Estimating the Travel Time and the Most Likely Path from Lagrangian Drifters

MICHAEL O'MALLEY,[a] ADAM M. SYKULSKI,[a] ROMUALD LASO-JADART,[b] AND MOHAMMED-AMIN MADOUI[b]

[a] *Lancaster University, Lancashire, United Kingdom*
[b] *Génomique Métabolique, Genoscope, Institut F. Jacob, CEA, CNRS, Univ. Evry, Univ. Paris-Saclay, Evry, France*

(Manuscript received 14 August 2020, in final form 26 January 2021)

ABSTRACT: We provide a novel method for computing the most likely path taken by drifters between arbitrary fixed locations in the ocean. We also provide an estimate of the travel time associated with this path. Lagrangian pathways and travel times are of practical value not just in understanding surface velocities, but also in modeling the transport of oceanborne species such as planktonic organisms and floating debris such as plastics. In particular, the estimated travel time can be used to compute an estimated Lagrangian distance, which is often more informative than Euclidean distance in understanding connectivity between locations. Our method is purely data driven and requires no simulations of drifter trajectories, in contrast to existing approaches. Our method scales globally and can simultaneously handle multiple locations in the ocean. Furthermore, we provide estimates of the error and uncertainty associated with both the most likely path and the associated travel time.

KEYWORDS: Ocean; Lagrangian circulation/transport; Transport; Optimization; Statistical techniques

1. Introduction

The Lagrangian study of transport and mixing in the ocean is of fundamental interest to ocean modelers (van Sebille et al. 2018, 2009; LaCasce 2008). In particular, the analysis of data obtained from Lagrangian drifting objects greatly contribute to our knowledge of ocean circulation, e.g., through analyzing the accuracy of numerical and stochastic models (Huntley et al. 2011; Sykulski et al. 2016), or the use of drifter data to better understand various pathways and where to search for marine debris (Miron et al. 2019; van Sebille et al. 2012; McAdam and van Sebille 2018).

Meehl (1982) used ship-drift data to create a surface velocity dataset on a $5° \times 5°$ grid. These velocities were used to simulate the Lagrangian drift of floating objects in Wakata and Sugimori (1990). More recent works focus on using drifting buoys to derive Lagrangian models to discover areas where floating debris tends to end up (van Sebille 2014; van Sebille et al. 2012; Maximenko et al. 2012). Advances in technology have resulted in much better data quality, which now permits the use of a more detailed method. The newer models provide densities of where debris ends up on grid scales as small as $0.5° \times 0.5°$.

In this paper, we propose a novel computationally fast method for estimating a so-called *most likely pathway* between two points in the ocean, alongside an estimated travel time for this pathway. The method is purely data driven. We demonstrate our method on data from the *Global Drifter Program* (GDP), but the method is designed to work with any Lagrangian tracking dataset. Additionally, we develop and test a related method for providing uncertainty on both the pathways and the travel times. Our method is automated with little expert knowledge needed from the practitioner. We provide a set of default parameters that allow the method to run as intended. The method simply takes in a set of locations within the ocean, and outputs a data structure containing most likely paths and corresponding travel time estimates between all pairs of locations. We focus on a global scale: we aim to provide a measure of Lagrangian connectivity for locations that are thousands of kilometers apart. An individual drifter trajectory is unlikely to connect two arbitrary locations far apart; hence the need for our method that fuses information across many drifters.

A tool that predicts travel times is of practical use in many fields. For example in ecological studies of marine species, genetic measurements are taken at various locations in the ocean (Watson 2018). Euclidean distance is often used as a measure of separability and isolation by distance (Becking et al. 2006; Ellingsen and Gray 2002) to find correlations with diversity metrics or genetic differentiation between communities or populations of organisms. Due to various currents and land barriers, we expect Euclidean distance to often be a poor measure of how ''distant'' or dissimilar communities or populations sampled in two locations are. The method proposed in this work would use the estimated travel times to supply a matrix containing a *Lagrangian distance* measure between all pairs of locations. This matrix can then be contrasted with a pairwise genetic distance matrix between these locations and

Denotes content that is immediately available upon publication as open access.

Supplemental information related to this paper is available at the Journals Online website: https://doi.org/10.1175/JTECH-D-20-0134.s1.

Corresponding author: Michael O'Malley, m.omalley2@lancaster.ac.uk

DOI: 10.1175/JTECH-D-20-0134.1

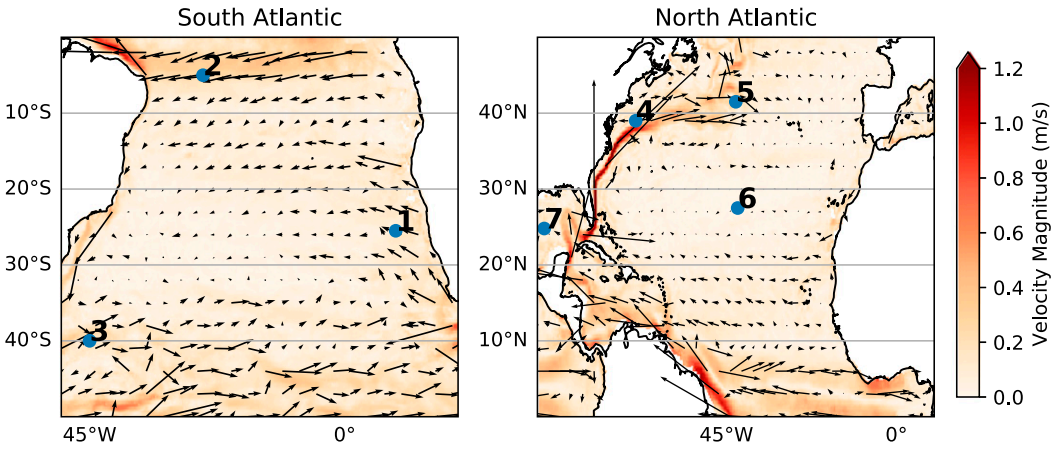

FIG. 1. Locations of interest from Table 1. Annual mean values of the near-surface currents derived from drifter velocities (Laurindo et al. 2017) are plotted. Arrows are drawn on a 3° × 3° grid to show current direction.

will yield new insights. In many instances the Lagrangian distance matrix will be more correlated with genetic relatedness than a Euclidean distance matrix. This observation was already made in the Mediterranean Sea when studying plankton (Berline et al. 2014), and off the coast of California for a species of sea snail (White et al. 2010). Both of the works by Berline et al. (2014) and White et al. (2010) rely on simulating trajectories from detailed ocean current datasets to estimate the Lagrangian distance. Such approaches do not scale globally and rely on simulated trajectories from currents rather than real observations.

In Fig. 1, we show seven locations plotted on a map with ocean currents. We use these locations as a proof-of-concept example throughout this paper. The exact coordinates are given in Table 1. The aim is to introduce and motivate a method that provides an estimate as to how long it would take to drift between any two of these locations. For example, the travel time from location 2 to location 3 in the South Atlantic Ocean should be smaller than the return journey because of the Brazil Current. We choose to include locations in both the North and South Atlantic as we wish to demonstrate that the method successfully finds pathways linking points that are extremely far apart.

Comparison with related works

In this section we give a brief overview of techniques that have used the Global Drifter Program to achieve a similar or related task. The work by Rypina et al. (2017) proposes a statistical approach for obtaining travel times. A source area is defined such that at least 100 drifters pass through the source area. The method focuses on obtaining a spatial probability map and a mean travel time for every 1° × 1° bin outside of the source area. This method successfully combines many trajectories; however, for multiple locations one would have to decide on a varying grid box for each location of interest. Such a grid box must be manually chosen by the practitioner meaning that the method does not scale well with multiple locations. Rypina et al. (2017) also focus on estimating a mean travel time, where our method provides a travel time associated with the most likely path and is hence more akin to estimating a mode or median travel time.

The method by van Sebille et al. (2011), which proposes the use of Monte Carlo supertrajectories (MCST), could naturally be used to estimate travel times. This method simulates new trajectories as sequences of unique grid indices along with corresponding travel time estimates for each part of that journey. The method is purely data driven; i.e., they only use real trajectories to fit the model. The travel time and pathway we supply here should be similar to the most likely MCST to occur between the two points. The advantage of our method is that we do not base the analysis on a simulation, such that the results from the method described in section 3 are not subject to any randomness due to simulation.

Various other works have made attempts to compute Lagrangian-based distances. For example, Berline et al. (2014) used numerically simulated trajectories to estimate *mean connection times* within the Mediterranean Sea. Smith et al. (2018) used MCST to estimate various statistics of how seagrass fragments could drift from the southeast coast of Australia to Chile. Specifically, Smith et al. (2018) simulated 10 million MCST starting from the SE coast of Australia and only 264 (0.002 64%) of the simulated trajectories were found to travel roughly to the Chilean coast.

The approach by Jönsson and Watson (2016) uses simulated drifter data to construct connectivity matrices between locations in the ocean. As the matrix is sparse, Dijkstra's algorithm is used to connect arbitrarily distant locations in the ocean to

TABLE 1. Table of station locations.

	Longitude	Latitude
1	9.0	−25.5
2	−25.0	−5.0
3	−45.0	−40.0
4	−69.0	39.0
5	−42.5	41.5
6	−42.0	27.5
7	−93.2	24.8

measure Lagrangian distance. Although this method may at first glance bare similarities with our method (specifically in the use of Dijkstra's algorithm), there are in fact many differences. First of all, the method uses simulated trajectories whereas we use real drifter trajectories. Second, Dijkstra's algorithm is performed by Jönsson and Watson (2016) on the *connectivity* matrix (which finds minimum connection times between locations), whereas our approach uses Dijkstra's algorithm on the *transition* matrix, which describes a probabilistic framework for drifter movement. We found the latter approach to perform much better with real data. Finally, we cannot directly implement the approach described in Jönsson and Watson (2016) as only connectivity values higher than one year are used by the algorithm. For real data such a step would result in a very sparse connectivity matrix making the method infeasible. An initial analysis we conducted using a similar method achieved poor results.

There are a variety of works that use Markov transition matrices for different aims to this work. Ser-Giacomi et al. (2015b) and Miron et al. (2019) look at probable paths, where both of these works find a path going between two points in a certain number of days using a dynamic program. Froyland et al. (2014) and Miron et al. (2017) study ocean dynamics by analyzing eigenvalues of the transition matrix. Other methods in the literature include characterizing dispersion and mixing (Ser-Giacomi et al. 2015a), identification coherent regions (Froyland et al. 2007; Ser-Giacomi et al. 2015a), forward integration of tracers (van Sebille et al. 2012; Maximenko et al. 2012), and guiding drifter deployments (Lumpkin et al. 2016). We differ from these works as we ultimately aim to find travel times, as well as pathways, between multiple fixed locations.

Our proposed algorithm for computing travel times and pathways will also use the aforementioned Markov transition matrix approach. Our key novelty is that we build on this conceptual approach by implementing and demonstrating the benefits of using the H3 spatial indexing system for discretization, and by supplying uncertainty quantification guidelines by applying grid rotations and data bootstrapping. The steps outlined in algorithm 1 are individually known across disparate literature; however, this is the first paper to our knowledge that effectively combines these steps to solve the problem of interest. We provide numerous examples to show how our method robustly outperforms state-of-the-art alternative approaches. In addition, we supply freely available software in the form of a Python package, of which all parameters in the model can easily be customized to suit the needs of the practitioner.

In summary, the novel contributions of this work are (i) the combination of the steps in section 3 to form a computationally efficient algorithm that applies directly to transition matrices to find most likely paths and travel times simultaneously, (ii) computation of uncertainty from discretization error and data sampling (section 4), and (iii) the demonstration of the method showing it successfully obtains robust measures of connectivity between both very distant and closely located points (section 5). The key outcome is that we obtain oceanographic travel times and most likely paths requiring no simulated trajectories.

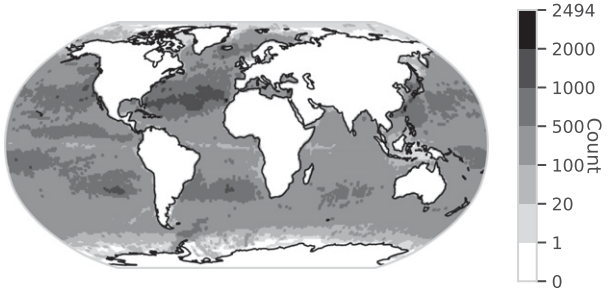

FIG. 2. Number of observations from the Global Drifter Program in each 1° × 1° longitude–latitude box.

We believe our method is preferable to Rypina et al. (2017) as we do not require custom treatment to different source areas. Jönsson and Watson (2016) requires the simulation of many very long and expensive-to-compute trajectories that obtain spurious results on real data. Using MCSTs as in Smith et al. (2018) relies on simulation. The estimation of a full pairwise travel time matrix of the locations in Table 1 requires 42 travel time estimations. With MCSTs this would likely require the simulation of millions of trajectories and manual analysis of each location pair. Our method, in contrast, can produce such a travel time matrix in a matter of seconds given that the transition matrix needs to be estimated just once a priori. In a similar manner, global travel time maps can be made in a matter of minutes, such as those that we will be showing in section 5.

2. Background and notation

a. Global Drifter Program

The GDP is a database managed by the National Oceanographic and Atmospheric Administration (NOAA) (Lumpkin and Centurioni 2019; Lumpkin and Pazos 2007). This dataset contains over 20 000 free-floating buoys temporally spanning from 15 February 1979 through to the current day. These buoys are referred to as *drifters*. The drifter design comprises a subsurface float and a drogue sock. Often this drogue sock detaches. We refer to the drifters that have lost their drogue sock as nondrogued drifters and use drogued for those that still have the drogue attached.

Here we use the drifter data recorded up to July 2020. We use data that have been recorded from drogued drifters only. This results in a total of 23 461 drifters being used, where the spatial distribution of observations is shown in Fig. 2. Only using drogued drifters is not a restriction; it would be straightforward to simply use the data from nondrogued drifters if a practitioner was interested in a species or object that experiences a high wind forcing, or a combination of both if it is believed that the species followed a mixture of near surface and wind-forced currents. The data are quality controlled and interpolated to 6-hourly intervals using the method from Hansen and Poulain (1996). These interpolated values do contain some noise due to both satellite error and interpolation; however, the magnitude of this noise is negligible in comparison to the size of grid we use in section 3. Hence, we

ignore this noise and treat the interpolated values as observations. For the same reason we note that the interpolation method used is not important here, instead of the six hourly product we could use the hourly product produced by a method proposed by Elipot et al. (2016), or drifter data smoothed by splines as proposed by Early and Sykulski (2020).

The value of using the Global Drifter Program is we obtain a true model-free representation of the ocean. All phenomena that act on the drifters are accounted for in the dataset. The other common approach is to first obtain an estimate of the underlying velocity field, then simulate thousands of trajectories using the velocity field. While this simulation approach is often satisfactory in some applications, the models generally do not agree completely with the actual observations.

b. Notation

Here we use $x°$, $y°$ to be a geographic coordinate corresponding to latitude and longitude, respectively. We refer to the longitude–latitude grid system using the notation $x° \times y°$, which means each grid box goes $x°$ along the longitude axis and $y°$ along the latitude axis. We use boldface font for any data that are in longitude–latitude pairs (e.g., $\mathbf{r} = r_{\text{lon}}, r_{\text{lat}}$) and nonboldface text for either a grid index or a single number. We use $\mathcal{S}$ to denote the set of all possible grid indices. A full table of notation is given in section b of the appendix.

c. Capturing drifter motion

We define the drifter's probability density function as

$$P(\mathbf{r}_1, t|\mathbf{r}_0, t_0),$$

where the drifter started at $\mathbf{r}_0 \in \mathbb{R}^2$ at time t_0 and moved to position $\mathbf{r}_1 \in \mathbb{R}^2$ at time t, where $\mathbf{r}_0$ and $\mathbf{r}_1$ are longitude–latitude pairs. In the absence of a model, this probability density cannot be estimated continuously from data alone. Therefore, we follow previous works that spatially discretize the problem (Maximenko et al. 2012; van Sebille et al. 2011; Miron et al. 2019; Rypina et al. 2017; Lumpkin et al. 2016). Instead of considering $\mathbf{r}_0 \in \mathbb{R}^2$, we consider $r_0 \in \mathcal{S}$, where $\mathcal{S}$ is some set of states that correspond to a polygon in space; we will define how these are formed in section 3b. Often these states are simply $1° \times 1°$ boxes (e.g., as used in Fig. 2). As in Maximenko et al. (2012), we assume that the process driving the drifter's movement is temporally stationary. In other words,

$$P(r_1, t|r_0, t_0) = P(r_1|r_0, t - t_0), \quad r_0, r_1 \in \mathcal{S};$$

that is, the probability of going from r_0 to r_1 depends only on the time increment. The probability does not depend on the start or finish time.

Furthermore, given that we are using data that are observed at regular and discrete times, we shall only consider discrete values of time. Let $\mathbf{s} = \{s_0, s_1, s_2, \ldots, s_n\}$ be a sequence of locations equally spaced in time where each entry s_i can take the value of anything within $\mathcal{S}$. We define the probability $p(s_{i+1} = q|s_i = k)$ as the probability that the position at time $i + 1$ is q given that the state at time i was k where $q, k \in \mathcal{S}$.

A Lagrangian decorrelation time causes the drifter to "forget" its history (LaCasce 2008). We aim to choose a quantity that is globally higher than the Lagrangian decorrelation time. The reasoning behind using this time is that if we consider a sequence of observations, which are at least the Lagrangian decorrelation time apart then the following Markov property is satisfied:

$$\begin{aligned} p(s_{i+1} = q_{i+1}|s_i = q_i, s_{i-1} = q_{i-1}, \ldots, s_0 = q_0) \\ = p(s_{i+1} = q_{i+1}|s_i = q_i), \end{aligned} \tag{1}$$

where q_i is just some fixed state and s_i is the random process. In other words, the Markov property states that probability of transition to state s_{i+1} is independent of all the past states at times $i - 1$ and earlier, given the state at time i is known. In this case, the physical time difference associated with $i + 1$ and i being larger than the chosen Lagrangian decorrelation time validates the use of the Markov assumption.

For the rest of this paper we assume that the time between discrete time observations is equal to $\mathcal{T}_L$. We call this quantity the Lagrangian cutoff time. Setting $\mathcal{T}_L$ higher than the decorrelation time allows us to use the Markov property from Eq. (1) freely. In so doing, alongside the simplification of discretizing locations, this allows the problem to be treated as a discrete time Markov chain. Here we fix $\mathcal{T}_L = 5$ days as this matches previous similar works (Maximenko et al. 2012; Miron et al. 2019). The estimated decorrelation time for the majority of the surface of the ocean is likely to be lower than 5 days [e.g., see Zhurbas and Oh (2004) for the Pacific Ocean and Lumpkin et al. (2002) for regions in the Atlantic Ocean]. In section e of the appendix, we conduct a sensitivity analysis to show our results are not overly sensitive to the choice of $\mathcal{T}_L$ as long as $\mathcal{T}_L > 2$ days.

3. Method for computing the most likely path and travel time

Maximenko et al. (2012) and van Sebille et al. (2012) focus on the use of a transition matrix estimated from drifters to discover points where drifters are likely to end up. In this section we build on such an approach by providing a method to take such a matrix and provide an ocean pathway and travel time.

In section 3a, we explain in detail how the transition matrix is formed. As a grid system is needed to form the discretization of data we introduce our chosen system in section 3b. Then in section 3c, we describe how we estimate the most likely path of a drifter to have taken. Finally, in section 3d, we explain how we turn the most likely path and transition matrix into an estimate of travel time. We give a summary of how this articulates in the pseudocode in algorithm 1.

a. Transition matrix

The location of a drifter at any given time is a continuous vector in $\mathbb{R}^2$, the longitude and latitude of the point. We define an injective map that maps this continuous process onto a discrete set of states that are indexed by integers in $\mathcal{S}$. We define the map as follows:

$$f:\mathbb{R}^2 \to \mathcal{S}. \tag{2}$$

We aim to make a Markov transition matrix $\mathbf{T}$ of size n_{states} rows and columns, where $T_{s,q}$ denotes, the probability of moving from s to q in one time step. Similar to the approach of Maximenko et al. (2012), we form our transition matrix using a gap method. In each drifter trajectory we only consider observations as a pair of points $\mathcal{T}_L$ days apart. When using this method for other applications we advise using $\mathcal{T}_L$ to be higher than the decorrelation time of velocity to justify the Markov assumption.

Consider a trajectory as a sequence of positions $\mathbf{y}_j = \{\mathbf{y}_{i,j}\}_{i=1}^{n_j}$ where j is the jth of N trajectories, n_j is the number of location observations in the trajectory, and $\mathbf{y}_{i,j} \in \mathbb{R}^2$ are the longitude–latitude positions. First, we map each trajectory into observed discrete states. We will denote these states as follows:

$$g_{i,j} = f(\mathbf{y}_{i,j}). \tag{3}$$

For each $s, p \in \mathcal{S}$ we estimate the relevant entry of our transition matrix $\mathbf{T}$ through using the following empirical estimate:

$$T_{s,p} = \frac{\sum_{j=1}^{N}\sum_{i=1}^{n_j-4\mathcal{T}_L} \mathbb{I}[g_{i+4\mathcal{T}_L,j} = p]\mathbb{I}[g_{i,j} = s]}{\sum_{j=1}^{N}\sum_{i=1}^{n_j-4\mathcal{T}_L} \mathbb{I}[g_{i,j} = s]}, \tag{4}$$

where $\mathbb{I}$ is the indicator function, such that it takes the value 1 if the statement inside it is true, and zero otherwise. Note that we take gaps of $4\mathcal{T}_L$ as observations are every 6 h in the GDP application and $\mathcal{T}_L$ is in days. The estimation of the transition matrix, using the discretization of trajectories in Eq. (3), in combination with Eq. (4), is commonly referred to as Ulam's approach (Ulam 1960). We expect that states in $\mathcal{S}$ that are not spatially close will have nonzero entries such that the matrix $\mathbf{T}$ will be very sparse, but this is not a problem for the method to work over large distances as we shall see.

b. Spatial indexing

Clearly the resulting transition matrix described in section 3a strongly depends on the choice of grid function in Eq. (2). Most previous works (van Sebille et al. 2012; Maximenko et al. 2012; Rypina et al. 2017; McAdam and van Sebille 2018; Miron et al. 2019) use longitude–latitude-based square grids where all grid boxes typically vary between 0.5° × 0.5° and 1° × 1°. A 1° × 1° grid cell around the equatorial region will be approximately equal area to a 111.2 km × 111.2 km square box. However, if we take such a grid above 60° latitude—for example, the Norwegian Sea—the grid cell will be approximately equal area to a 55.6 km × 111.2 km square box.

There are a few other choices that we argue are more suitable for tracking moving data on the surface of Earth. Typically, three types of grids exist for tessellating the globe: triangles, squares, or a mixture of hexagons and pentagons. Here we choose to use hexagons and pentagons as they have the desirable property that every neighboring shape shares precisely two vertices and an edge. This is different to say a square grid where only side-by-side neighbors share two vertices and an edge, whereas diagonal neighbors share only a vertex. This equivalence-of-neighbors property for hexagons and pentagons is clearly desirable for the tracking of objects as this will result in a smoother transition matrix.

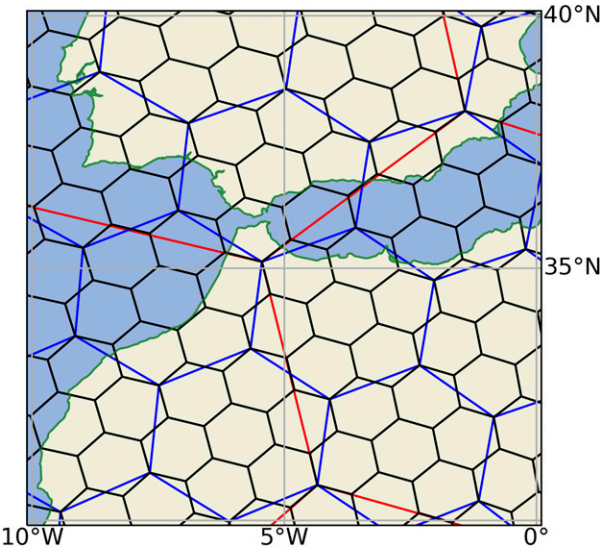

FIG. 3. A small area around the Strait of Gibraltar that is tessellated using the H3 spatial index. We show resolutions 1, 2, and 3 in red, blue, and black, respectively. Black is the resolution used in this work.

We specifically use the grid system called H3 by UBER (UBER 2019). This system divides the globe such that any longitude and latitude coordinate is mapped to a unique hexagon or pentagon. This shape will have a unique *geohash* that we can use to keep track of grid index. The benefit of using such a spatial indexing system is that attention is paid to ensuring that each hexagon is approximately equal area. We use the *resolution 3* index in which each hexagon has an average area of 12 392 km^2. A square box of size 111.32 km × 111.32 km has roughly the same area as this, which is very similar to the size of a 1° × 1° grid cell near the equator. An example of an area tessellated by H3 is shown in Fig. 3. Other potential systems that could be used include S2 by Google, which is a square system, or we could simply use a longitude–latitude system as various other works do. We show some example pathways using different grid systems and resolutions in Fig. S1 of the online supplemental material. The longitude–latitude system results in pathways that unrealistically follow long blockwise vertical or horizontal straight-line motions, in contrast to the more realistic and meandering pathways produced by the hexagonal–pentagonal H3 grid system.

c. Most likely path

For our analysis, the first step is to define a most likely path. A path is simply a sequence of states such that the first element is the origin and the last element is the destination.

We also require that two neighboring states are not equal to each other.

1) DEFINITION 1 (PATH)

We define the space of possible paths $\mathcal{P}_{o,d}$, between the origin $o \in \mathcal{S}$ and destination $d \in \mathcal{S}$, as the following:

$$\mathcal{P}_{o,d} = \{\mathbf{p} = (p_0, p_1, p_2, \ldots, p_n) : p_i \in \mathcal{S},$$
$$\forall i \in \{1, \ldots, n-1\}, p_0 = o, p_n = d, p_{i-1} \neq p_i\},$$

with a cardinality operator $|\mathbf{p}| = n$ that denotes the length of the path.

Given the transition matrix $\mathbf{T}$ we define the probability of such a path:

$$P(\mathbf{p}) = \prod_{i=0}^{n-1} P(s_{i+1} = p_{i+1} | s_i = p_i) = \prod_{i=0}^{n-1} T_{p_i, p_{i+1}}. \tag{5}$$

2) DEFINITION 2 (MOST LIKELY PATH)

Consider any path $\mathbf{p} \in \mathcal{P}_{o,d} = \{p_0, p_1, p_2, \ldots, p_n\}$. By the most likely path $\hat{\mathbf{p}}$ we mean the path that maximizes the probability of observing that path:

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p} \in \mathcal{P}_{o,d}} \{P(\mathbf{p})\} = \arg\max_{\mathbf{p} \in \mathcal{P}_{o,d}} \left\{ \prod_{i=0}^{n-1} T_{p_i, p_{i+1}} \right\}. \tag{6}$$

Optimizing Eq. (6) appears intractable at first glance. However, this can easily be solved with shortest path algorithms such as Dijkstra's algorithm (Dijkstra 1959). We give precise details on how to find this pathway in section c of the appendix.

d. Obtaining a travel time estimate

The most likely path is often a quantity of interest in itself; however, we can also obtain a travel time estimate of this path. The method should be fast and efficient as it should be able to run for large sets of locations quickly. We achieve this by giving a formula to estimate the travel time based directly on the transition matrix.

Consider the path, $\mathbf{p} = \{p_1, \ldots, p_n\}$, from which we aim to estimate the expected travel time. The key consideration this section addresses is that the path is a sequence of unique states, whereas when simulating from a discrete time Markov chain, the chain has a probability of remaining within the same state for multiple time steps. We therefore aim to obtain an estimate of how long the Markov chain takes, on average, to jump between p_i and p_{i+1}, and then aggregate this over the path to form a travel time estimate.

We assume that the only possibility is that the drifter follows the path in which we are interested. So, p_i must be followed by p_{i+1}. Now we use t to index the time of the Markov chain and suppose $s_t = p_i$. We are then interested in the random variable k where $t + k$ is the first time that the process transitions from p_i to p_{i+1}. Note that the only possibility for states $\{s_{t+l}\}_{l=1}^{k-1}$ is that they are all p_i, otherwise the object would not be following the path of interest. Therefore, we obtain the distribution of k as follows (see the proof in section d of the appendix):

$$P(s_{t+k} = p_{i+1}, \{s_{t+l} = p_i\}_{l=1}^{k-1} | s_t = p_i, \mathbf{p}\})$$
$$= \frac{T_{p_i, p_{i+1}} T_{p_i, p_i}^{k-1}}{(T_{p_i, p_i} + T_{p_i, p_{i+1}})^k}. \tag{7}$$

Note that if we set $a = T_{p_i, p_i}/(T_{p_i, p_i} + T_{p_i, p_{i+1}})$ in Eq. (7) we get

$$P(s_{t+k} = p_{i+1} | s_t = p_i, \mathbf{p}) = a^{k-1}(1-a), \tag{8}$$

which is the probability distribution function of a negative binomial distribution with success probability a and the number of failures being 1. We denote the random variable for the travel time between p_i and p_{i+1} as k_i. As the negative binomial distribution corresponds to the time until a failure, we are interested in taking one time increment longer than this as we require k_i to be the time that we move from p_i to p_{i+1}, that is, the time of the failure. Therefore, the distribution of k_i exactly follows $k_i - 1 \sim \text{NB}(1, a)$. Also, note that k_i is in units of the chosen Lagrangian cutoff time $\mathcal{T}_L$.

To get the expectation of the total Lagrangian travel time we consider the sum of all of the individual parts of the travel times $\mathbf{k} = \sum_{i=0}^{n-1} k_i$, such that we obtain

$$\mathbb{E}[\mathbf{k}] = \sum_{i=0}^{n-1} \mathbb{E}[k_i] = \sum_{i=0}^{n-1} \left(\frac{T_{p_i, p_i}}{T_{p_i, p_{i+1}}} + 1 \right), \tag{9}$$

where we have used that the expectation of the negative binomial (NB) is $\mathbb{E}[x \sim \text{NB}(1, a)] = a/(1-a)$.

We could attempt to obtain a simple variance estimate for the estimate $\mathbb{E}[\mathbf{k}]$ with classical statistics. However, we would only be able to account for variability within the estimates of the entries of the transition matrix, because we would have to assume $\mathbf{p}$ is known. In our case we are interested in the time of $\hat{\mathbf{p}}$, which is itself an estimate as it depends on the transition matrix. Obtaining any analytical uncertainty in the estimation of the most likely path would be intractable due to the complexity of the shortest path algorithm. Therefore, we propose to address the issue of uncertainty in $\mathbb{E}[\mathbf{k}]$ and $\mathbf{p}$ due to data randomness in section 4b using the nonparametric bootstrap. To finish this section, we provide the pseudocode for our approach in algorithm 1:

Input: Drifter dataset $\mathbf{y}$, a set of locations $\mathbf{x}$, Lagrangian cutoff time $\mathcal{T}_L$
Map all of the drifter locations $\mathbf{y}$ to their grids $g_{j,i} = f(\mathbf{y}_{j,i})$ using the map from Eq. (2).
Map all of the locations of interest to their grids $g^{x_i} = f(x_i)$.
Form transition matrix $\mathbf{T}$ using Eq. (4).
For each unique pair o and d in $\{g^{x_i}\}_{x_i \in \mathbf{x}}$ *do*
  Find and store the most likely path $\hat{\mathbf{p}}_{o,d}$ by optimizing Eq. (6).
  Using this path, find and store the expected travel time $\mathbb{E}[\hat{\mathbf{k}}_{o,d}]$ using Eq. (9).
End
Result: Travel times $\mathbb{E}[\hat{\mathbf{k}}_{o,d}]$ for every pair of locations in $\mathbf{x}$ and a corresponding path $\hat{\mathbf{p}}_{o,d}$ given as a sequence of grid indices in $\mathcal{S}$.

Algorithm 1: Pseudocode that summarizes how section 3 is used to turn drifter data and a spatial index function into most likely paths and travel time estimates.

4. Stability and uncertainty

a. Random rotation

A key consideration is that the final results of the algorithmic approach may strongly rely on the precise grid system f chosen in Eq. (2). To address the uncertainty due to the discretization we propose to *randomly sample* a new grid system then run the algorithm for the new grid system. In a simple 2D square grid one could sample a new grid system by sampling two numbers between 0 and the length of a side of the square, then shifting the square by these sampled amounts in the x and y direction. In global complicated grid systems such as the one we consider here proposing uniform random shifting is not trivial.

Rather than trying to reconfigure the grid system, instead we suggest a more universal alternative. We suggest randomly rotating the longitude–latitude locations of all the relevant data using random rotations. Such a strategy will work for any spatial grid system as it just involves a prepossessing step of transforming all longitude–latitude coordinates.[1] Note that for each rotation we are required to reassign the points to the grid and reestimate the transition matrix. These are the two most computationally expensive procedures of the method. To generate the random rotations we use the method suggested by Shoemake (1992). In summary, it amounts to generating 4 random numbers on a unit four-dimensional hypersphere as the quaternion representation of the three-dimensional rotation, which can equivalently be represented as a rotation matrix $\mathbf{M}$. Then we apply this rotation to the Cartesian representation of longitude and latitude.

To obtain travel times that remove bias effects from discretization, we sample n_{rot} rotation matrices $\mathbf{M}^{(i)}$. We then run algorithm 1; however, as a preprocessing step we rotate all locations of the drifter trajectories and locations of interest. For each rotation matrix this will result in a set of travel times $\hat{d}^{(i)}$. The sample mean of these rotations will be more stable than the vanilla method. The sample standard deviation will inform us about uncertainty in travel times due to discretization.

b. Bootstrap

If we required a rough estimate of uncertainty we could consider that $\hat{\mathbf{p}}$, the most likely path, is fixed and then estimate $\text{Var}[\hat{\mathbf{k}}]$. However, this would be a poor estimate because such an estimate would assume that 1) the transition matrix entries follow a certain distribution, and 2) the path $\hat{\mathbf{p}}$ is the true most likely path. In reality neither of these are true, they will both just be estimates. The transition matrix elements are estimated from limited data and the shortest path strongly depends on the estimated transition matrix; e.g., a small change in the transition matrix could result in a significantly different path. Therefore, we obtain estimates of uncertainty by bootstrapping (Efron 1993).

Bootstrapping is a method to automate various inferential calculations by resampling. Here the main goal is to estimate uncertainty around $\hat{\theta} = \mathbb{E}[\hat{\mathbf{k}}]$. The bootstrap involves first resampling from the original drifters to obtain a new dataset. We call $\mathbf{y}^* = \{\mathbf{y}_j^*\}_{j\,=\,1,\ldots,N}$ a bootstrap sample, where $\mathbf{y}_j^*$ is a drifter trajectory that has been sampled with replacement from the original N drifters. Then we use $\mathbf{y}^*$ as the input dataset to algorithm 1.

We do this resampling B times to obtain B estimates of $\hat{\theta} = \mathbb{E}[\hat{\mathbf{k}}]$; we denote these bootstrap estimates as $\{\hat{\theta}^{(b)}\}_{b=1}^{B}$. We then estimate our final bootstrapped mean and standard deviation estimates as the following:

$$\text{sd}^2_{\text{boot}} = \left\{\frac{\sum_{b=1}^{B}[\hat{\theta}^{(b)} - \hat{\theta}^{(\cdot)}]^2}{B-1}\right\},$$

where

$$\hat{\theta}^{(\cdot)} = \sum_{b=1}^{B}\hat{\theta}^{(b)}/B. \tag{10}$$

In addition to the uncertainty measure in travel time that both the bootstrap and rotation methods provide, these methods also supply a collection of sample most likely paths. These paths can be used to investigate various phenomena, such as why the uncertainty is high. We can plot the paths for a fixed origin–destination pair and may see for example that many paths follow one current where another collection of paths follow a different current. We give numerous examples of this in sections 5b and 5c.

5. Application

We use the locations given in Table 1 for the demonstration of the method described in this paper. These locations were chosen for multiple reasons; 1) they were placed on or near ocean currents, such as the South Atlantic Current, the Equatorial Current, and the Gulf Stream, the magnitudes of which can be seen in Fig. 1, and 2) stations were placed in both the North and South Atlantic Ocean to show how the method can find pathways that are not trivially connected. First, we go over an application of the vanilla method from section 3, and then in sections 5b and 5c we respectively provide brief results that use the adaptions using bootstrap and rotations that are described in section 4. In section a of the appendix, we supply a link to a Python package and code used to create these results.

Prior to our analysis we take a practical step to improve the reliability of the method. we find the states corresponding to −79.7°, 9.07°, −80.73°, 8.66° (two points on the Panama landmass), −5.6°, 36°, and −5.61°, 35.88° (two points on the Strait of Gibraltar) and then remove the corresponding rows and columns from $\mathbf{T}$. If this step is not taken the method often uses pathways crossing the Panama landmass, resulting in impossibly short connections to the Pacific Ocean. The reasoning for removing the points on the Strait of Gibraltar is data driven; further details are in the online supplemental material, particularly how one can adapt the method to specify travel times into and out of the Mediterranean Sea.

Figure 4 shows the pathways between a representative sample of the stations. First we note what features are observed

[1] Conditional on the grid system having a reasonable minimum area. This method rotates the poles to a random point, which would give spurious results in a longitude–latitude grid—thus providing another reason why the H3 system is more suitable.

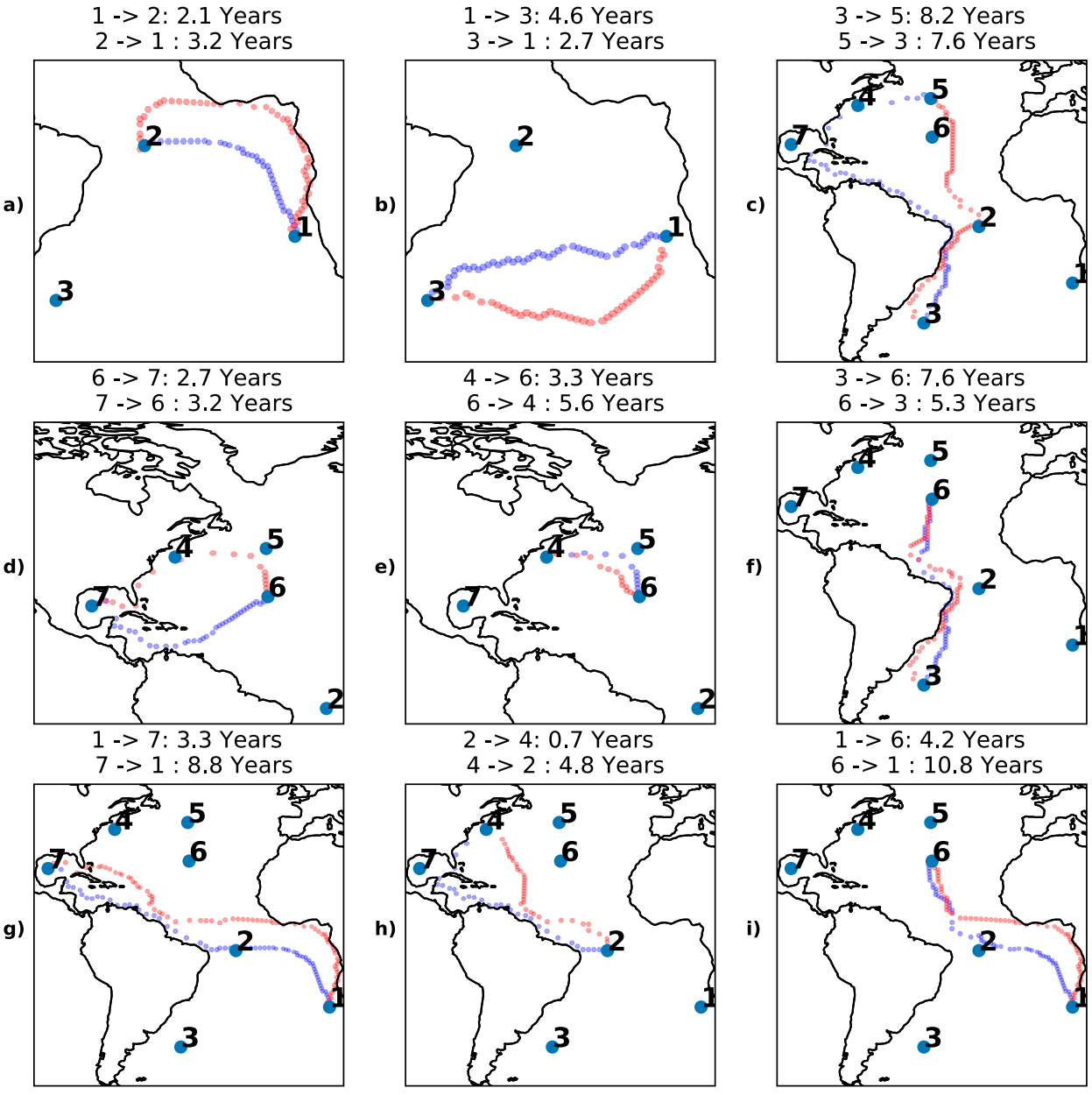

FIG. 4. Example pathways found from the method. Sequences of blue hexagons are going from the lower number to the higher number. Sequences of red hexagons are going from the higher number to the lower number. Numbered locations are as in Table 1. The expected travel time of the most likely path is given in the title of each plot. Similar plots can be provided for every location pair using the online code; however, these are not presented here owing to page-length considerations.

in the most likely path. The Gulf Stream is used on almost every path trying to access locations 4, 5, or 6 in Fig. 4. Observe in Fig. 4c when going from location 3 to 5 that the method chooses to enter the Gulf of Mexico and then uses the Gulf Stream to access location 5, even though the actual geodesic distance of this path is long. Other examples that use the Gulf Stream include Figs. 4d and 4h. Generally, any of the paths leaving location 1 and attempting to travel northwest use the Benguela Current—for example, Figs. 4a, 4g, and 4i.

The travel times obtained between the sample stations in Fig. 4 show interesting results with regard to the lack of symmetry when reversing the direction of the path between two stations. When going from location 2 to location 4 we estimate a long most likely path in terms of physical distance. However, the resulting travel time of this path (0.7 yr) is smaller than the travel time of the more direct path from location 4 to location 2 (4.8 yr)—which is much shorter in distance. This is because the path going from location 2 to location

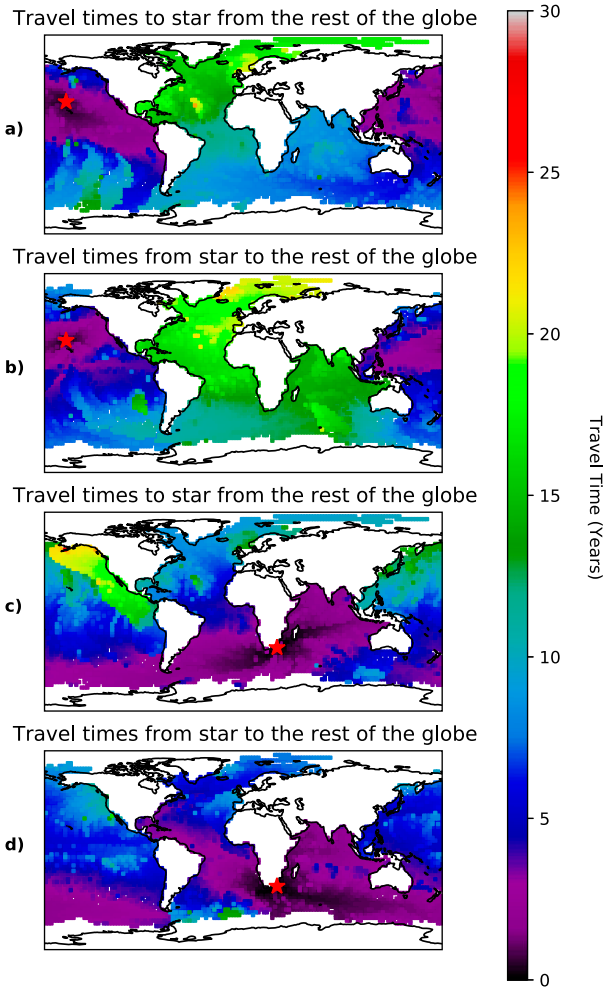

FIG. 5. Travel times of the most likely path originating from the red stars and going to or from (indicated by the title) the centroid of a 2.5° × 2.5° square grid system. Figure setup and locations taken to match Fig. 2 of Jönsson and Watson (2016).

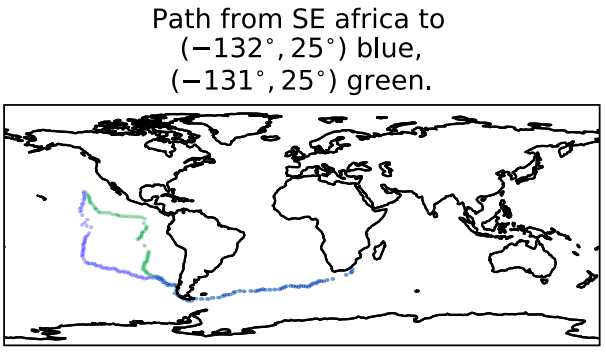

FIG. 6. The most likely path from two points in the North Pacific to the southeast coast of Africa. The green and blue pathways are almost identical as they cross the South Atlantic. The pathways differ greatly, however, as they cross the Pacific, even though the two starting points in the North Pacific are only 1° apart. The path going from −131°, 25° has an expected travel time of 21.2 yr; the path going from −132°, 25° has an expected travel time of 11.4 yr.

4 follows strong currents such as the North Equatorial Current and the Gulf Stream. Another interesting result is that going from 3 to 5 and vice versa are relatively close in terms of travel time even though from 3 to 5 uses the Gulf Stream but the return does not. In the most likely path from 3 to 5, up until around −16° latitude the travel time is 5.2 yr, which we expect as the pathway seems to be going against the Brazil Current. After this point the rest of the path takes the remaining 3 years despite the remainder being over half the actual physical distance of the pathway. We expect this short time is due to the method finding a pathway along the North Brazil Current, followed by the Caribbean Current, followed by the Gulf Stream.

a. Global travel times

Figure 5 shows the travel time distribution to and from two fixed locations, taken to match the studied locations of Jönsson and Watson (2016), to the entire globe. We note that the travel time map is less smooth than the one shown in Jönsson and Watson (2016). The black and purple areas however (up to 5 years of travel time) are similar to those found in Jönsson and Watson (2016), showing agreement over short spatial scales. For larger distances, we generally find that the maps are markedly different. For example, the yellow patch in the northeast pacific in Fig. 5c is not seen in Jönsson and Watson (2016). Such discrepancies can be attributed to many reasons, such as the following: 1) they reflect the difference in methods, where we use a transition matrix approach, and Jönsson and Watson (2016) use a connectivity matrix; 2) Jönsson and Watson (2016) aim to find the shortest path in time, whereas we aim to find the expected time of the most likely path; and 3) the results shown here are derived from real data, whereas Jönsson and Watson (2016) use simulated trajectories.

We show an example in Fig. 6 that explains the lack of spatial smoothness in Fig. 5, where we show two pathways both originating from a fixed point and ending at two distinct points only 1° latitude apart. The points are on either side of the discontinuity in the north-east Pacific seen in Fig. 5c. The pathways become visibly different after they have both reached the South Pacific. Such a phenomenon results in the lack of spatial smoothness of travel time distributions. This demonstrates that the travel times do not necessarily obey the triangle inequality. If smoothness is desired, we show an alternative approach in the online supplemental material, in which instead a minimum travel time is the objective, which is then more analogous to the Jönsson and Watson (2016) approach. We argue however that the expected travel time of the most likely path, rather than the minimum travel time, is a more relevant metric for estimating connectivity and Lagrangian distance in applications measuring spatial dependence between points in the ocean.

b. Bootstrap

To show the value of the bootstrap we show the results for one particular pair of stations, the pathway going from location 1 to location 3 and back. The pathways that result from the

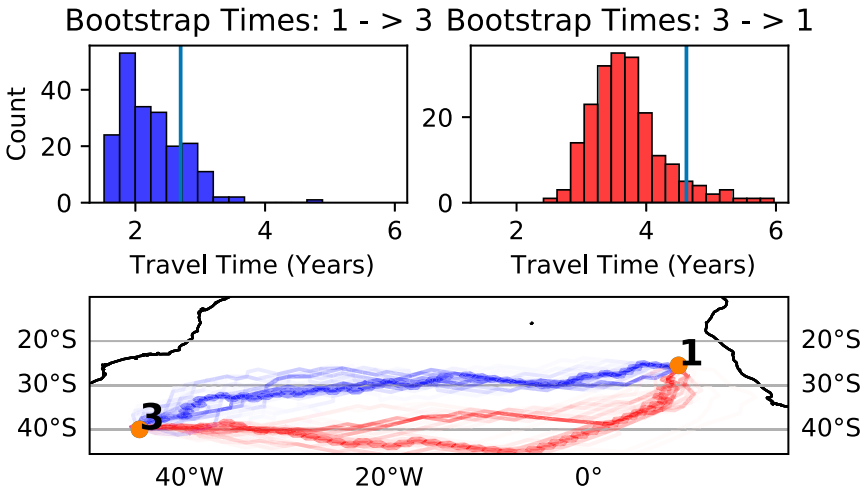

FIG. 7. (top) Two bootstrap distributions of travel times resulting from 200 bootstrap samples. The vertical line is the travel time if the full data are used to estimate the path and time. (bottom) The corresponding bootstrapped paths. Blue lines and hexagons are for going from 1 to 3; red lines and hexagons are for going from 3 to 1. The lines connect the centroids of the spatial index of the bootstrapped paths. Darker lines mean that path is taken more often. The light hexagons are the pathway taken if the full data are used with no resampling, e.g., the pathway shown in Fig. 4.

bootstrap are shown in the bottom panel of Fig. 7. The darker lines on the figure imply that that this transition is used more often. We see that for most of the journey the darker lines closely follow the original path. The bootstrap discovers some slightly different paths, for example around −20° longitude the path going from 3 to 1 occasionally seems to find that going farther south is a more likely path. Also, around the beginning of the path going from 1 to 3, we see that the most likely path taken most frequently by the bootstrap samples often does not follow the most likely path from the full data.

The main goal of the bootstrap is that we obtain an estimate of the standard errors. In this case we get standard error estimates using Eq. (10) of 0.5 yr for going from 3 to 1 and 0.6 yr for going from 1 to 3. In general, we found that the standard error was lower when the path follows the direction of flow. The top row of plots in Fig. 7 appears to show that there is a slight bias between the bootstrap mean and the vanilla method travel time. We believe this is due to the variance within the paths. The mean estimated from the bootstrap samples are close to the estimates from the rotation method we will shortly present (in Fig. 9). The rotation mean estimates are within 0.4 yr of the bootstrap means in both cases shown here.

c. *Rotation*

If we consider two points in the same H3 index, for example location 1 (9°, −25.5°) and a new point 9°, −26.2° (as shown in Fig. 8), then using the original grid system the method will simply produce a travel time of 0. To solve this problem, we consider using 100 rotations as explained in section 5a. For each rotation we estimate the travel time both back and forth. In 22 of the rotations, the two points ended up in the same hexagon, resulting in a zero travel time. We plot the distribution of the other 78 travel times in the bottom row of Fig. 8. The mean of all of the entries including the zeros is 20.5 days for going from the new point to location 1 and 22.2 days for going from location 1 to the new point.

The second benefit of performing rotationsis to make estimates less dependent on the grid system. We use the same 100 rotations as with the previous example and compute the most likely path and the mean travel times. In Fig. 9, we plot the

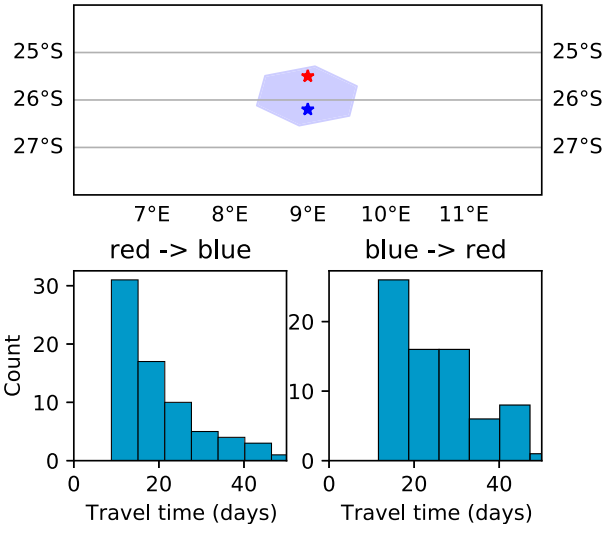

FIG. 8. (top) Plot of location 1 from Table 1 and the point 9°, −26.2°, which is 0.7° south of location 1. The relevant H3 hexagon is plotted over the points. (bottom) The histogram and density estimate of the travel times in each direction from applying 100 rotations. The 22 zeros for when the two locations are in the same hexagon are not included in the histogram.

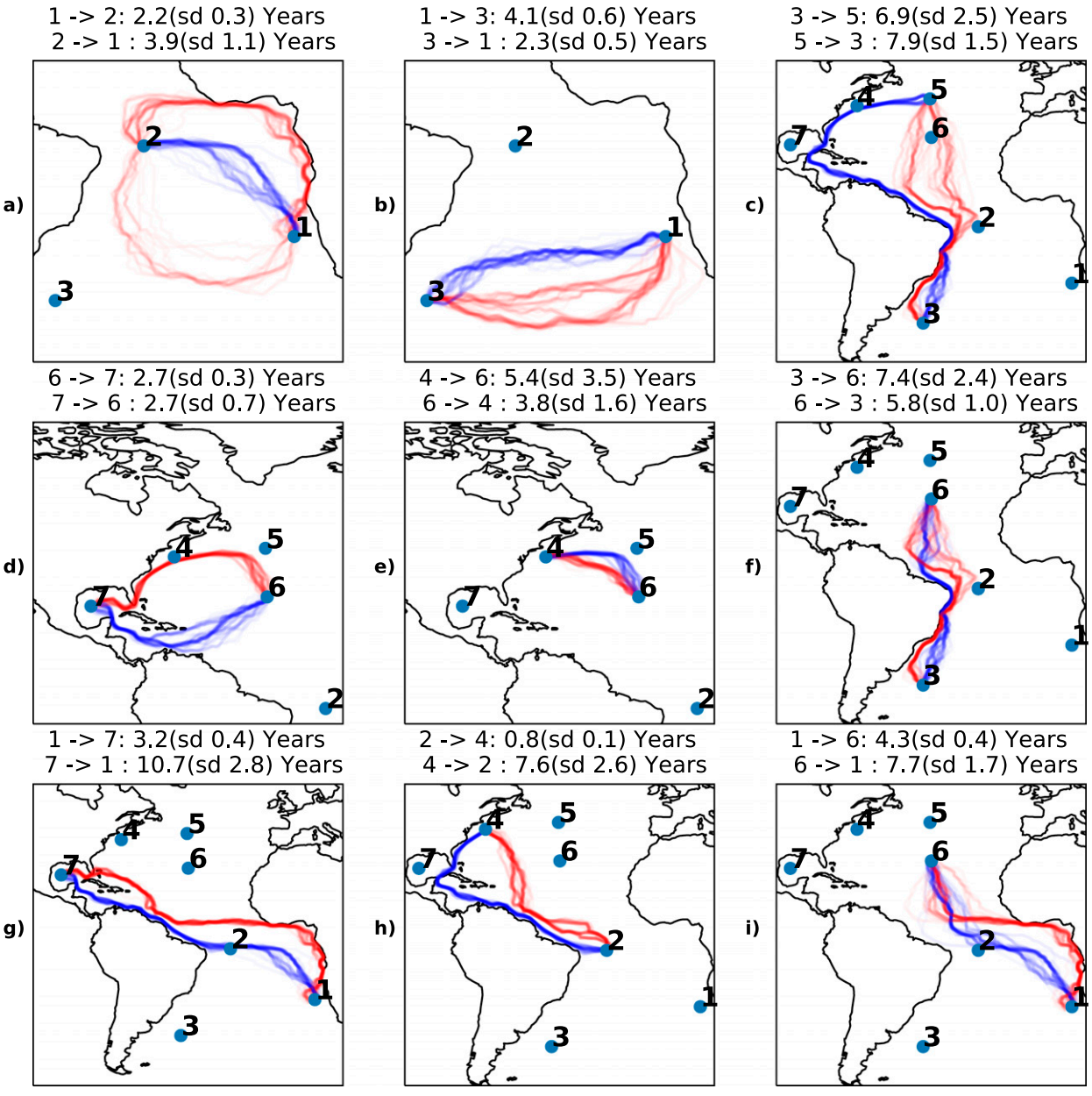

FIG. 9. This figure layout is the same as in Fig. 4, except here we plot paths resulting from 100 random rotations. Each line connects the centroid of each hexagon within the path. Note that the hexagons now come from rotated grid systems, so the centroids could be at any location—hence the smooth continuous-looking lines. The lines are plotted with transparency; when multiple lines overlap these lines will look darker. Standard deviations of the travel times of the 100 paths are reported in the title of each figure.

pathways with the mean and standard deviation of the travel times resulting from these 100 rotations. The travel times and paths shown in this figure are comparable to those given in Fig. 4. In most of the pathways we see that the darkest, most popular paths match up with the pathways in Fig. 4.

One of the more interesting results from this analysis is the path going from 2 to 1 in Fig. 9a. Most of the paths go up closer to the equator, then use the Equatorial Countercurrent, followed by the Guinea and Gulf of Guinea Currents as in the original vanilla application of the method. A small number of the rotations result in pathways that end up crossing the South Atlantic, to the south of location 2, then follows the South Atlantic Current over to location 1.

In general, the travel times from the rotation and original method can be significantly different, which supports the need for this rotation method. If we compare Figs. 4 and 9, most of the distances stay close to what they were in the original results using no rotations. We see that going from 6 to 4 drops from 5.6 yr in Fig. 4e to 3.8 yr in Fig. 9e and from 4 to 6 increases from 3.3 to 5.4 yr. This causes the ordering of the distances to change

as from 6 to 4 is now the shorter travel time. We believe the case in e) is mainly due to 4 being located just northwest of the stronger currents of the Gulf Stream, which makes it sensitive to the grid system. However, the high standard errors in Fig. 9 suggest we are uncertain about this travel time.

6. Discussion and conclusions

In contrast to van Sebille (2014), our method as presented does not take into account seasonality. We have a few ideas for how seasonality could be incorporated in future work. An obvious adaptation, if the aim was to obtain a short travel time that is expected to lie in a small 3-month window, is to just estimate $\mathbf{T}$ using drifter observations that are in that time window. Alternatively, we could use $\mathcal{T}_L$ to be a certain jump such as a gap of two months, then we estimate six transition matrices, say $\mathbf{T}^{(k)}$, where the entries $T^{(k)}_{i,j}$ are probabilities of going from the previous time period at state i to state j at the current time. Such a set up could still be solved using our shortest path algorithm. We justify our approach in the same way as Maximenko et al. (2012): we aim to provide a global view and a simple general concept explaining the pattern of potential pathways and travel times. The base method can then be adapted by practitioners to account for local spatial or temporal considerations.

More results demonstrating the robustness of our method, along with motivation of parameter choices, can be found in the online supplemental material. A key finding that we discuss here is that we found the size of the grid system affects the estimated travel times significantly, regardless of whether the latitude–longitude or the H3 grid system is used. Therefore, we do not recommend comparing travel times obtained from two different grid sizes. In general, the results are correlated in an order comparison sense; however, their magnitudes change. Typically, a smaller grid system results in shorter travel times. Because of this we would only advise the results to be used in relative comparison to each other, for example by saying that the travel time from a to b is 2 times that than from b to c, where both times are obtained with the same grid system. The choice to show resolution 3 in this paper was found to perform robustly (balancing the error from discretization and data sparsity) and follows grid sizes that approximately match previous works where $1° \times 1°$ grids are used, but this can be changed easily in the online package.

The use of the bootstrap and rotations are relatively easy methods to implement, each of which provides effective estimates of uncertainty from data uncertainty and discretization, respectively. However, combining these procedures into one requires careful consideration. If we wanted to run n_{rot} rotations and B bootstraps for each rotation, we still require a method to combine these estimates of travel times. We could treat every rotation equivalently, so say that our bootstrap sample in Eq. (10) is all $n_{\text{rot}} \times B$ samples to obtain an estimate of uncertainty in travel time due to the combination of grid discretization and data randomness. Additionally, we could decompose the uncertainty and provide a standard error for just the data randomness by estimating a standard error for each rotation using just the B samples in each rotation, and then taking the average of all n_{rot} standard error estimates.

Our choice of the Lagrangian decorrelation time of 5 days may be too low in some instances. Previous works have found correlations in the velocity data lasting longer than 5 days in certain regions (Lumpkin et al. 2002; Zhurbas and Oh 2004; Elipot et al. 2010). This may suggest that using a larger value for $\mathcal{T}_L$ may be needed to justify the Markov assumption. The trade-off however is resolution, where shorter time scales allow pathways and distances to be computed with more detail. Our method is designed flexibly such that the practitioner can pick the most appropriate time scale for the spatial region and application of interest.

In general, some unexpected features of the method do occur such as the discontinuity discussed in section 5a. We expect there would be less of a discontinuity if these times were computed with the rotation method; however, we argue that the discontinuities with travel times of most likely pathways should always exist. If smoothness of travel times was a major requirement, then one could consider the *shortest* path in travel time rather than the *most likely* path. We briefly show this adaptation in the online supplemental material. We expect the results would require more careful checking in such an approach, as the shortest path would be more likely to use any glitches in the grid system such as if there was a connection over Panama.

To summarize, in this paper we have created a novel method to estimate Lagrangian pathways and travel times between oceanic locations, thus offering a new, fast, and intuitive tool to improve our knowledge of the dynamics of marine organisms and oceanic transport and global circulation.

Acknowledgments. The work of M. O'Malley was funded by the Engineering and Physical Sciences Research Council (Grant EP/L015692/1). The work of A. M. Sykulski was funded by the Engineering and Physical Sciences Research Council (Grant EP/R01860X/1).

Data availability statement. The drifter data were provided by the Global Drifter Program (Lumpkin and Centurioni 2019). The currents used for visualization purposes in Fig. 1 are V3.05 of the dataset supplied on the Global Drifter Program website (Laurindo et al. 2017).

APPENDIX

Additional Material

a. Package

Code to reproduce all figures related to the method is available online (https://github.com/MikeOMa/MLTravelTimesFigures). The Python package implementing all of the methods in this paper alongside an interactive demonstration can also be found online (https://github.com/MikeOMa/DriftMLP). The package takes roughly 3 min total to go from raw data to a pairwise travel time matrix for the locations shown in Table 1 using algorithm 1.

b. Table of notation

We include a table of mathematical notation for reader reference in Table A1.

TABLE A1. Table of mathematical notation.

$P(x\|y)$	Denotes the probabilities of event(s) x given that y occurs
$\mathbb{E}[x]$	The expectation of x
$f(s)$	The discretization function, e.g., H3
$\mathbb{I}[x]$	Indicator function giving 1 if x is true and 0 otherwise
$\arg\max_{x\in S}$	An operator that gives the input value, which maximizes the function q, restricted to the set S
$\mathbf{T}$, $T_{i,j}$	$\mathbf{T}$ denotes transition matrix, with entries $T_{i,j}$, $i, j \in \mathcal{S}$, denoting the probability of moving from state i to j in $\mathcal{T}_L$ days
$x^\circ \times y^\circ$	Refers to a longitude–latitude grid system, x degrees in the longitudinal direction, y degrees in the latitudinal direction
$\mathcal{T}_L$	Lagrangian cutoff time
$\mathcal{S}$	The set of all possible spatial indices
$\mathcal{P}_{o,d}$	The set of all possible paths going from o to d
$\mathbf{p} = \{p_i\}_{i=1}^n$	A pathway of length n; indicates a sequence $p_1, p_2, \ldots, p_n$; all $p_i \in \mathcal{S}$
$\mathbf{k}$	The expected travel time of a path $\mathbf{p}$
$\hat{\mathbf{p}}$, $\hat{\mathbf{k}}$	Caret notation implies that we are considering the most likely path and travel time of that path, respectively
s_t	Used to index the state of the Markov chain after t steps

c. *Finding the shortest path*

To solve the optimization of Eq. (6), we can equivalently consider the logarithm of $P(\mathbf{p})$:

$$\log P(\mathbf{p}) = \sum_{i=0}^{n-1} \log T_{p_i,p_{i+1}}.$$

Then we use the fact that

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}\in\mathcal{P}_{o,d}}\{\log P(\mathbf{p})\} = \arg\min_{\mathbf{p}\in\mathcal{P}_{o,d}}\{-\log P(\mathbf{p})\}$$

$$= \arg\min_{\mathbf{p}\in\mathcal{P}_{o,d}}\left\{-\sum_{i=0}^{n-1}\log T_{p_i,p_{i+1}}\right\}. \quad \text{(A1)}$$

Now, in this form this equation can be solved using the vast literature on shortest path algorithms.

Shortest path algorithms (Gallo and Pallottino 1988; Dijkstra 1959), such as Dijkstra's algorithm, are popular algorithms that find the so-called shortest path within a graph. In our case the graph is formed such that the vertices or nodes uniquely correspond to a grid system index, that is, a row/column in the transition matrix $\mathbf{T}$. If there is a nonzero probability in $T_{i,j}$ we add an edge denoted $e_{i,j}$, where the weight on this edge is denoted $w(e_{i,j}) = -\log(T_{i,j})$ between the vertex i and going to the vertex j. Note that $T_{i,j}$ is not necessarily the same as $T_{j,i}$; hence, we have a *directed graph*. Given a start vertex o and an end vertex d, shortest path algorithms will find the path $P = \{v_1 \ldots, v_n\}$ such that P minimizes the following:

$$\sum_{i=1}^{n-1} w(e_{v_i,v_{i+1}});$$

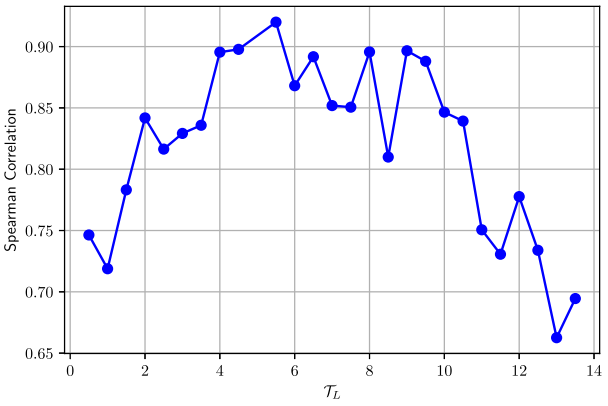

FIG. A1. Spearman correlation coefficient between the nondiagonal elements of the travel time matrix generated by $\mathcal{T}_L = 5$ and the matrices generated by the values of $\mathcal{T}_L$ on the x axis.

hence, it solves the problem in Eq. (A1). The algorithm used is exact; hence, if no path is found then no path exists given the current network.

d. *Derivation of Eq. (7)*

The derivation uses the Markov property, the conditional probability definition, and the fact that $P(x \in \{a, b\}) = P(x = a) + P(x = b)$:

$$\begin{aligned}
&P(s_{t+k} = p_{i+1}, \{s_{t+l} = p_i\}_{l=1}^{k-1} | s_t = p_i, \mathbf{p}\}) \\
&\quad = P(s_{t+k} = p_{i+1} | s_{t+k-1} = p_i, s_{t+k} \in \{p_i, p_{i+1}\}) \\
&\qquad \times \prod_{l=1}^{k-1} P(s_{t+l} = p_i | s_{t+l-1} = p_i, s_{t+l} \in \{p_i, p_{i+1}\}) \\
&\quad = \frac{P(s_{t+k} = p_{i+1} | s_{t+k-1} = p_i)}{P(s_{t+k} \in \{p_i, p_{i+1}\} | s_{t+k-1} = p_i)} \\
&\qquad \times \prod_{l=1}^{k-1} \frac{P(s_{t+l} = p_i | s_{t+l-1} = p_i)}{P(s_{t+l} \in \{p_i, p_{i+1}\} | s_{t+l-1} = p_i)} \\
&\quad = \frac{P(s_{t+k} = p_{i+1} | s_{t+k-1} = p_i)}{P(s_{t+1} \in \{p_i, p_{i+1}\} | s_t = p_i)^k} \\
&\qquad \times \prod_{l=1}^{k-1} P(s_{t+l} = p_i | s_{t+l-1} = p_i) \\
&\quad = \frac{T_{p_i,p_{i+1}} T_{p_i,p_i}^{k-1}}{(T_{p_i,p_i} + T_{p_i,p_{i+1}})^k},
\end{aligned}$$

where the first equality follows from the explanation given in section 3d.

e. *Brief sensitivity analysis to cutoff time*

The main tuning parameter that we have fixed in this paper is the Lagrangian cutoff time used when estimating the transition matrix $\mathbf{T}$. The method is not especially sensitive to this choice, as we shall now demonstrate. To show the sensitivity we ran an experiment in which for a grid of values for $\mathcal{T}_L$ we estimated a pairwise travel time matrix for the locations in Table 1 and then estimated the Spearman correlation coefficient between the nondiagonal entries of

each matrix to the corresponding entry of the travel time matrix generated from $\mathcal{T}_L = 5$. Results are shown in Fig. A1. The experiment shows that the distances change but that overall the matrices are very strongly correlated, particularly for $\mathcal{T}_L > 2$. For comparison, the average correlation value between the pairwise travel time matrix $\mathcal{T}_L$ and the travel time matrices generated from the 100 rotations used in section 5c is 0.8. A similar analysis that considers sensitivity to grid sizes is given in the online supplemental material.

REFERENCES

Becking, L. E., D. F. Cleary, N. J. de Voogd, W. Renema, M. de Beer, R. W. van Soest, and B. W. Hoeksema, 2006: Beta diversity of tropical marine benthic assemblages in the Spermonde Archipelago, Indonesia. *Mar. Ecol.*, **27**, 76–88, https://doi.org/10.1111/j.1439-0485.2005.00051.x.

Berline, L. O., A. M. Rammou, A. Doglioli, A. Molcard, and A. Petrenko, 2014: A connectivity-based eco-regionalization method of the Mediterranean Sea. *PLOS ONE*, **9**, e111978, https://doi.org/10.1371/journal.pone.0111978.

Dijkstra, E. W., 1959: A note on two problems in connexion with graphs. *Numer. Math.*, **1**, 269–271, https://doi.org/10.1007/BF01386390.

Early, J. J., and A. M. Sykulski, 2020: Smoothing and interpolating noisy GPS data with smoothing splines. *J. Atmos. Oceanic Technol.*, **37**, 449–465, https://doi.org/10.1175/JTECH-D-19-0087.1.

Efron, B., 1993: *An Introduction to the Bootstrap. Stat. Appl. Probab. Monogr.*, No. 57, Chapman and Hall, 436 pp.

Elipot, S., R. Lumpkin, and G. Prieto, 2010: Modification of inertial oscillations by the mesoscale eddy field. *J. Geophys. Res.*, **115**, C09010, https://doi.org/10.1029/2009JC005679.

——, ——, R. C. Perez, J. M. Lilly, J. J. Early, and A. M. Sykulski, 2016: A global surface drifter data set at hourly resolution. *J. Geophys. Res. Oceans*, **121**, 2937–2966, https://doi.org/10.1002/2016JC011716.

Ellingsen, K., and J. Gray, 2002: Spatial patterns of benthic diversity: Is there a latitudinal gradient along the Norwegian continental shelf? *J. Anim. Ecol.*, **71**, 373–389, https://doi.org/10.1046/j.1365-2656.2002.00606.x.

Froyland, G., K. Padberg, M. H. England, and A. M. Treguier, 2007: Detection of coherent oceanic structures via transfer operators. *Phys. Rev. Lett.*, **98**, 224503, https://doi.org/10.1103/PHYSREVLETT.98.224503.

——, R. M. Stuart, and E. van Sebille, 2014: How well-connected is the surface of the global ocean? *Chaos*, **24**, 033126, https://doi.org/10.1063/1.4892530.

Gallo, G., and S. Pallottino, 1988: Shortest path algorithms. *Ann. Oper. Res.*, **13**, 1–79, https://doi.org/10.1007/BF02288320.

Hansen, D. V., and P.-M. Poulain, 1996: Quality control and interpolations of WOCE–TOGA drifter data. *J. Atmos. Oceanic Technol.*, **13**, 900–909, https://doi.org/10.1175/1520-0426(1996)013<0900:QCAIOW>2.0.CO;2.

Huntley, H. S., B. Lipphardt Jr., and A. Kirwan Jr., 2011: Lagrangian predictability assessed in the East China Sea. *Ocean Modell.*, **36**, 163–178, https://doi.org/10.1016/j.ocemod.2010.11.001.

Jönsson, B. F., and J. R. Watson, 2016: The timescales of global surface-ocean connectivity. *Nat. Commun.*, **7**, 11239, https://doi.org/10.1038/ncomms11239.

LaCasce, J. H., 2008: Statistics from Lagrangian observations. *Prog. Oceanogr.*, **77**, 1–29, https://doi.org/10.1016/j.pocean.2008.02.002.

Laurindo, L. C., A. J. Mariano, and R. Lumpkin, 2017: An improved near-surface velocity climatology for the global ocean from drifter observations. *Deep-Sea Res. I*, **124**, 73–92, https://doi.org/10.1016/j.dsr.2017.04.009.

Lumpkin, R., and M. Pazos, 2007: Measuring surface currents with surface velocity program drifters: The instrument, its data, and some recent results. *Lagrangian Analysis and Prediction of Coastal and Ocean Dynamics*, A. Griffa et al., Eds., Cambridge University Press, 39–67.

——, and L. Centurioni, 2019: Global Drifter Program quality-controlled 6-hour interpolated data from ocean surface drifting buoys. NOAA/National Centers for Environmental Information, accessed 20 January 2020, https://doi.org/10.25921/7ntx-z961.

——, A.-M. Treguier, and K. Speer, 2002: Lagrangian eddy scales in the northern Atlantic Ocean. *J. Phys. Oceanogr.*, **32**, 2425–2440, https://doi.org/10.1175/1520-0485-32.9.2425.

——, L. Centurioni, and R. C. Perez, 2016: Fulfilling observing system implementation requirements with the global drifter array. *J. Atmos. Oceanic Technol.*, **33**, 685–695, https://doi.org/10.1175/JTECH-D-15-0255.1.

Maximenko, N., J. Hafner, and P. Niiler, 2012: Pathways of marine debris derived from trajectories of Lagrangian drifters. *Mar. Pollut. Bull.*, **65**, 51–62, https://doi.org/10.1016/j.marpolbul.2011.04.016.

McAdam, R., and E. van Sebille, 2018: Surface connectivity and interocean exchanges from drifter-based transition matrices. *J. Geophys. Res. Oceans*, **123**, 514–532, https://doi.org/10.1002/2017JC013363.

Meehl, G. A., 1982: Characteristics of surface current flow inferred from a global ocean current data set. *J. Phys. Oceanogr.*, **12**, 538–555, https://doi.org/10.1175/1520-0485(1982)012<0538:COSCFI>2.0.CO;2.

Miron, P., F. J. Beron-Vera, M. J. Olascoaga, J. Sheinbaum, P. Pérez-Brunius, and G. Froyland, 2017: Lagrangian dynamical geography of the Gulf of Mexico. *Sci. Rep.*, **7**, 7021, https://doi.org/10.1038/s41598-017-07177-w.

——, ——, ——, and P. Koltai, 2019: Markov-chain-inspired search for MH370. *Chaos*, **29**, 041105, https://doi.org/10.1063/1.5092132.

Rypina, I. I., D. Fertitta, A. Macdonald, S. Yoshida, and S. Jayne, 2017: Multi-iteration approach to studying tracer spreading using drifter data. *J. Phys. Oceanogr.*, **47**, 339–351, https://doi.org/10.1175/JPO-D-16-0165.1.

Ser-Giacomi, E., V. Rossi, C. López, and E. Hernández-García, 2015a: Flow networks: A characterization of geophysical fluid transport. *Chaos*, **25**, 036404, https://doi.org/10.1063/1.4908231.

——, R. Vasile, E. Hernández-García, and C. López, 2015b: Most probable paths in temporal weighted networks: An application to ocean transport. *Phys. Rev.*, **92E**, 012818, https://doi.org/10.1103/PHYSREVE.92.012818.

Shoemake, K., 1992: Uniform random rotations. *Graphics Gems III*, Elsevier, 124–132.

Smith, T. M., and Coauthors, 2018: Rare long-distance dispersal of a marine angiosperm across the Pacific Ocean. *Global Ecol. Biogeogr.*, **27**, 487–496, https://doi.org/10.1111/geb.12713.

Sykulski, A. M., S. C. Olhede, J. M. Lilly, and E. Danioux, 2016: Lagrangian time series models for ocean surface drifter trajectories. *J. Roy. Stat. Soc.*, **65**, 29–50, https://doi.org/10.1111/rssc.12112.

UBER, 2019: H3 spatial index. Accessed 8 January 2020, https://eng.uber.com/h3/.

Ulam, S. M., 1960: A Collection of Mathematical Problems. Interscience Tracts in Pure and Applied Mathematics, Vol. 8, Interscience Publishers, 150 pp.

van Sebille, E., 2014: Adrift.org.au—A free, quick and easy tool to quantitatively study planktonic surface drift in the global ocean. *J. Exp. Mar. Biol. Ecol.*, **461**, 317–322, https://doi.org/10.1016/j.jembe.2014.09.002.

——, P. van Leeuwen, A. Biastoch, C. Barron, and W. de Ruijter, 2009: Lagrangian validation of numerical drifter trajectories using drifting buoys: Application to the Agulhas system. *Ocean Modell.*, **29**, 269–276, https://doi.org/10.1016/j.ocemod.2009.05.005.

——, L. M. Beal, and W. E. Johns, 2011: Advective time scales of Agulhas leakage to the North Atlantic in surface drifter observations and the 3D OFES model. *J. Phys. Oceanogr.*, **41**, 1026–1034, https://doi.org/10.1175/2010JPO4602.1.

——, M. H. England, and G. Froyland, 2012: Origin, dynamics and evolution of ocean garbage patches from observed surface drifters. *Environ. Res. Lett.*, **7**, 044040, https://doi.org/10.1088/1748-9326/7/4/044040.

——, and Coauthors, 2018: Lagrangian ocean analysis: Fundamentals and practices. *Ocean Modell.*, **121**, 49–75, https://doi.org/10.1016/j.ocemod.2017.11.008.

Wakata, Y., and Y. Sugimori, 1990: Lagrangian motions and global density distributions of floating matter in the ocean simulated using shipdrift data. *J. Phys. Oceanogr.*, **20**, 125–138, https://doi.org/10.1175/1520-0485(1990)020<0125:LMAGDD>2.0.CO;2.

Watson, J. R., 2018: The geography of the world's oceans explains patterns of planktonic diversity. *2018 Ocean Sciences Meeting*, Portland, OR, Amer. Geophys. Union.

White, C., K. A. Selkoe, J. Watson, D. A. Siegel, D. C. Zacherl, and R. J. Toonen, 2010: Ocean currents help explain population genetic structure. *Proc. Roy. Soc.*, **277B**, 1685–1694, https://doi.org/10.1098/rspb.2009.2214.

Zhurbas, V., and I. S. Oh, 2004: Drifter-derived maps of lateral diffusivity in the Pacific and Atlantic Oceans in relation to surface circulation patterns. *J. Geophys. Res.*, **109**, C05015, https://doi.org/10.1029/2003JC002241.