAN is challenging if there is no underlying physical intuition for the density structure. As described in section 3.3, OPTICS allows one to derive any DBSCAN clustering result, with the same value for the parameter s_{min}, after computing the reachability plot, i.e. after one can get first insights into the clustering structure of the dataset to make an appropriate choice for ϵ. Furthermore, it also allows one to use the ξ-clustering method instead of DBSCAN (cf. section 3.3).

A more recent and powerful technique to detect finite-time coherent sets in sparse trajectory data was presented by Froyland et al. (2019), based on dynamic Laplacian and transfer operators (Froyland and Junge, 2018). Froyland et al. (2019) apply their method to a trajectory dataset in the Western Boundary Current region in the North Atlantic Ocean, and successfully detect many eddies by superposing individual eigenvectors. The methods presented there are based on a form of spectral embedding, derived from discretized dynamical operators. Based on this embedding, clustering results have also been derived with k-Means by Froyland and Junge (2018) and with individual thresholding by Froyland et al. (2019). Froyland et al. (2019) also show how the low-order eigenvectors correspond to large-scale coherent features, while the individual eddies are derived by a sparse eigenbasis approximation of a number of eigenvectors. The latter approach is essentially a transformation of the embedding to represent the most reliable features, such that a superposition of the eigenvectors alone yields the information about the location and size of finite-time coherent sets (without a clustering step). This is essentially an optimized form of embedding, i.e. the second step in fig. 1. Our aim here is to focus on the third step in fig. 1, i.e. to demonstrate the potential of the density-based clustering algorithm OPTICS, together with a very simple embedding of eq. (3).

A downside of our method compared to other approaches is the rather ad-hoc choice of embedding, cf. eq. (3). Different from many other methods, most notably the ones of Banisch and Koltai (2017), Froyland and Junge (2018) and Froyland et al. (2019), this type of embedding is not derived from a meaningful dynamical operator. It could be fruitful to explore a combination of these more meaningful embeddings together with OPTICS as a clustering algorithm in future research.

4 Results

4.1 Bickley jet flow

We start with the direct embedding of the ~~trajectories. As explained in section 2, the~~ Bickley jet flow trajectories, cf. section 2. The data matrix has dimension ~~$X \in \mathbb{R}^{12,000\times 143}$~~ $X \in \mathbb{R}^{12,000\times 123}$. We apply the OPTICS algorithm to the resulting points ~~,~~ together with DBSCAN clustering, choosing $s_{min} = 80$ as a minimum size of the finite-time coherent sets. In the following, all axis units are in multiples of 1000 km. Figure 2 shows the reachability plot, together with the DBSCAN clustering result of three different choices of ϵ. The six vortices and the jet are clearly visible as the major valleys in the reachability plot. The hierachical structure of the DBSCAN clustering with decreasing ϵ is visible in the figures from top (~~larg scale~~ large-scale coherence) to bottom (~~small scale~~ small-scale coherence). ~~Being able to study this hierarchical structure with one run of OPTICS is a major advantage compared to DBSCAN and other methods to detect finite-time coherent sets. Note again that one run of OPTICS provides~~ Note that for the DBSCAN clustering ~~result of any parameter ϵ (with the same s_{min}).~~ results, boundary points of the clusters can be above the hozitonal line at $y = \epsilon$. This is because of the definition of the DBSCAN clustering in section 3.3.

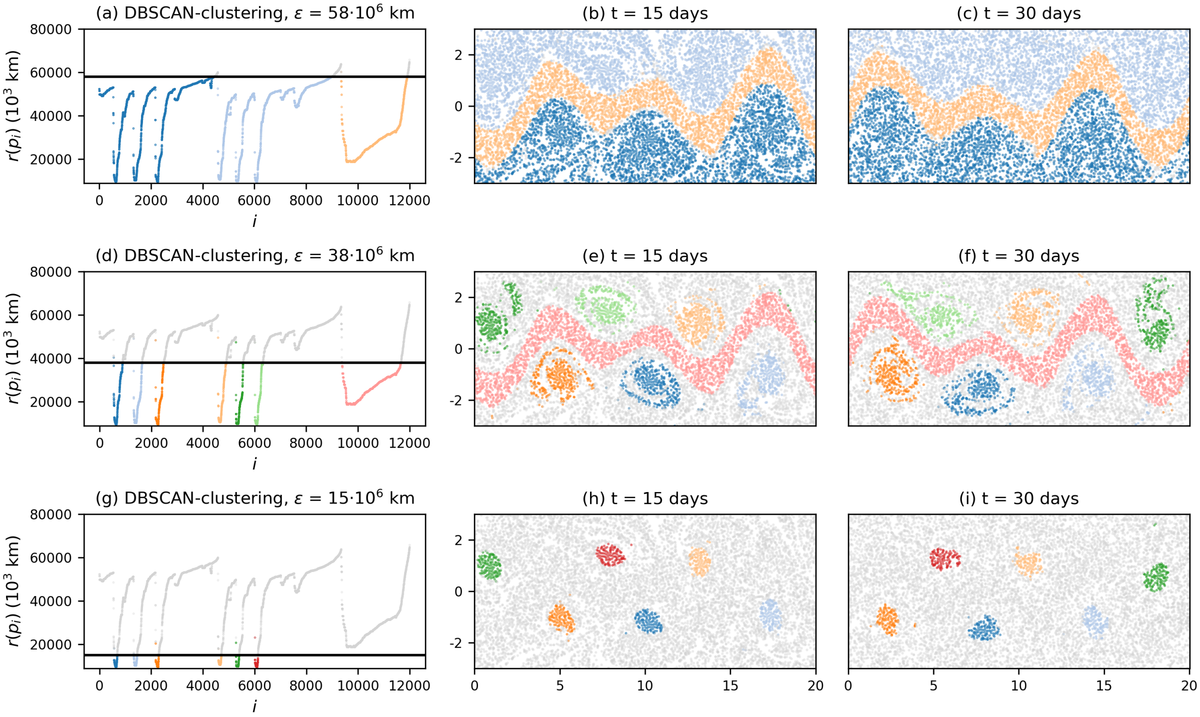


Figure 2. Result of the OPTICS algorithm applied to the direct embedding of the trajectories. (a), (d) and (g) show the reachability plot with different DBSCAN clustering results, indicated by the black horizontal line. The corresponding clustering results of each choice of DBSCAN parameter ϵ is shown on the right of the reachability plots for different times. Grey particles correspond to noise. Axis units in the centre and right column are in 1000 km.

~~Next~~To illustrate the difference between OPTICS and k-Means, we use the embedded trajectories and apply classical MDS to obtain a 2-dimensional embedding. As ~~mentioned~~ described in section 3.2.2, this assures to capture the major variance along the embedding axes. The spectrum of B in eq. (4) is shown in fig. A1 in the appendix, with two clearly dominant eigenvalues. ~~Figure 4 shows the result of OPTICS for this case of embedding. Most notably, applying MDS has lead to the vortices and the jet having comparable depth in the reachability plot, such that a single DBSCAN clustering result detects all six vortex centres and the jet in the middle~~The fact that there are two very dominant eigenvalues assures that the illustration of the data in the plane captures the major variance of the data points. Figure 3a shows the corresponding embedding of the trajectories in the 2-dimensional ~~embedding space, and fig. 3b~~ Euclidean space. The star-shaped distribution of data points reflect the strong symmetries of the underlying idealized Bickley jet flow. Such symmetry is not expected to be present for more realistic flows. Figures 3b and 3c show the cluster labels for OPTICS with DBSCAN clustering at ~~$\epsilon = 1000$, as shown in fig. 4. The jet and the six vortices are clearly recognizable as dense accumulations of points in this 2-dimensional space. Figure 3c shows the result~~ $\epsilon = 10^6$ km, and for a k-Means clustering with $K = 8$ clusters, ~~which~~ respectively. $K = 8$ corresponds to the six vortices, the jet, and one noise cluster as suggested by Hadjighasem et al. (2016).

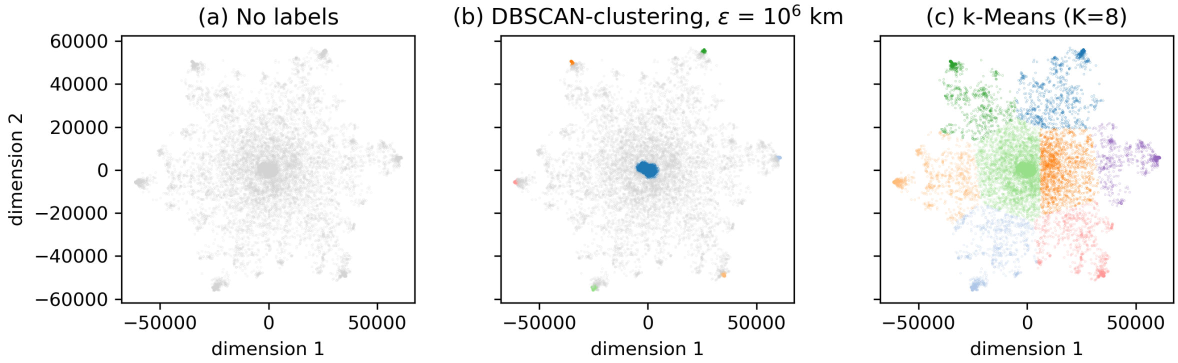


Figure 3. a: 2-dimensional embedding of the classical MDS method (cf. section 3.2.2) of the trajectories. b: with labels according to the DBSCAN result of fig. 4. The six vortices and the jet are clearly visible as dense regions. Grey particles correspond to noise. c: k-Means clustering result for K=8, see fig. 5 for the spatial clustering result of k-Means.

The corresponding clustering ~~result is shown in fig. 5 in the appendix, showing~~ results in real space are shown in figs. 4 and 5 for OPTICS and k-Means, respectively. The jet and the six vortices are clearly recognizable as dense accumulations of points in the 2-dimensional space of fig. 3b, see fig. 4 for the corresponding colours. The clustering result with k-Means in fig. 5 shows that the clusters corresponding to the vortices are much less focussed. In addition, each of the eight clusters in fig. 3c contains some of the noisy points of fig. 3b, which shows that using one additional cluster for noise does not ~~really address the issue of not detecting the vortices properly for this case of embedding~~work in this situation. It is interesting to note that capturing the noisy data points of fig. 3b by an additional cluster in k-Means is geometrically impossible, simply because k-Means clusters are circular. Covering all noisy points without including the centre, i.e. the jet in fig. 3b, is not possible for k-Means.

It should be noted here that the poor performance of k-Means in figs. 3c and 5 is not representative for other methods that use k-Means. For example, the method of Banisch and Koltai (2017) captures the coherent structures in the Bickley jet rather well, including the jet in the middle. We emphasize again that we use classical MDS here mostly for visualization purposes, as the computation of the classical MDS embedding is difficult for large particle sets. In our case, a dense $12,000 \times 12,000$ symmetric matrix has to be diagonalized, which already takes a significant amount of computation time.

We finally also tested the performance of our algorithm with a random subset of 2,000 particles, using data for every five days instead of every day, cf. fig. A1 in the appendix. OPTICS still detects the six vortices and the jet, although the cluster boundaries are less clearly defined compared to fig. 2. Froyland and Junge (2018) detect the vortices and the jet by using data

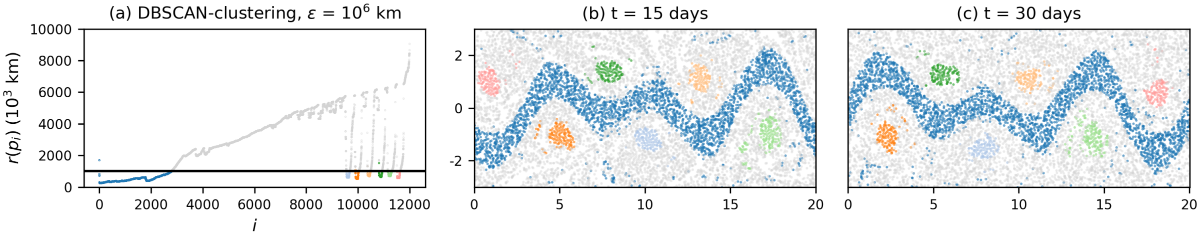


Figure 4. Result of DBSCAN clustering of the 2-dimensional embedding of the classical MDS method. a: reachability plot with black line representing the DBSCAN parameter ϵ. b-c: corresponding clustering results at different times. Grey particles represent noise. Axis units are in 1000 km.

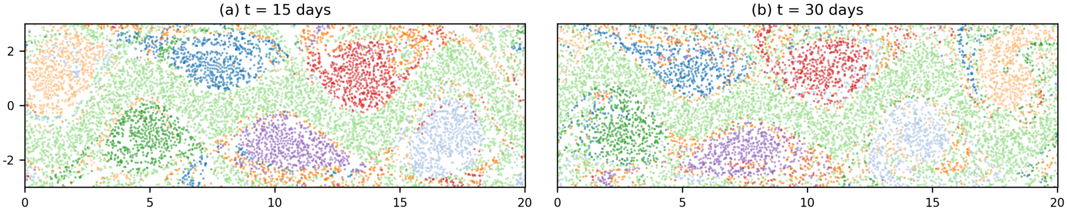


Figure 5. ~~a: 2-dimensional embedding~~ Result of $K = 8$ k-Means clustering of the 2-dimensional embedding from classical MDS~~method (~~, cf. ~~section 3.2.2) of the trajectories. b: with labels according to the DBSCAN result of~~ fig. 4. ~~The six vortices and the jet~~ Axis units are ~~clearly visible as dense regions~~in 1000 km.~~Grey particles correspond to noise. c: k-Means clustering result for K=8, see fig. 5 for the spatial clustering result of k-Means.~~

of 3,000 particles only at initial and final times ($t = 0$ and $t = 40$ days). Our method is not able to detect the expected finite-time coherent sets with using only initial and final particle data. This is likely to be a result of the ad-hoc direct embedding, cf. eq. (3), see the discussion at the end of section 3.4.

4.2 Agulhas rings

We next apply OPTICS to the Agulhas trajectories. As described in section 2, we have $\bar{X} \in \mathbb{R}^{N\times 63}$ with $N = 23,821$. We choose $s_{min} = 100$ in the following, which corresponds initially to a square cell of 2° × 2°, i.e. a reasonable minimum size of an Agulhas ring. Figure 6 shows the result of the direct embedding. The reachability plot in fig. 6a is much more jagged than for the Bickley jet model flow (cf. fig. 2a). The narrow deep valleys and the wider valleys in the reachability plot indicate the presence of ~~large and small scale~~ large- and small-scale coherence patterns. Figure 6a-c ~~shows~~ show the DBSCAN clustering result for a relatively large value of ϵ. The main separation of fluid domains is between the red and the blue particles, with a few vortices at their boundary. These two water masses are the northern and southern parts of the subtropical gyre in the South Atlantic, the red particles moving to the west, the blue particles to the east. The second and third rows of fig. 6 show ~~the~~ other clustering results for the DBSCAN- and the ξ-clustering method~~with different values of ξ~~, respectively. The valleys ~~with steep boundaries~~ in fig. 6g with steepest boundaries as detected by the ξ-clustering method mostly correspond to eddy-like structures, separated by background noise. ~~Our method also performs well when almost 80% of the trajectories is removed. Figure ?? shows a similar result for a reduced data set where we keep 5,000 randomly chosen particles and set $s_{min} = 20$ to account for the reduction in the number of particles. The large-scale structure as well as~~ Note that not all clusters in the figure correspond to eddies. For example, the blue cluster in figs. 6g-i stays approximately coherent over the considered time interval, although it is certainly not an Agulhas ring. An animation of the detected finite-time coherent sets for the full two years of trajectory data based on the ξ-clustering method as in the last row of fig. 6 can be found on Zenodo (Wichmann, 2020a), showing that many of the ~~eddies shown in fig. 6 are still visible.~~ sets stay coherent for significantly longer times than the first 100 days.

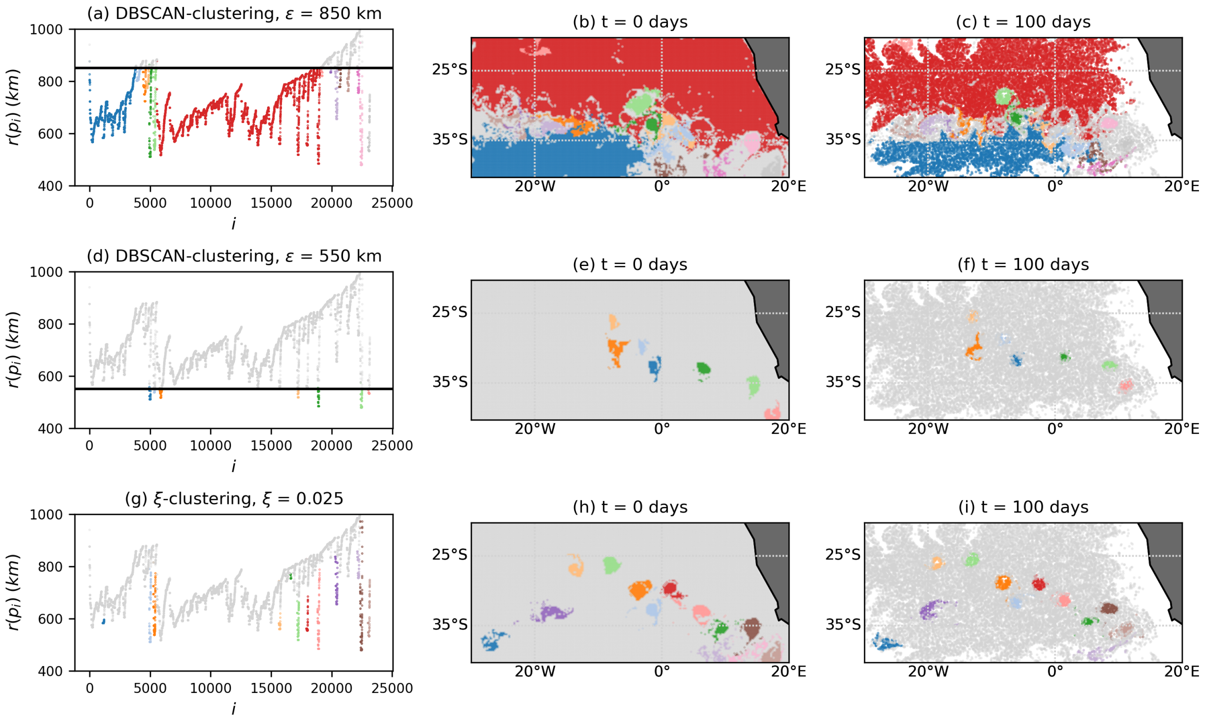


Figure 6. Result of the OPTICS algorithm applied to the direct embedding of the trajectories, with different clustering methods. Grey particles correspond to noise.

~~We~~ Figure 6 shows that for this situation, the ξ-clustering method detects more Agulhas rings than DBSCAN. While the clustering results shown in the figure all depends on the parameter values for ξ and ϵ, it is visible in the reachability plot of fig. 6g that the definition of some eddies includes the entire boundary of the valleys, i.e. up to very high reachability values. At the same time, the detection of the large-scale clusters as in 6a-c is not possible with the ξ-clustering method. These findings are in fact expected, cf. the discussion of the two clustering methods at the end of section 3.3. DBSCAN is best to detect global density structures, i.e. when the reachability values of all points are compared to the same cut-off ϵ. Regions that are dense locally but not necessarily globally are better detected with the ξ-clustering method. Despite these differences between the two clustering methods, we again emphasize that the main result of OPTICS is the reachability plot itself. Fig. 7 shows a colour map at initial time of the reachability values. We clearly see Agulhas rings as the dark regions corresponding to lowest values of reachability. The regions of large reachability correspond to trajectories that are relatively noisy compared to all the other trajectories.

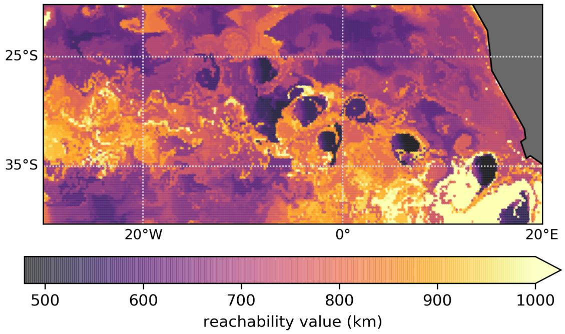


Figure 7. Reachability values at initial time, resulting from the OPTICS algorithm applied to the direct embedding of the trajectories. The regions with lowest values clearly correspond to Agulhas rings. The colour bar is cut off at a reachability of 1000 km to show the relevant structure of variations.

In order to illustrate again the difference between OPTICS and k-Means for this example, we choose 12,000 random trajectories and again embed the trajectories in a 2-dimensional space with classical MDS (cf. section 3.2.2). The reduction of the particle set is necessary to simplify the eigendecomposition of the matrix B in eq. (4), and we therefore choose $s_{min} = 30$. The corresponding spectrum of B is shown in fig. C in the appendix, showing that there are again two dominant eigenvectors~~.~~, i.e. visualizing the netwok in the plane captures the main variance of the data. Figure 8 shows the embedded trajectories together with OPTICS / DBSCAN clustering (fig. 8b) and k-Means (fig. 8c) for K=40. Figs. 9 and 10 show the corresponding clustering results in the fluid ~~domains~~domain. It is visible that k-Means does not detect a single vortex, but splits the fluid domain into regions of approximately similar size. OPTICS ~~easily~~ detects multiple Agulhas rings by finding the ~~steepest~~ deepest valleys in the reachability plot.

It is interesting to note that the use of classical MDS in fig. 9 has lead to the detection of many of the vortices of fig. 6d-f with DBSCAN instead of the ξ-clustering method. The transformation to the reduced 2D space has hence lead to a simplification of the reachability plot, which now represents the major variations in the distances of the embedded trajectories. At the same time, the large-scale structure of 6a is not visible any more in fig. 9. This indicates that exploring more dimensionality reduction techniques could be useful for future research, in particular those that are computationally more efficient than classical MDS. Spectral embeddings derived from networks together with ~~partition based~~ partition-based clustering have a similar problem as the one illustrated in ~~fig~~figs. 8c and 10 (Froyland et al., 2019). Similar to the case discussed here, OPTICS can be used to overcome the problems of k-Means. We show this in appendix C for the network proposed by Padberg-Gehle and Schneide (2017) for the Agulhas region, together with a brief introduction of the network and how to construct spectral embeddings. In summary, k-Means again fails to detect any of the vortices, while OPTICS detects many of the coherent vortices in the

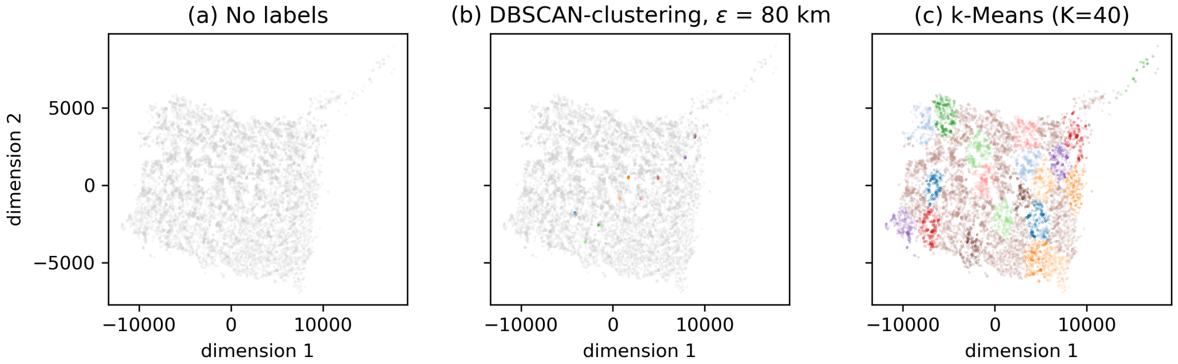


Figure 8. Embedding of the Agulhas trajectories in the 2-dimensional space defined by the leading eigenvectors of the MDS Kernel matrix B. a: no labels. b: clustering labels of OPTICS / DBSCAN, see fig. ~~8~~ 9 for the corresponding plot in the Agulhas region. Grey particles represent noise. c: k-Means with $K = 40$, see fig. 10 for the corresponding plot in the Agulhas domain.

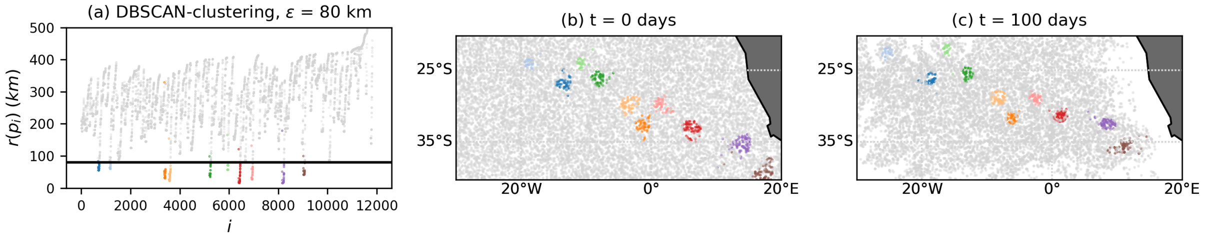


Figure 9. Result of OPTICS applied to the 2-dimensional embedding of 12,000 randomly selected particles with the classical MDS method, cf. fig. 8b, and $s_{min} = 30$. The corresponding spectrum is shown in fig. C in the appendix, showing that there are two dominant eigenvectors. Grey particles are classified as noise.

spectrally embedded network. Yet, other flow features are also present that result from the physical motivation of the network definition, see the results in appendix C.

5 Conclusions

The abstract embedding of particle trajectories in a metric space with subsequent clustering is a promising field of research for the detection of finite-time coherent sets in oceanography~~, as it can be potentially applied to sparse sets of trajectories e. g. from drifter release experiments.~~ . Yet, most of the existing methods ~~lack the ability to separate finite-time coherent structures from noisy trajectories that do not belong to any such structure, which hampers the application to large ocean domains. This is because the clustering methods proposed so far~~ have been based on graph partitioning, which ~~treats~~ has no concept of noisy, unclustered ~~data points insufficiently. In this article, we presented a simple way to overcome this problem by using~~ trajectories.

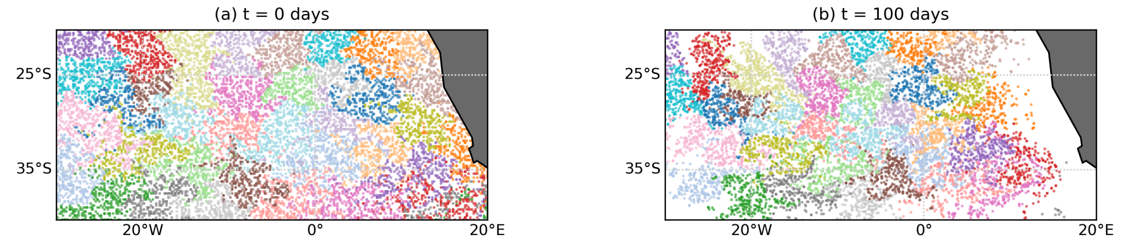


Figure 10. Result of the k-Means clustering with $K = 40$ applied to the 2-dimensional embedding with classical MDS, cf. fig. 8c.

This is a problem for applications in the ocean, where many eddies are transported in a noisy background flow on large domains. This study is motivated by the success of Froyland et al. (2019) in overcoming the problem of graph partitioning by a sophisticated form of trajectory embedding. Here, we show how the density-based clustering algorithm OPTICS (Ankerst et al., 1999) can be used instead of graph partitioning, in order to detect small-scale eddies in large ocean domains. Different from ~~partition based~~ partition-based clustering methods such as k-Means, OPTICS ~~detects the clustering structure of the embedded trajectories by looking for~~ does not require to fix the number of clusters beforehand. Clusters are detected by identifying dense accumulations of points, i.e. groups of trajectories that are close to each other in embedding space. Coherent groups of particle trajectories can be identified as valleys in the reachability plot computed by the OPTICS algorithm. This plot also has a natural interpretation in terms of cluster hierarchies, i.e. finite-time coherent sets that are by themselves part of a larger scale finite-time coherent set. Such hierarchies are present in the surface ocean flow, where the subtropical basins are approximately coherent and at the same time ~~comprise~~ contain other finite-time coherent structures such as eddies and jets.~~This hierarchical property is a clear advantage compared to DBSCAN, which has been used before to detect coherent sets (Schneide et al., 2018). One run of OPTICS can in principle produce all possible results of DBSCAN clustering for the same parameter s_{min}. In addition, different from DBSCAN, OPTICS can detect clusters of varying density, and detect them by locating the valleys with the steepest boundaries in the reachability plot with the ξ-clustering method of Ankerst et al. (1999).~~

We apply OPTICS to Lagrangian particle trajectories directly, in the spirit of Froyland and Padberg-Gehle (2015). OPTICS successfully detects the expected coherent structures in the Bickley jet model flow, separating the six vortices and the jet from background noise. We also apply ~~our method to~~ OPTICS to simulated trajectories in the eastern South Atlantic and successfully identify Agulhas rings, separated by noise. We visualize the difference ~~of OPTICS to~~ between OPTICS and k-Means with a 2-dimensional embedding of the trajectories based on classical multidimensional scaling. We also show how OPTICS can be applied to the spectral embedding of the ~~particle based~~ particle-based network proposed by Padberg-Gehle and Schneide (2017), providing a necessary amendment to ~~this~~ their method to detect coherent vortices in a large ocean domain, i.e. when k-Means fails. Our method is ~~different from previous approaches used to detect finite-time coherent sets in ocean models from Lagrangian trajectory data as it has a clear interpretability in terms of clustering hierarchy, where large-scale and small scale structures are visible in the reachability plot produced by OPTICS. Our method can also be applied to scarce trajectory data sets, i.e. without too much concern about the spatial coverage of a fluid domain with initial~~

~~conditions. Finally, our method is~~ very simple to implement in Python, as OPTICS is available in the SciPy sklearn package. While we here present the results of OPTICS with three different kinds of ~~embedding~~embeddings, it is likely that OPTICS also works for other trajectory embeddings, ~~or even other methods using clustering such as transfer operator based finite-time coherent sets (Froyland et al., 2010) or dynamic Laplacians (Froyland et al., 2019).~~ such as the spectral embeddings of Banisch and Koltai (2017) or Froyland and Junge (2018). Using such dynamically motivated embeddings instead of the ad-hoc direct embedding presented here could be a promising direction for future research.

Extending our method to datasets with more trajectories can be made more efficient by choosing a finite generating distance for OPTICS (Ankerst et al., 1999). While this is better from a computational point of view, it requires some knowledge or intuition about the spatial distribution of the embedded trajectories. A major challenge for the method proposed here is the embedding dimension. For ~~very~~ long trajectories, it is ~~important~~ necessary to reduce the dimensionality of the trajectories before applying OPTICS. A complication here is the desired property of an embedding to preserve both local and global distances in order to make full use of the hierarchical properties of OPTICS. This means, for example, that the popular method of a locally linear embedding (Roweis and Saul, 2000) is not suitable, unless only the ~~small scale~~ small-scale (densest finite-time coherent sets) are to be detected. Using classical multidimensional scaling (MDS), as we did here to visualize the clustering results, in principle preserves local and global distances, ~~but is~~ although our results indicate that the large-scale coherence structure in the Agulhas flow is less pronounced for the classical MDS embedding compared to the full embedding of trajectories. In any case, classical MDS is not an option for very large ~~data sets~~ datasets, as it requires the diagonalization of a dense symmetric square matrix of size equal to the particle number. Spectral embeddings of derived networks such as the ones of Hadjighasem et al. (2016), Padberg-Gehle and Schneide (2017) and Banisch and Koltai (2017) are useful to achieve lower-dimensional embeddings, but they come with the introduction of additional parameters for the network construction and heuristics to truncate the embedding dimension. Further research into other non-linear dimensionality reduction techniques that have not been explored in the context of finite-time coherent sets can lead to more efficient and robust methods.

Code and data availability. All code is available at https://github.com/OceanParcels/coherent_vortices_OPTICS, including the code to generate the Bickley jet trajectories. The data for the virtual particles in the South Atlantic is available on Zenodo (Wichmann, 2020b). Details on the Parcels simulation for the virtual trajectories in the ocean can be found at the GitHub repository of our previous paper, https://github.com/OceanParcels/near_surface_microplastic. The data from the NEMO ORCA-006 run are available at http://opendap4gws.jasmin.ac.uk/thredds/nemo/root/catalog.html

Appendix A: ~~Agulhas rings with smaller particle set~~Additional figures for the Bickley jet flow

Appendix B: ~~Additional figures for the classical MDS embedding~~

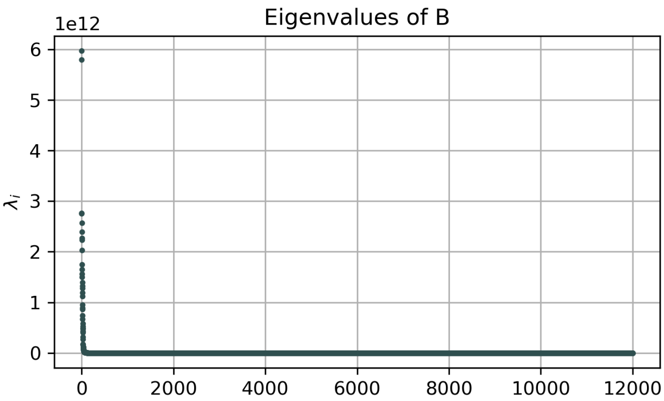


Figure A1. ~~Result~~ Spectrum of the ~~OPTICS algorithm~~ classical MDS kernel matrix B for the ~~Agulhas particles, applied to 5,000 randomly selected trajectories and $s_{min} = 20$~~Bickley jet flow. ~~The large-scale structure as well as many of the eddies~~ It is visible that there are ~~very similar to the full dataset case, see fig~~two dominant eigenvalues. ~~6. Grey particles are classified~~ We choose the vectors corresponding to these first two eigenvalues as ~~noise~~embedding vectors in section 4.1.

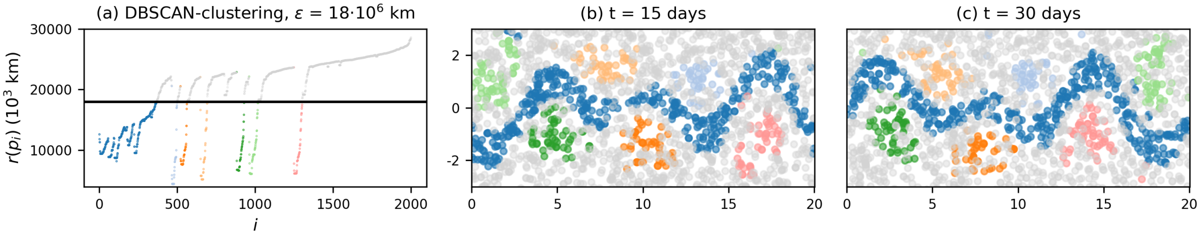


Figure A1. Result of ~~$K = 8$ k-Means clustering~~ the OPTICS algorithm for a random subset of 2,000 particles in the ~~2-dimensional embedding from classical MDS~~Bickley jet flow, ~~cf~~with particle data every 5 days instead of every day. ~~fig~~To account for the smaller number of particles, we set $s_{min} = 15$ for this case. ~~4. Axis units~~ The six vortices and the jet are ~~in 1000 km~~still clearly visible.

~~Result of OPTICS applied to the 2-dimensional embedding of 12,000 randomly selected particles with the classical MDS method, cf. fig. 8b, and $s_{min} = 30$. The corresponding spectrum is shown in fig. C in the appendix, showing that there are two dominant eigenvectors. Grey particles are classified as noise.~~

Appendix B: Additional figures for the Agulhas flow

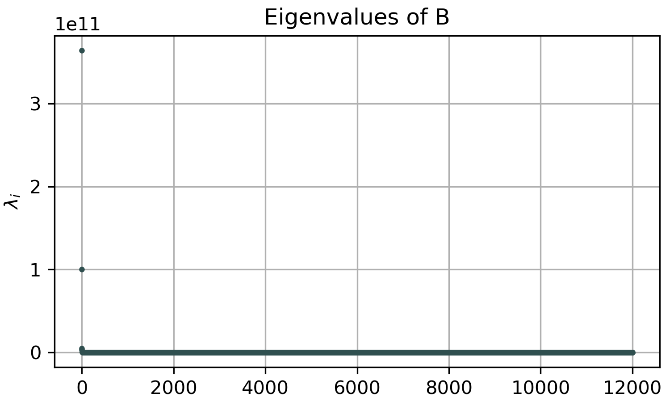


Figure B1. ~~Result~~ Spectrum of the ~~k-Means clustering with $K = 40$ applied to the 2-dimensional embedding with~~ classical MDS kernel matrix B for the Agulhas flow, ~~cf~~where we first constrain the particle data to 12,000 randomly selected trajectories. ~~fig~~There are again two dominant eigenvalues, for which we choose the corresponding vectors for the embedding in section 4.2.~~8c.~~

Appendix C: ~~Spectra of the MDS kernel matrices~~

~~Spectrum of the classical Multidimensional Scaling Kernel matrix K for the Bickley Jet example. It is visible that there are three dominant eigenvalues. In the manuscript, we choose the vectors corresponding to he first two for visualization purposes.~~

~~Spectrum of the classical Multidimensional Scaling Kernel matrix K for the Agulhas flow, where we first constrain the particle data to 12,000 randomly selected trajectories. There are again two dominant eigenvalues, for which we choose the corresponding vectors for the embedding.~~

Appendix C: Detecting Agulhas rings with a ~~particle based~~ particle-based network

535 To demonstrate that OPTICS can also be applied to the spectral embedding of a ~~particle based~~ particle-based network, we use the network proposed by Padberg-Gehle and Schneide (2017). If we have a set of particle trajectories $x_i(t)$, where $i = 1, \ldots, N$, $t = t_1, t_2, \ldots, t_T$ with N the number of particles and T the number of time steps, the network $A \in \mathbb{R}^{N \times N}$ is defined as:

$$A_{ij} = \begin{cases} 1, & \text{if } \exists t \in \{t_1, t_2, \ldots, t_T\} \;\; s.t. \;\; ||x_i(t) - x_j(t)|| < d, \\ 0, & \text{otherwise}. \end{cases} \tag{C1}$$

Here, $||.||$ denotes the Euclidean norm and ~~$d \in \mathbb{R}$~~ $d > 0$ is a fixed pre-determined cut-off parameter, see Padberg-Gehle and

540 Schneide (2017) for a discussion on the choice of d (called ϵ in Padberg-Gehle and Schneide (2017)). Similar to Padberg-Gehle

and Schneide (2017), we embed the nodes in a lower dimensional space $\mathbb{R}^K$ by means of the eigenvectors of its random walk Laplacian, (see e.g. Von Luxburg (2007))

$$L_r = D^{-1}A, \tag{C2}$$

where D is a diagonal matrix with $D_{ii} = \sum_j A_{ij}$. The embedding of node i is defined by

$$y_i = (v_{1,i}, v_{1,i}, \ldots, v_{K,i}) \in \mathbb{R}^K, \tag{C3}$$

where ~~, the~~ v_i, $i = 0, \ldots, N-1$ are the right eigenvectors corresponding to the largest eigenvalues λ_i of L_r. The eigenvalues are assumed to be ordered in descending order, i.e. $1 = \lambda_0 > \lambda_1 \geq \ldots, \geq \lambda_N$. ~~This is the most common network embedding for the detection of finite-time coherent sets so far (Padberg-Gehle and Schneide, 2017; Banisch and Koltai, 2017; Hadjighasem et al., 2016)~~ ~~.~~ The classical simultaneous K-way normalized cut proceeds with applying the k-Means algorithm to the embedding defined in eq. (C3) to detect K clusters (Von Luxburg, 2007), resulting in an approximate solution to the normalized cut problem (Shi and Malik, 2000).

Figure C1 shows the spectrum of the resulting random walk Laplacian with $d = 200$ km. No obvious spectral gap is visible that would suggest a truncation of the embedding space. Figure C2 shows the clustering result if we apply a k-Means algorithm as suggested by Padberg-Gehle and Schneide (2017) to detect $K = 40$ clusters. It is visible that the ~~partition based~~ partition-based k-Means clustering method does not detect any individual Agulhas rings, but partitions the state space into regions of approximately equal size.

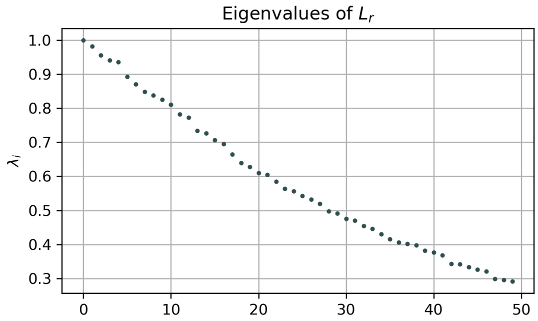


Figure C1. Spectrum of the random walk Laplacian, cf. eq. (C2) of the network proposed by Padberg-Gehle and Schneide (2017) applied to the Agulhas trajectory data. No clear gap exists that suggest a truncation of the embedding.

Applying OPTICS instead ~~ot~~ of k-Means with a subsequent ξ-clustering detects some of the Agulhas rings, see fig. C3, where we choose $s_{min} = 100$ as in section 4.2. Note that also other structures than typical circular eddies are detected. While this depends on the clustering parameter ξ (or ϵ for DBSCAN), this is also a consequence of the *physically motivated* network defined by eq. (C3), where particles are connected equally if they are close to each other at least once in time. This is different from the direct embedding, where we require particles to stay close to each other along the entire trajectory.

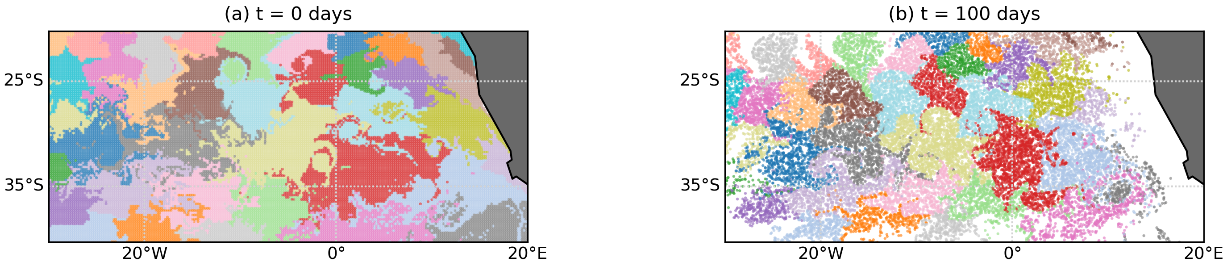


Figure C2. Result of k-Means clustering applied to the 40 leading eigenvectors of the random walk Laplacian, cf. eq. (C2), looking for 40 clusters. No individual vortices are detected.

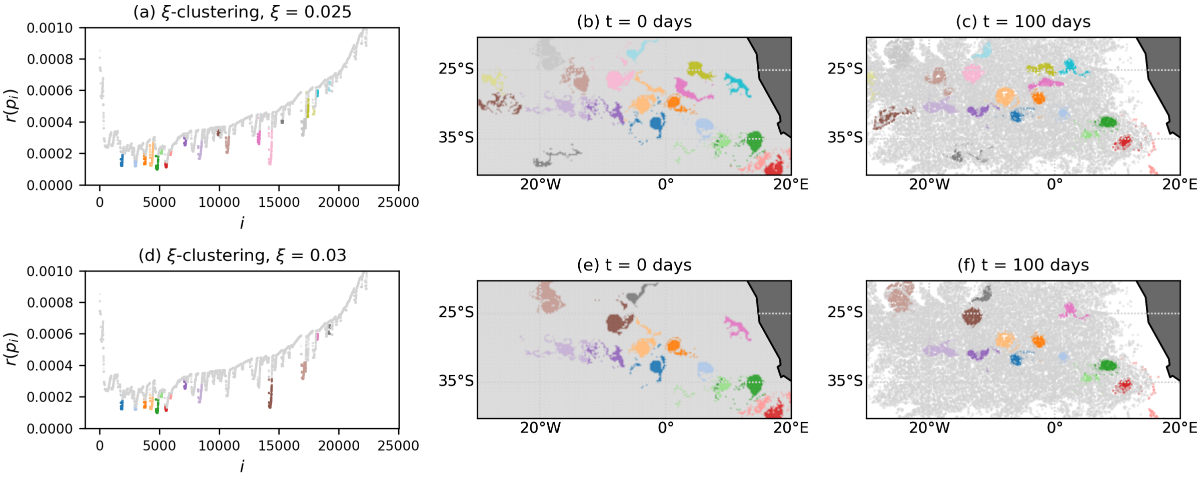


Figure C3. Result of ~~optics~~ OPTICS applied to the $K = 40$ spectral embedding of the network defined in eq. (C1) with $d = 200$ km and $s_{min} = 100$. Grey particles are classified as noise.

Author contributions. DW performed the analysis, with support from CK, EvS and HD. DW wrote the manuscript and all authors jointly edited and revised it.

Competing interests. The authors declare no competing interests

565 *Acknowledgements.* David Wichmann, Christian Kehl and Erik van Sebille are supported through funding from the European Research Council (ERC) under the European Union Horizon 2020 research and innovation programme (grant agreement No 715386). This work was partially carried out on the Dutch national e-infrastructure with the support of SURF Cooperative (project no. 16371). We thank Andrew Coward for providing the ORCA-N006 simulation data.

References

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J.: OPTICS: Ordering Points to Identify the Clustering Structure, ACM Sigmod record, 28, 49–60, https://doi.org/https://doi.org/10.1145/304181.304187, 1999.

Banisch, R. and Koltai, P.: Understanding the geometry of transport: Diffusion maps for Lagrangian trajectory data unravel coherent sets, Chaos: An Interdisciplinary Journal of Nonlinear Science, 27, 035 804, https://doi.org/https://doi.org/10.1063/1.4971788, 2017.

Beron-Vera, F. J., Wang, Y., Olascoaga, M. J., Goni, G. J., and Haller, G.: Objective Detection of Oceanic Eddies and the Agulhas Leakage, Journal of Physical Oceanography, 43, 1426–1438, https://doi.org/https://doi.org/10.1175/JPO-D-12-0171.1, 2013.

Bickley, W.: LXXIII. The plane jet, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 23, 727–731, https://doi.org/https://doi.org/10.1080/14786443708561847, 1937.

Brach, L., Deixonne, P., Bernard, M. F., Durand, E., Desjean, M. C., Perez, E., van Sebille, E., and ter Halle, A.: Anticyclonic eddies increase accumulation of microplastic in the North Atlantic subtropical gyre, Marine Pollution Bulletin, 126, 191–196, https://doi.org/https://doi.org/10.1016/j.marpolbul.2017.10.077, 2018.

del Castillo-Negrete, D. and Morrison, P.: Chaotic transport by Rossby waves in shear flow, Physics of Fluids A: Fluid Dynamics, 5, 948–965, https://doi.org/https://doi.org/10.1063/1.858639, 1993.

Delandmeter, P. and van Sebille, E.: The Parcels v2.0 Lagrangian framework: new field interpolation schemes, Geoscientific Model Development, 12, 3571–3584, https://doi.org/https://doi.org/10.5194/gmd-12-3571-2019, 2019.

Dencausse, G., Arhan, M., and Speich, S.: Routes of Agulhas rings in the southeastern Cape Basin, Deep-Sea Research Part I: Oceanographic Research Papers, 57, 1406–1421, https://doi.org/https://doi.org/10.1016/j.dsr.2010.07.008, 2010.

Dong, C., McWilliams, J. C., Liu, Y., and Chen, D.: Global heat and salt transports by eddy movement, Nature Communications, 5, 3294, https://doi.org/https://doi.org/10.1038/ncomms4294, 2014.

Dussin, R., Barnier, B., and Brodeau, L.: The making of Drakkar forcing set DFS5, Tech. rep., LGGE, Grenoble, France., 2016.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, p. 226–231, AAAI Press, 1996.

Fouss, F., Saerens, M., and Shimbo, M.: Algorithms and models for network data and link analysis, Cambridge University Press, 2016.

Froyland, G. and Junge, O.: Robust FEM-based extraction of finite-time coherent sets using scattered, sparse, and incomplete trajectories, SIAM Journal on Applied Dynamical Systems, 17, 1891–1924, https://doi.org/https://doi.org/10.1137/17M1129738, 2018.

Froyland, G. and Padberg-Gehle, K.: A rough-and-ready cluster-based approach for extracting finite-time coherent sets from sparse and incomplete trajectory data, Chaos: An Interdisciplinary Journal of Nonlinear Science, 25, 087 406, https://doi.org/https://doi.org/10.1063/1.4926372, 2015.

Froyland, G., Santitissadeekorn, N., and Monahan, A.: Transport in time-dependent dynamical systems: Finite-time coherent sets, Chaos: An Interdisciplinary Journal of Nonlinear Science, 20, 043 116, https://doi.org/https://doi.org/10.1063/1.3502450, 2010.

Froyland, G., Stuart, R. M., and van Sebille, E.: How well-connected is the surface of the global ocean?, Chaos: An Interdisciplinary Journal of Nonlinear Science, 24, 033 126, https://doi.org/https://doi.org/10.1063/1.4892530, 2014.

Froyland, G., Horenkamp, C., Rossi, V., and van Sebille, E.: Studying an Agulhas ring's long-term pathway and decay with finite-time coherent sets, Chaos: An Interdisciplinary Journal of Nonlinear Science, 25, 083 119, https://doi.org/https://doi.org/10.1063/1.4927830, 2015.

Froyland, G., Rock, C. P., and Sakellariou, K.: Sparse eigenbasis approximation: Multiple feature extraction across spatiotemporal scales with application to coherent set identification, Communications in Nonlinear Science and Numerical Simulation, 77, 81 – 107, https://doi.org/https://doi.org/10.1016/j.cnsns.2019.04.012, 2019.

Hadjighasem, A., Karrasch, D., Teramoto, H., and Haller, G.: Spectral-clustering approach to Lagrangian vortex detection, Physical Review E, 93, 063 107, https://doi.org/https://doi.org/10.1103/PhysRevE.93.063107, 2016.

Hadjighasem, A., Farazmand, M., Blazevski, D., Froyland, G., and Haller, G.: A critical comparison of Lagrangian methods for coherent structure detection, Chaos: An Interdisciplinary Journal of Nonlinear Science, 27, 053 104, https://doi.org/https://doi.org/10.1063/1.4982720, 2017.

Haller, G. and Beron-Vera, F. J.: Coherent Lagrangian vortices: The black holes of turbulence, Journal of Fluid Mechanics, 731, R4, https://doi.org/https://doi.org/10.1017/jfm.2013.391, 2013.

Lange, M. and van Sebille, E.: Parcels v0.9: prototyping a Lagrangian ocean analysis framework for the petascale age, Geoscientific Model Development, 10, 4175–4186, https://doi.org/10.5194/gmd-10-4175-2017, https://gmd.copernicus.org/articles/10/4175/2017/, 2017.

Ma, T. and Bollt, E. M.: Relatively Coherent Sets as a Hierarchical Partition Method, International Journal of Bifurcation and Chaos, 23, 1330 026, https://doi.org/https://doi.org/10.1142/S0218127413300267, 2013.

Madec, G.: NEMO ocean engine, Note du Pôle de modélisation, No 27, 2008.

Padberg-Gehle, K. and Schneide, C.: Network-based study of Lagrangian transport and mixing, Nonlinear Processes in Geophysics, 24, 661–671, https://doi.org/https://doi.org/10.5194/npg-24-661-2017, 2017.

Roweis, S. T. and Saul, L. K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science, 290, 2323–2326, https://doi.org/https://doi.org/10.1126/science.290.5500.2323, 2000.

Schneide, C., Pandey, A., Padberg-Gehle, K., and Schumacher, J.: Probing turbulent superstructures in Rayleigh-Bénard convection by Lagrangian trajectory clusters, Physical Review Fluids, 3, 113 501, https://doi.org/https://doi.org/10.1103/PhysRevFluids.3.113501, 2018.

Schouten, M. W., de Ruijter, W. P. M., van Leeuwen, P. J., and Lutjeharms, J. R. E.: Translation, decay and splitting of Agulhas rings in the southeastern Atlantic Ocean, Journal of Geophysical Research: Oceans, 105, 21 913–21 925, https://doi.org/https://doi.org/10.1029/1999jc000046, 2000.

Shi, J. and Malik, J.: Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22, 888–905, https://doi.org/https://doi.org/10.1109/34.868688, 2000.

Tarshish, N., Abernathey, R., Zhang, C., Dufour, C. O., Frenger, I., and Griffies, S. M.: Identifying Lagrangian coherent vortices in a mesoscale ocean model, Ocean Modelling, 130, 15–28, https://doi.org/https://doi.org/10.1016/j.ocemod.2018.07.001, 2018.

Van Sebille, E., Aliani, S., Law, K. L., Maximenko, N., Alsina, J. M., Bagaev, A., Bergmann, M., Chapron, B., Chubarenko, I., Cózar, A., Delandmeter, P., Egger, M., Fox-Kemper, B., Garaba, S. P., Goddijn-Murphy, L., Hardesty, B. D., Hoffman, M. J., Isobe, A., Jongedijk, C. E., Kaandorp, M. L., Khatmullina, L., Koelmans, A. A., Kukulka, T., Laufkötter, C., Lebreton, L., Lobelle, D., Maes, C., Martinez-Vicente, V., Morales Maqueda, M. A., Poulain-Zarcos, M., Rodríguez, E., Ryan, P. G., Shanks, A. L., Shim, W. J., Suaria, G., Thiel, M., Van Den Bremer, T. S., and Wichmann, D.: The physical oceanography of the transport of floating marine debris, Environmental Research Letters, 15, 023 003, https://doi.org/https://doi.org/10.1088/1748-9326/ab6d7d, 2020.

Von Luxburg, U.: A Tutorial on spectral clustering, Statistics and Computing, 17, 395–416, https://doi.org/https://doi.org/10.1007/s11222-007-9033-z, 2007.

Wichmann, D.: Animation of finite-time coherent sets in the Agulhas region, https://doi.org/https://doi.org/10.5281/zenodo.4103741, 2020a.

Wichmann, D.: Lagrangian particle dataset (2 years) for Agulhas region surface flow, https://doi.org/https://doi.org/10.5281/zenodo.3899942, 2020b.

645 Wichmann, D., Delandmeter, P., and van Sebille, E.: Influence of Near-Surface Currents on the Global Dispersal of Marine Microplastic, Journal of Geophysical Research: Oceans, 124, 6086–6096, https://doi.org/https://doi.org/10.1029/2019JC015328, 2019.

Wichmann, D., Kehl, C., Dijkstra, H. A., and van Sebille, E.: Detecting flow features in scarce trajectory data using networks derived from symbolic itineraries: an application to surface drifters in the North Atlantic, Nonlinear Processes in Geophysics Discussions, 2020, 1–20, https://doi.org/https://doi.org/10.5194/npg-2020-18, 2020.