UCLouvain

epl

École polytechnique de Louvain

Innovation by thinking inside the box

Detecting communities on the surface of the North Atlantic Ocean

Author: **Asmaa SEMMAH**
Supervisors: **Eric DELEERSNIJDER, Jean-Charles DELVENNE**
Readers: **Emmanuel HANERT, Jonathan LAMBRECHTS**
Academic year 2021–2022
Master [120] in Data Sciences Engineering

Abstract

The goal of this master's thesis is to cluster accurately the surface of the North Atlantic Ocean. For that, we constructed a Lagrangian flow network modeling the exchange of buoyant particles between subareas or boxes. Once we had our model, we selected two methods of clustering that suited well the particularities of the flow network : Infomap and Constant Potts model. We chose parameters for the clustering methods that gave us good quality metrics and then applied them onto the flow network. An extensive comparison of the clustering solutions is then done in order to assess the limitations of the two methods, in particular the resolution limit and the field-of-view limit.

Acknowledgments

I would like to thank my supervisors Eric Deleersnijder and Jean-Charles Delvenne
for their patience and kindness,
Alain, Gaëlle and Noëlie for their support and all the happy memories,
Antoine for his encouragements, his help and all the laughs,
Imane for being the best sister one could hope for,
and all the wonderful people in my life that believed in me in difficult times.

Contents

Chapter 1

Introduction - *On the surface*

The goal of this master's thesis is to explore the connectivity of an oceanic domain in its surface in order to subdivide it into *clusters*, *communities*, *provinces*, the meaning of the previous words being equivalent in this report. In order to answer this main research question, others arise :

1. *Why do we study community detection in the oceanic context ?*
2. *How do you model the surface of an ocean ?*
3. *What is the meaning of a cluster in this context ?*
4. *How to accurately detect clusters ?*
5. *How should we judge the quality of the clustering solutions ?*
6. *What are the persistently found structures ?*

These questions precisely delimit the content of this dissertation.

1.1 Structure of the thesis

In the Introduction (chapter 1) you are currently reading, we will address the first question with an overview of similar work. A short description of each section and a description of the data used and why we chose it is also included.
The second chapter (chapter 2) is dedicated to the second question where we will construct what is called a *Lagrangian flow network*. This graph has the particularity to be varying with time, weighted and directed. This make the study of clustering harder, most algorithms being designed for undirected ones.
In the third chapter (chapter 3), an overview of the community detection techniques will be given, justifying the use of *Infomap* and *Constant Potts model* respectively used in chapter 4 and chapter 5.
Chapter 6, is dedicated to assess the solutions found for the community detection problem and determining the structure that are frequently found.

Finally, we will conclude on the challenges encountered, the main takeaways and the perspective of improvements of this work (chapter 7).

1.2 Oceanographic context

The study of the connectivity of suspended objects on the surface of the ocean is of great interest : many objects can be modelled as buoyant particles and those particles can be used in Lagrangian ocean analysis, which is the subject of chapter 2. Examples of modelled buoyant particles include but are not restricted to : larvae [22], [17], plastic litter [7], oil droplet [14] and fluid parcels themselves. With the advection of those particles we will be able to construct an adjacency matrix from the exchange of particles between the discretized subdomains. This adjacency matrix is then used to study connectivity and to establish clusters.

Our goal in this master's thesis is to delineate accurately oceanic provinces that are well mixed within themselves, while sparsely connected with other provinces. It could be particularly useful to do complex calculations on that new-found discretized domain. It has been found to be used to predict the zone of accumulation of plastic litter [11] or understanding marine population connectivity in the context of marine ecology [27] but the applications are vast and diverse. Furthermore, the research on the ocean surfaces are essential in order to mitigate human-induced stressors: for instance pollution, overfishing and habitat destruction. The design of Marine Protected Areas (areas where damaging practices are controlled in order to protect populations) is crucial to management and conservation of ecosystems and could use studies on community detection to, for example, increase the chance of *persistence* of marine populations. It goes without saying that oceans play a major role in the regulation of the climate and is the place of life and reproduction of countless species. Global warming and by extension the preservation of ecosystems being the main concern of the century, as evidenced by the increase in research in the field, there are a lot of answers to the question : *"Why do we study community detection on the surface of oceans ?"*.

Work has already been made on the subject, for instance in the Mediterranean basin [17], approximated to be a closed domain. More recently, communities have been detected in the Arctic[15], which is an open domain. Both of the previous papers used the algorithm *Infomap*, Thomas et al. 2015 [22] however used *Potts model* to detect communities. A chapter will be dedicated to the choice of algorithms as it is a crucial one : the oceanic domain being subjected by greatly varying time scales and by extension spatial scales, the clustering problem is rather a challenging one.

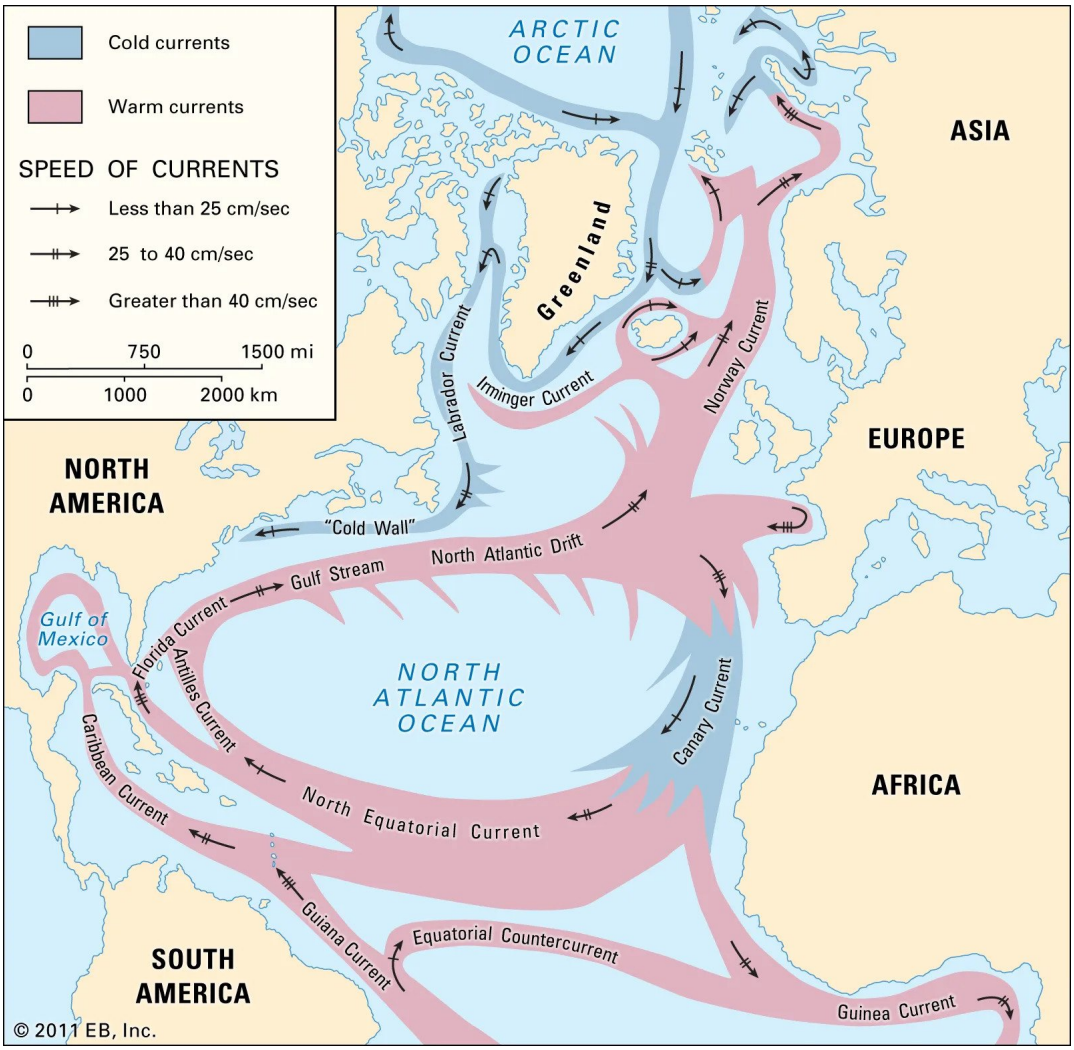


Figure 1.1: Illustration of the domain of interest and its major currents taken from *Encyclopedia Britannica*

In this study, we will design our clusters on the North Atlantic Ocean. As any ocean in contact with humans, it suffers from marine litter, tending to accumulate at the center of gyres and coastlines and containing the infamous *North Atlantic garbage patch*.

1.3 Overview of thesis

1.3.1 Lagrangian ocean analysis

The next question to answer is *"How do we model the surface of the North Atlantic Ocean ?"*. In that regard, our first objective will be to construct a Lagrangian flow network. A Lagrangian flow network will consider each subarea of the domain as a node and study the connectivity with each pair of subareas by analyzing the exchange of Lagrangian particles. The first step is consequently to discretize the domain while assuring that the subdomains have roughly the same area, so the exchange of particles between the nodes are comparable. Lagrangian particles are any buoyant and passive particles, in that regard they are particularly useful to model the surface of the ocean, which is our domain of interest. We simulate ocean pathways of many suspended objects and the connectivity will be the (normalized) exchanges of those objects between the different subdomains.

Because of the definition of Lagrangian particles, we can model larvae dispersal in a Lagrangian flow network. Many species disperse in a largely passive manner, and those that do not can be modelled as such by neglecting their capacity of orienting themselves and their vertical movements. Other examples of Lagrangian particles are plastic debris and fluid parcels themselves. However, the tracking of larvae is particularly interesting in the design of MPA because it is essential to the planning of oceanic populations and to determine broad scale connectivity. The challenge is that it can be difficult to predict their movements in terms of pattern and magnitude, and it is essential to plan their persistence in the area we want to protect.

Before the release of the Lagrangian particles we will have to do a proper discretization on our domain of interest. Reijnders et al. 2021 [15] considers the Arctic and selected icosahedral-hexagonal grid to have subareas they call *bins* of comparable shape and size. While Rossi et al. 2014 [17] used quasi-square boxes that allowed them to consider important mesoscale features of the Mediterranean circulation. In our case, the latter choice of grid is relevant because we don't have a significant convergence of meridians, such as in the Arctic. After the discretization of the domain, we chose a sufficient amount of particles to be advected and finally

study the exchange of them between the nodes for two different time steps : 30 and 60 days.

1.3.2 Community detection

From the network obtained - the Lagrangian flow network - we will implement community detection algorithms and the results will be clusters of various sizes and shape. We will call those clusters : *hydrodynamical provinces*. They are no single definition of a community in consequence, they are several types of community detection algorithms such as those using spectral clustering, an information theoric approach like *Infomap*, modularity based like the *Louvain method*. In the research process, we observed that most of similar study to our own used *Infomap* for this purpose. In this section of the thesis, we want to conduct a (non-exhaustive) comparative study of the literature on community detection with a stress on being meaningful in the regard of the definition of a cluster in this context, the goal being to conserve physical interpretation. What is important in the selection of a proper method is to be able to not chose the number of clusters a priori, to have a method capable of detecting largely different sizes of clusters, identifying nested communities, and taking into account the directed nature of the weighted network. The algorithms selected will be tested on the North Atlantic dataset.

1.3.3 Quality of clusters and detection of coherent structure

When the clustering solutions are found, the quality of them will be assessed by having an idea of the *mixing parameter* : how well are my clusters mixed inside ? And the *coherence ratio* : how sparsely are my clusters connected with each other ?

An important remark is that an algorithm can give different clustering solutions at different time steps and initial time. Ideally, we will be able to find clustering boundaries that are persistent. It is relevant to analyze what is variable from a clustering solution to another and what features are consistent. Rossi et al. 2014 [17] finds consistent boundaries coinciding with well-known oceanic features (such as major currents). This part of the work is useful to detect Lagrangian coherent structure or LCSs, that is material lines that are linearly stable or unstable for longer times than surrounding regions. In the literature, there is not yet a consensus on the definition. Some returns partitions of the domain where each structure is maximally coherent or, equivalently, where the exchange of clusters are minimal.

1.4 Hydrodynamical data and model

1.4.1 Testing on Global Ocean Physical Reanalysis

We will use the Global Ocean Physical Reanalysis product by the Copernicus Marine Environment Monitoring Service (CMEMS):
"GLOBAL_MULTIYEAR_PHY_001_030" a global ocean eddy-resolving with 1/12° horizontal resolution and 50 vertical levels. Testing will be done from year 2010 to year 2019, with an emphasis on the most recent complete year available : 2019.

Mercator has been regularly upgrading its product, starting from the GLORYS2V4 which has 1/4° resolution and 75 vertical levels and is eddy-permitting and covering 1993 to 2016, improvements have been made on the ocean model and the assimilation scheme giving an eddy-resolving GLORYS12V1. The reanalysis product is constructed by assimilating an ocean model output from the NEMO platform driven at surface by ECMWF ERA-Interim then ERA5 reanalyses for recent years with current real-time data.

This product includes daily and monthly mean files for temperature, salinity, currents, sea level, mixed layer depth and ice parameters from the top to the bottom. The global ocean output files are displayed on a standard regular grid at 1/12° (approximately 9 km) and on 50 standard levels. In our study, only the first layer of those 50 will be used along with the current's northward and eastward velocities.

The spatial domain is taken to have a latitude ranging from : 0° 56' 9.7" South to 68° 38' 19.4" North and a longitude ranging from 98° 3' 14.1" West to 12° 0' 21.4" East, for illustration, see Figure 1.1.

1.5 Notation and terminology

Throughout this dissertation, we will work on directed and weighted graphs (call interchangeably networks).
We write $G = (V, E, W)$. Where :

- V is the set of nodes (or vertices), the size of the set being $|V| = n$;
- E the set of edges $E \subseteq V \times V$, the size of the set being $|E| = m$;
- W the weight matrix of the graph, such that for each edge $(i, j) \in E$ the associated weight is positive ($w_{ij} \geq 0$).

The network can also be represented by its adjacency matrix $\mathbf{A}$ defined as :

$$A_{ij} = \begin{cases} w_{ij} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

We will use a normalized version of the adjacency matrix, the transition matrix : P_{ij}.

Furthermore, considering the directed nature of the graphs, we can define the *in-degree* and *out-degree* of a node, respectively for each $i \in V$:

- $k_i^{in} = \sum_{(j,i)\in E} w_{ji}$, for each i, the sum of weights of the edges going to i ;
- $k_i^{out} = \sum_{(i,j)\in E} w_{ij}$, for each i, the sum of weights of the edges starting from i.

Chapter 2

Lagrangian Flow Network - *Modelling the surface of the ocean*

The goal of this chapter is to model as closely as we can the surface of the ocean, the model type of choice is a Lagrangian Flow network, for its convenience. Keeping in mind our main goal is community detection, working on network is convenient. The edges between the subareas (the nodes) are quantified by the number of particles going from a node to another. Because the flow is directed, so is our network, we also mention that it is weighted by the proportion of particles exchanged. The major challenge while modelling oceanic processes is taking into account the wide spectrum of spatial and oceanographic scales. If we consider the scales applicable for the design of an MPA for instance, we are looking at a timescale varying from the order of a day to several years and in terms of spacial scale from an order of the hectometer to 10^3 kilometers, in reference Figure 2.1, taken from [3]. This illustration implying that the timescale and spatial scale are correlated, further discussion will be given on the subject of scales.

In this chapter, we are going to use Lagrangian ocean analysis [26], in order to track a large amount of particles and map out interesting pathways of seawater motion. It might give us insight into the timescale of marine transport and the connectivity of particles. Because fluid particles carry tracers such as salt, nutrients, heat, plankton and marine debris, this type of study can play a major role in the field of marine ecology. Our observation are on the surface of the domain of interest, we'll integrate the following equations in a two-dimensional manner. We model what is called *(Lagrangian) particles* which represents accurately passive objects suspended at the surface of the domain.

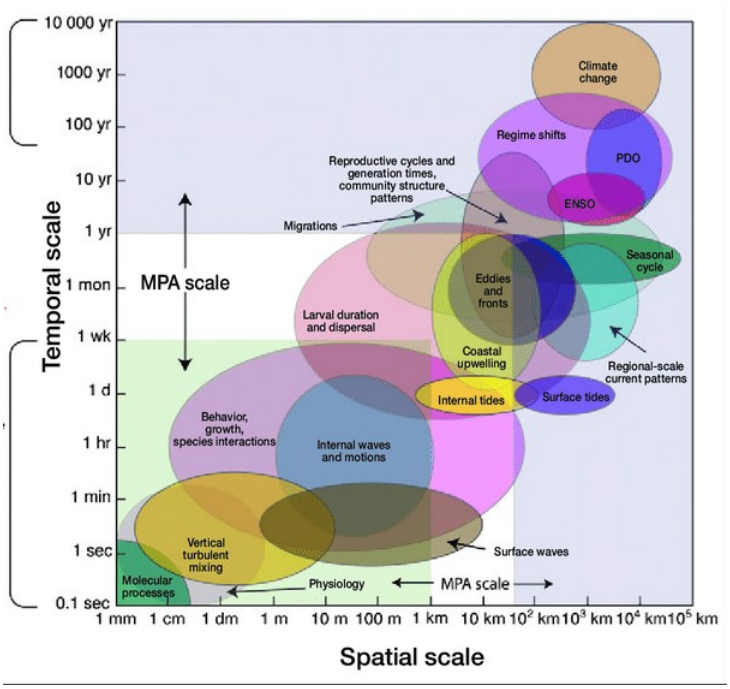


Figure 2.1: Spatial and temporal scales of oceanographic processes

In recent years, a lot of studies have been published on large scale applications, by way of illustration :

- Floating debris can be modelled as Lagrangian particles, such as in the work of Lebreton et al. 2012 [11], where a global ocean circulation model is coupled to a Lagrangian particle tracking model, it can be used to determine the distribution and the accumulation zone of debris (Northern Hemisphere). Ricker et al. [16] did a similar study focused on tidal basins on what they call floating marine litter (FML) and the chemical substances they release;
- Paris et al. [14], used Lagrangian particle tracking to evaluate the transport of oil droplets in the context of the Deepwater Horizon incident.
- Studies have also been made on turtle hatchlings [10] and profusely on larval dispersal [17, 22]

2.1 Theory

2.1.1 Estimating pathways : Offline and online

The tracking of virtual passive Lagrangian particles of zero spatial extent is done by advecting them using velocity fields from either mathematical model or interpolation from observational data. The advantage of estimating pathways this way is that the entire trajectory history of the particles can in principle be stored, this is particularly useful to have an idea of the connectivity of the ocean surface and it allows for a posteriori analysis of connectivity. You could also do conditional statistics, where you will be able to analyze particles that follow a certain rule, particles that circle multiple time one of the pole for instance.

Particles can be advected online, which means trajectories are computed each time step that the Eulerian model is updated ; or offline which make use of stored velocities sampled from the Eulerian model, the latter will be used for our case. You can compute trajectories in a forward mode : from their starting point forward in time ; or backward, from their ending point backward in time. Here we only consider the forward mode.

Another advantage of Lagrangian particles is that it allows investigation of the origin of water masses found at a certain location by doing reverse time analysis using the velocity field stored (possible at least in an offline mode).

2.1.2 The non-zero divergence problem

An important hypothesis is that oceans are nearly incompressible Boussinesq fluid so the velocity field should be divergent-free :

$$\nabla \cdot \mathbf{v} = 0 \tag{2.1}$$

The equation 2.1 states that we need a divergent-free flow, however, we generally have non-zero horizontal divergence on surface trajectories generated by numerical simulations. It can be interpreted as the surface trajectories not mapping volume transport pathways accurately. Nevertheless, it does give insight on preferred pathways of the surface flow, and it is then useful as a diagnosis tool. Our goal being to use the connectivity to detect communities, having a perfectly modelled flow is not as important as having sufficient data on the surface trajectories.

We should be aware of the limitations of the Lagrangian Flow network but looking at the Lagrangian flow network in Figure 2.4 representing the network on the North Atlantic between the 16 December 2015 and the 14 January 2016, we can guess the main currents, even if this example is of significantly lower resolution (for visualization purposes) than what we'll use to do the analysis. The resolution is only of 30x30 boxes, while we use 70x110 semi-squared boxes for the analysis (corresponding to approximately 1°x1° of resolution), furthermore only 120 particles are released in this example, compared to 529 particles in the rest of our work. The Lagrangian Flow network of higher resolution should highlight even more the main currents of the North Atlantic, it is less readable but Figures 2.5, 2.6, 2.7 and 2.8 give the in and out-degrees of the flow network for time step of 30 and 60 days.

You should be able to recognize some things... In order of obviousness, the Gulf Stream, the North Equatorial Current, the North Atlantic Drift, the West Wind Drift, the Florida and the Labrador Current, currents you can see on a map in Figure 1.1.

2.1.3 Deriving the equations for transport

We release N discrete particles into each box of the model. We describe the discrete evolution of each of the particle position $X^{(n)}(t)$. In the following sections of this thesis, we will drop the n label, and it will be assumed we talk about a virtual particle.

The velocity of a fluid particle is the time derivative of the trajectory, computed with material coordinate held fixed (in material coordinate a). Equation 2.2 makes the connection between Lagrangian and Eulerian descriptions by equating the particle velocity crossing a point in space at a material coordinate a to the fluid velocity field at that point :

$$\left(\frac{\partial X(a,t)}{\partial t}\right)_a = v(x,t) \tag{2.2}$$

where $X(a,t) = x$.

Temporal integration of the virtual particles

For the nth particle, located at point $X^{(n)} = x$, we find it's next location. Here, we work in a forward manner in time but a backward temporal integration can be easily derived.

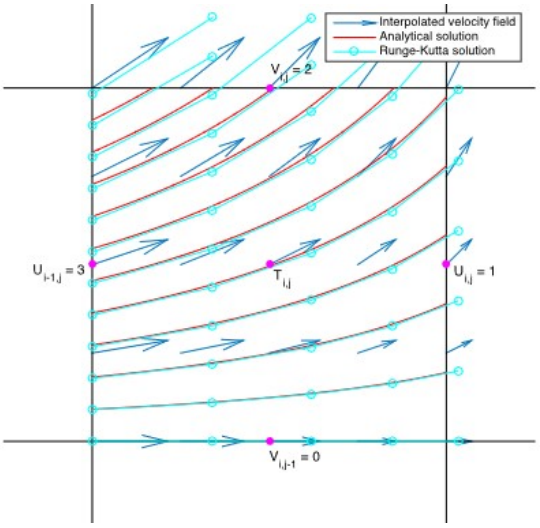


Figure 2.2: Comparison of time stepping solutions on an Arakawa C-grid

$$\mathbf{X}^{(n)}(t_0+\tau) = \mathbf{X}^{(n)}(t_0) + \int_{t_0}^{t_0+\tau} \mathbf{v}(\mathbf{x}(t),t)dt \tag{2.3}$$

The link between the Eulerian world and the Lagrangian one is again that the equation 2.3 involves Eulerian velocity $\mathbf{v}(\mathbf{x},t)$ which is equated to the Lagrangian velocity $\frac{d\mathbf{X}}{dt}(t)$.
Dropping the (n), we note :

$$\Phi_{t_0}^{\tau}(\mathbf{X}_0) = \mathbf{X}(t_0+\tau) \tag{2.4}$$

where X_0 is the initial position of the particle t_0 the initial time and τ the time step.
Unresolved physics can be represented by adding a term in the right-hand side of equation 2.3.
$\Phi_{t_0}^{\tau}$ will map the position of particles in initial time with their position after the time step and we compute a transition matrix from it after its derivation.

Explicit time stepping method

Explicitly to compute the integration term of equation 2.3, you can either multiply the velocity with the time step or use higher order method to integrate the velocity. One popular for the latter is the fourth order Runge-Kutta scheme.

When virtual particle trajectories are computed offline temporal interpolation is needed. Indeed, the interval between 2 velocities stored is bigger than the time step :

thus, we need an idea of what is the velocity to be integrated in between the 2 values.

However, we have to keep in mind that interpolation can be a large source of errors if the velocity fields are stored at a frequency that is too low, i.e. a period in the order of a few days. We can see on Illustration 2.2 that the Runge-Kutta solution and the analytical solution give the same results and the interpolation of velocity between the two are given within the grid.

2.1.4 Constructing the Lagrangian flow network

To represent the problem in a graph $G = (V, E)$, we need to map the nodes V, corresponding to subareas of the domains, with weighted directed edges in E. We can compute the final position of each of the particles with equation 2.3.

Given $m(B_i)$ particles distributed in the box i in initial time t_0, we can assess the fraction of those particles going from box i to box j in a time step τ:

$$\mathbf{P}(t_0, \tau)_{i,j} = \frac{|x : x(t_0) \in B_i \text{ and } x(t_0 + \tau) \in B_j|}{m(B_i)} \tag{2.5}$$

$\mathbf{P}(t_0, \tau)$ is our transition matrix which allows us to approximate the flow as a Markov chain, it can be looked at as an adjacency matrix and used to create the network we will ultimately work on to delineate communities.

Lagrangian particles

To make sure that the diagnosed pathways have statistical significance, we can try to increase their count. If it has no impact on the physical result of interest, the initial number of particles was enough to describe the preferred pathways, if it does, we need to increase their count within reason. Following this logic and keeping in mind the limitation of computational power we have available, we choose a distribution of the particles uniform with 23 x 23 particles per box and 110x 70 boxes. This resolution corresponds to roughly 1°x1° per semi-square. In latitude, it gives us approximately 111 km for the boxes and the particles are spaced by approximately 4.83 km. The division into boxes chosen allows this approximation to also holds for longitude.

How do we deal with an open domain ?

In this master's thesis we will be working on open domains. Note that we don't consider the particles that could be entering the system. When working on a

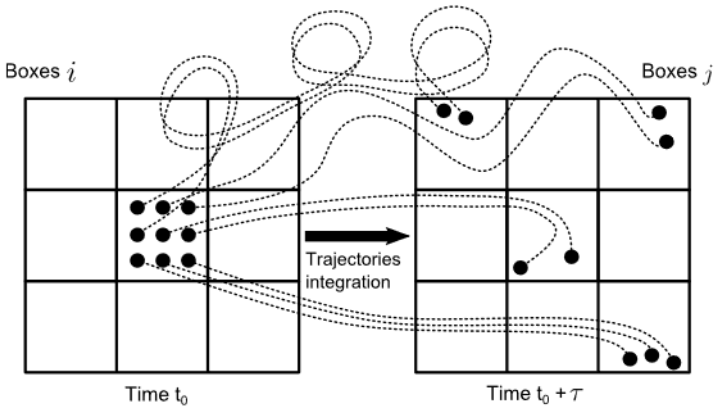


Figure 2.3: Connectivity matrix construction from particles advection

domain that is not closed, the particles released might fall outside the domain and we lose informations on connectivity. In particular, we lose the row-stochastic property of $P(t_0, \tau)$ that ensures that the outgoing edge have a sum weight of one, consequently the matrix no longer corresponds to the probability of a particle travelling between boxes.

In the literature, a few solutions are proposed, the first one is to freeze the particles that exit the boundary. We can keep them stuck at land cells (if you don't have an impermeable boundary at the coasts) or stuck at the boxes forming the boundary at the open border. In that scenario we will still be using the transition matrix given by equation 2.5 to map out the path of the particles. The domain can in consequence be approximated by a closed one.

The second technique we considered is to construct an alternative transfer matrix that represents the probability of reaching B_j starting from B_i and to still remain in the system. It equates the system to a closed-flow network because the particles considered are those who cannot leave. This alternative Markov chain is given by 2.6.

$$\mathbf{Q}(t_0, \tau) = \begin{cases} \frac{\mathbf{P}(t_0,\tau)_{ij}}{S_O(i)}, & \text{if } S_O(i) \neq 0. \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

where $S_O(i)$ is the out-strength of node i, obtained by summing the rows of the adjacency matrix $\mathbf{P}(t_0, \tau)$, it gives the fraction of particles that started in box i and that are still in the system. The latter is the option we will use for the open domain of the Atlantic Ocean.

2.2 Application on the North Atlantic Ocean

2.2.1 Hydrodynamical data

There is multiple way of acquiring surface velocity data : ocean general circulation model, measures by satellite altimetry or high frequency radar or observational data from floats or drifters. We are interested by observational data with a model component to have a more realistic depiction of the flow, that is a global ocean reanalysis product. Recent work on the waters around New Zealand aimed to compare 4 global ocean reanalysis products, mentionning the Copernicus GLORYS one to be the best overall. [**?**]

The Global Ocean Physical Reanalysis product by the Copernicus Marine Environment Monitoring Service (CMEMS) we will use has the identifier "GLOBAL _MULTIYEAR _PHY _001 _ 030" and is the GLORYS12V1, a global ocean eddy-resolving with 1/12° horizontal resolution and 50 vertical levels. The performances have been improved from it's predecessor the GLORYS2V4, by way of inscreasing the model resolution, improving physics, new atmospheric reanalyses, improved quality observational data and better assimilitation of data. [**?**] From a 0.25° resolution which was eddy-permitting, we now have a 0.083° resolution, eddy-resolving.

We test different timesteps for the advection : 30 days, 60, 90, 180 and 365 days. We construct 12 Lagrangian flow network for each of the timesteps (respectively the amount of time studied is then 1 year, 2, 3 and 6 years), except for the time step of a year where we selected the 5 most recent and complete year at our disposition : from the first January 2015 to the 31 December 2019.

For our study we take only the first layer of hydronamical data available, and the velocities for the spatial domain concerned : a latitude ranging from : 0° 56' 9.7" South to 68° 38' 19.4" North and a longitude ranging from 98° 3' 14.1" West to 12° 0' 21.4" East.

2.2.2 Discretization of the domain

The first step in the construction of the Lagrangian Flow network is to divide accurately the domain into subareas that will be our nodes in the chapters to follow. We want to do this division meaningfully. In order to have a good resolution and boxes of roughly same areas, we use semi-squared boxes. This gives us boxes of similar sizes because the meridians do not converge significantly at the latitudes selected.

We should have enough boxes to describe accurately the physical structures, for

Figure 2.4: Lagrangian flow network of low resolution 30x30 subdomains, 120 particles in each box

that we aimed for a resolution of 0.5° which turned out to be a too high demand on the computational power available. We compromised on a resolution of 1°, that translates into a division of the domain into 110 parts for the longitude and 70 for the latitude, giving 7700 boxes of 111 km x 111 km (12321 km^2).

2.2.3 Particles release

The particles are virtual lagrangian particles, that are considered passive and floating along the surface. In each of the boxes previously described we release 529 of those particles uniformly, this means there is approximately 4.83 km separating each of the particles. The particles are then advected using the OceanParcels project. The particles that will leave the domain after the time step considered will be removed from the analysis. We also remove the particles that lie on land at initial time t_0 and the particles that found themselves on land after the time step at $t_0 + \tau$, the particles beached, will be considered stuck there. Note that a way of improving the modelling is to include a velocity field set that simulate the reentering of the particles into the waters after they are beached, however, this is not central for this master's thesis. Another idea for improvement of the model is to consider particles entering the domain, which we did not explore further. We deem our 29 x 29 x 110 x 70 = 4073300 particles to be sufficient to describe accurately

the flow.

2.2.4 Modelling with Parcels

The offline Lagrangian code we will use for the advection of the particles is OceanParcels. They were multiple choices available, for instance Ariane, TRACMASS, Octopus,... Our choice fall for OceanParcels or Parcels - Probably A Really Computationally Efficient Lagrangian Simulator, by Lange and van Sebille (2017) which has the particularity to be customizable to any OGCM with NetCDF data format. The advection method propose different type of integration including Runge-Kutta of order 4 but the 5th order is also available and the method is extendable by the user to take into account customizable advection methods, this could be useful if we wanted to implement a fieldset that reenters the particles that were beached. When compared by the other option provided in van Sebille et al. 2018 [26], Parcels would be the ideal choice for an individual based modelling like ours.

Parcels is a set of Python classes and methods, the user interface is in Python while the computational intensive parts are computed in C. The code is designed to be as customizable as possible and allow the user to explore and experiment on the tracking of Lagrangian particles, its main use is for large scale oceanography. It is still under development to be able to tackle parallelization of the particles tracking (available now for Linux and MacOs but not Windows) and to anticipate the always increasing amount of data and resolution of tomorrow's ocean circulation models.

The first use of Parcels will be to create a FieldSet object from the NetCDF files corresponding to the range of time we want to study. A FieldSet object, as its name indicates, contains information on the northward and eastward velocities in function of the latitude and longitude and the time considered. The partial code bellow give an idea on how to proceed.

```
fieldset = FieldSet.from_netcdf(selectedFiles, # we give the netCDF
    files of the period we want to study
                                variables, # specify that U and V are
                                    the variable
                                dimensions, # specify that U and V
                                    depends on the time, the longitude
                                    and latitude
                                indices = indices, # longitude and
                                    latitude considered
                                allow_time_extrapolation = True,#
                                    refers to the interpolation we
```

```
                                    discuss in section about 'Explicit
                                    time stepping method'
                                )
fieldset.computeTimeChunk(fieldset.U.grid.time[0], 1) # times in the
    netCDF files
fieldset.landMask = np.isnan(ds.uo[0, minLatIdx:maxLatIdx,
    minLonIdx:maxLonIdx].data)# vo could be used too, leads to the
    same landMask - useful to remove particles on land
```

The next steps are to initialize the set of particles, to advect them and to register their destination into the particles set. For the time step used in the advection function provided by Parcels, we will use a time step of 20 minutes for an advection of 30 days and 40 minutes for the advection of 60 days.

```
p_set = particles(lons,lats, releaseTime = t0) # lons, lats, lists
    of the longtitude and latitude where the particles are released
p_set.remove_on_land(f_set) # use of the landMask of fieldset object
    to remove particles not in water

p_out = advection(f_set, p_set, experiment_name = file_expe, runtime
    = delta(days = t),
              dt = delta(minutes = 40), outputdt = delta(hours = 12),
                  overwrite = False) # calls the function execute of
                  Parcels with 'BoundedAdvectionRK4'
```

The final step is to determine the matrix of transition between the boxes of the network with Equation 2.5. From this transition matrix a simple networkx.to_network() suffice to give us the network we will use our clustering methods on.

For further information on Parcels, documentation, tutorials and articles are available online, see `oceanparcels.org`.

2.3 Resulting flow networks

Following the method described above we illustrate the result in the most readable way : with in-degrees and out-degrees. See Figure 2.5 for the in-degrees of the flow network on a time step of 30 days; see 2.6 for a time step of 60 days. Figures 2.7 and 2.8 give representation of the out-degrees for a time step of 30 and 60 days. Each square is a subarea modelled by a node, therefore we can compute the sum of the weight of the edges going to each node : the in-degree and the sum of weighted

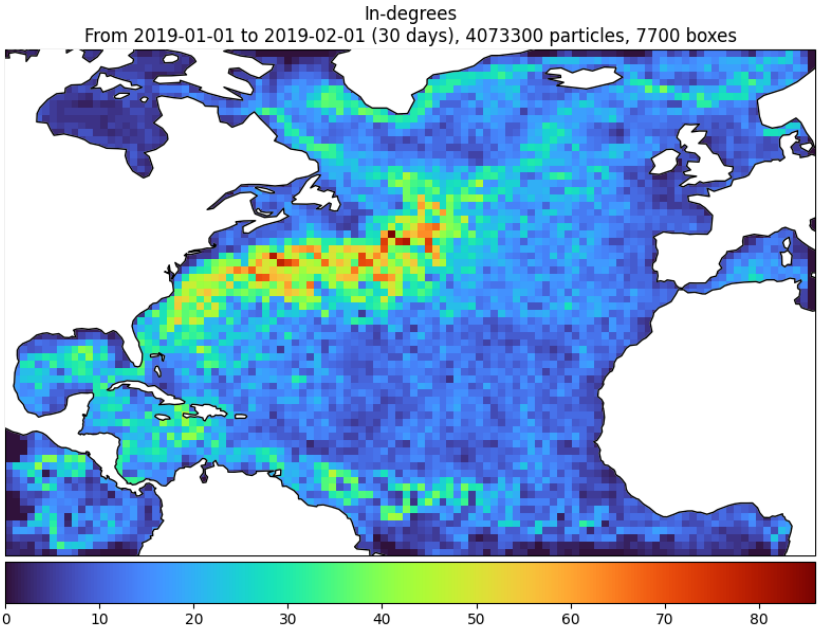


Figure 2.5: In-degrees of the network for $t_0 = 1/01/2019$ and a time step of 30 days

of out-going edges for each node : the out-degree.

As previously stated in the section on *The non-zero divergence problem*, as an argument in favour of the Lagrangian flow network method despite it not being able to strictly map transport pathways, we can observe the main currents of the domain in area where the degrees of the nodes are higher.

We can observe that the in-degrees and out-degrees with a time step of two months instead of one return a higher degree count, this is an expected result since the particles have more time to travel and the connectivity between the boxes is increased.

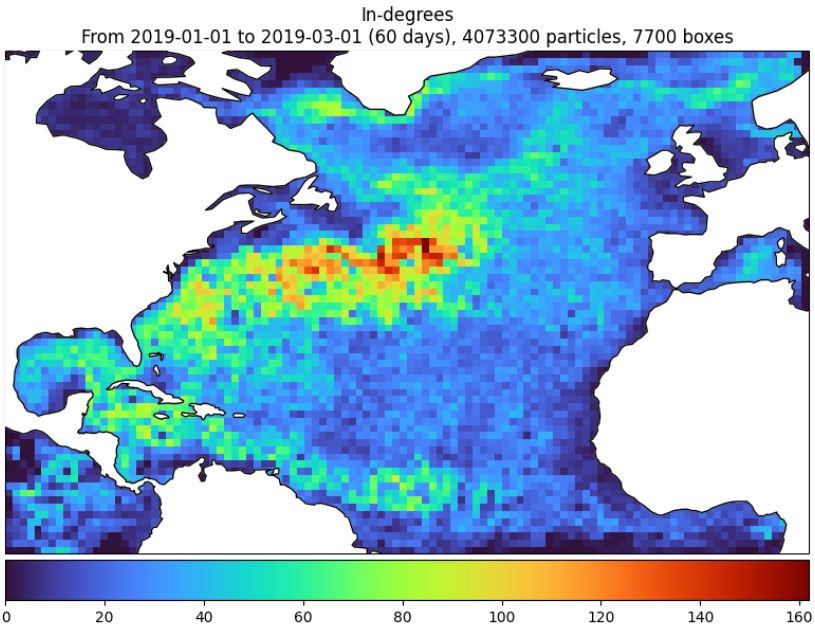


Figure 2.6: In-degrees of the network for $t_0 = 1/01/2019$ and a time step of 60 days

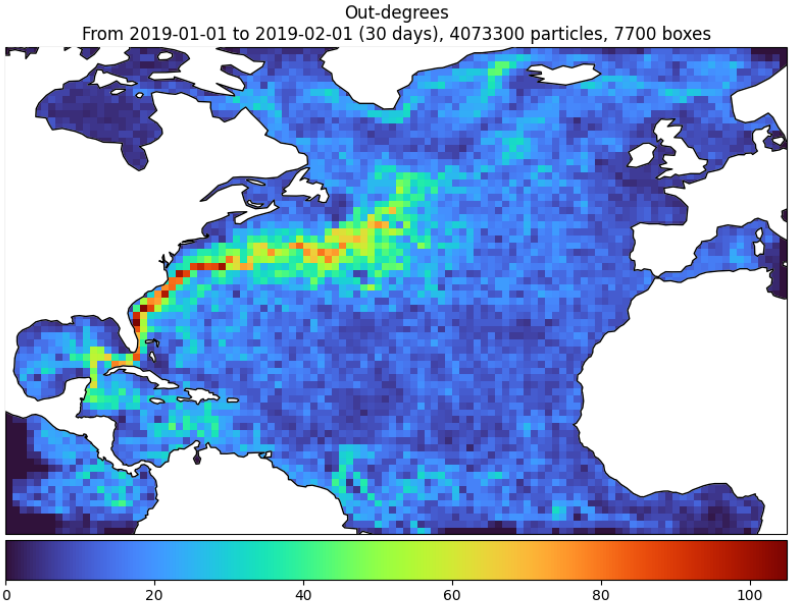


Figure 2.7: Out-degrees of the network for $t_0 = 1/01/2019$ and a time step of 30 days

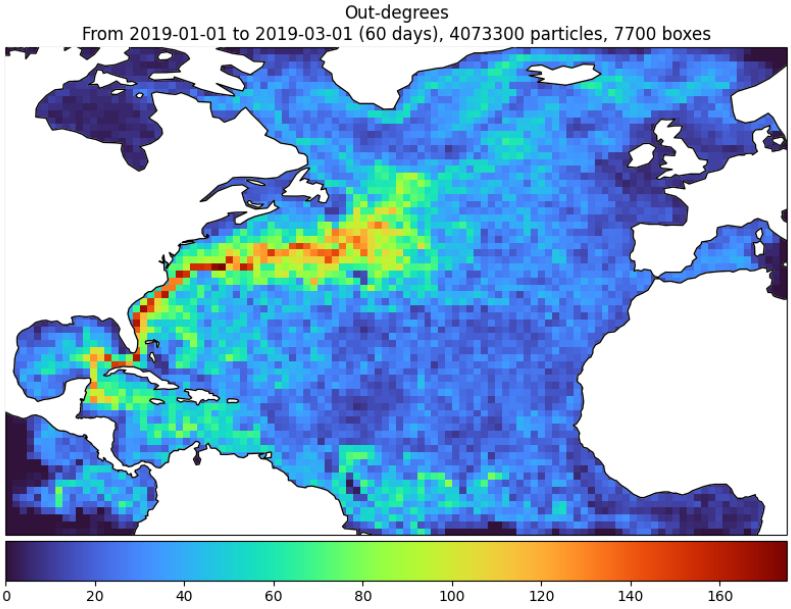


Figure 2.8: Out-degrees of the network for $t_0 = 1/01/2019$ and a time step of 60 days

Chapter 3

Community Detection overview - *Finding suitable methods to detect communities*

This chapter is aimed at giving an overview of the community detection methods we could use in the context previously stated. Community detection offers fascinating results on the structure of graphs, however there is multiple definition of what constitutes a community. For instance, community detection algorithms have been profusely used on social network, who differ significantly from the networks we computed in the previous chapter. The algorithms performing well in the case of social network might be of no use for ours. There is no universally better algorithm performing well on all types of network. We must take into account the type of community we want to find and the nature of the network.

The study of graphs offers a great variety : we can mention social network (such as citation network between different groups of researcher), information networks (e.g. the hyperlink structure of the internet), biological network (e.g. protein-protein interaction networks). The interdisciplinary nature of network science on real world networks makes it a prolific field. In particular, since the emergence of Newman's and Girvan's modularity based community detection, the field benefits from extensive research. With Newman's and Girvan's modularity, we expect to distinguish the structure of the graphs to be clustered from a random graph. If the subgraphs studied exhibit a higher density that what would be expected in a random graph, this subgraph makes a candidate for a community, we can hypothesize that it isn't governed by randomness. A random graph often used for this comparison is the Erdös-Rényi random graph model. [12]

Clustering and community detection are used interchangeably in this disserta-

tion. Clustering is a form of unsupervised learning, in fact, its goal is to cluster together similar object, the similarity measure is in community detection linked to the density of the subgraphs. In the litterature community detection is called clustering, unsupervised learning, partitioning, coarse graining, block modelling, etc. There is a lack of consistent terminology on the subject so we should not get too attach to the technical differences of the vocabulary used in this field, they are rather umbrella terms.

A subgraph with a high density will be considered to have nodes close to each other and will be a candidate to be a cluster (or community). Density as a measure of similarity is not exactly suiting in our case, because of the directness of edges, we want to go beyond the first intuition on community which is that it should have an high edge-density whitin a community and a low edge-density with the other clusters. Here we introduce the directionnality of edges and density is no longer relevant on its own. We are looking for clusters that exhibit patterns related to the flow in our Lagrangian flow network, therefore, we will select two algorithms with a pattern-based approach to be compared in the following chapters.

In the following sections we will refer to *cliques* : a set of nodes that are totally connected, all the possible connections between the set of nodes are present ; they are usually defined on undirected graphs and are the most connected community one could detect. Another concept is the *overlapping communities* : the nodes can belong to several communities. We also refer to *hierachical clustering* : this is a type of clustering that detect nested communities : small communities are contained by bigger one on multiple levels.

3.1 Challenges of a directed network

In the clustering of our Lagrangian flow network, the first thing we have to tackle is the directed nature of the edges. Directionality gives information that should not be neglected, if the network is relaxed to be an undirected one, we lose tremoundous amount of information on the real community structure of the domain. Directness is essential to the meaning of our graph, neglecting the directeness of the edges while simplifying a lot the complexity with lead to erronous clusters. However, the relations are now asymmetrical, which makes it hard because not all traditional community detection methods are extendable to the directional case. In other words by Fortunato [5]: *Developping methods of community detection for directed graphs is a hard task. For instance, a directed graph is characterized by asymmetrical matrices, so spectral analysis is much more complex. Only a few methods can be*

easily extended from the undirected to the directed case. Otherwise, the problem must be formulated from scratch.

Fortunately for us, modelling complex systems into networks has been generalized to more and more sophisticated relationships. Nowadays, we can even see graphs with directed and negatively weighted edges, to represent a relationship that is hostile in a political context for instance. [25] We have enough algorithms formulated for (or extendable to) directed graphs to be able to chose among them.

3.2 Challenges of a multiscale network

As discussed at the begining of the chapter on Lagrangian flow network, ocean offer widely varying temporal and spatial scales. It is essential to be able to detect communities of very different sizes. Methods who fail at this task will be disregarded. However, some algorithm while being relatively well performing on different sizes of clusters, still have a lower bound where they do not detect a community : *the resolution limit* ; communities smaller than this limit will not be detected and be merged with others. There is also an upper bound on the detection of communities : *the field of view limit*, methods suffering from it detect smaller communities than it should be, they split communities in several parts.

Those bounds make the detection of communities on ocean surface particularly challenging.

3.3 Different approaches to Community Detection

In their paper *Different Approaches to Community Detection*[19], Rosvall et al. 2019 discuss four main approaches for community detection :

- the cut-based perspective : the constraint is for the graph to be split into group of roughly equal size with no constraint on the internal density of the groups : this is not suited for our application because we want to find largely different sizes of subgraphs. This statement holds for any spectral clustering method who have been found to not detect accurately different sizes of clusters simultaneously ;
- the clustering perspective : in this paper the clustering approache discussed is the modularity based community detection and its (resolution) limits. We

will discuss it further bellow but we already mention that this technique will not be retained. In short, modularity fails to take into account the multi-scale nature of the context studied, which is a major drawback ;

- stochastically equivallent nodes, this is an approach that we did not implement either and who could have been interesting because the aim of this approach is to group node by the similarity of their role in the network (mainly their connectivity profile), in other words : nodes in the same cluster play a similar role whitin the network. Giving our application on flow networks, this spikes interest, indeed role detection might return interesting structure to be compared with community detection. Unfortunately, choices had to be made and the Constant Potts model was preferred to be compared with Infomap.

- the dynamical perspective concerns Infomap and the Markov stability method, the aim of those methods is to use random walkers paths to assess in which subgraphs the walker stay longer. The perspective is to equate communities by region of the graph where the flow persist longer. This perspective would be particularly interesting in this context because it pairs perfectly with the definition of communities we had in mind in the previous chapter, when we discussed zone of convergence of the surface of the ocean, Marine Protected Areas, larvae dispersal, etc. Between the two examples given, we chose Infomap for its reported performance on the different benchmarks and because it was often used on similar work. It should be interesting to compare the results found with a method used often on similar studies with another promising one : the Constant Potts model.

The paper does not mention our second algorithm of choice that allows us to chose a resolution parameter, that is the Constant Potts Model, which we will compare Infomap with. [25] CPM is said to be *scale-invariant*, because we can choose a resolution parameter that does not depend on the network itself. As a consequence, we have to spend some time finding the right resolution parameter. In our opinion CPM falls into the clustering perspective discussed above. This technique will be discussed further in section 3.5.3. Infomap and CPM both have the advantage that we do not have to assume the number of clusters apriori.

3.4 Benchmarks

In order to assess the performances of the multitude of mathematical formulations in the field of community detection, benchmarks have been a great tool. However, they display a panel of limitations we should mention and the results they provide on performances should be taken with a grain of salt. In chronological order, we

learned about three main benchmark : GN, LFR and RB-LFR.

As previously stated, community detection is an unsupervised task. In order to detect if the cluster have been accordingly detected, benchmarks (or test networks) will create artificial network with artificially generated denser subgraphs to be detected. However, the mathematics behind this generation of artificial cluster could advantage a community detection that uses similar mathematical formulation for the definition of a cluster. It is still rather handy as benchmarks provide a way of comparing algorithms performances. The benchmarks are computed to be more and more realistic [13] but we should be aware that real network are often more complex.

3.4.1 GN benchmark

The GN benchmark (for Girvan and Newman) [6], uses a mixing parameter to construct nodes of the same expected degree and communities of equal sizes. For the 2 previous reasons, any algorithms that have good performances on the GN benchmarks might be lacking in our domain. We can say that this test network is biased. We should therefore use another way to evaluate the methods. It has been found that the LFR benchmark discriminate more the several community detection algorithms. [9]

While the GN benchmark was useful, it is far from realistic, it exhibit a Poissonian degree distribution while real network show a fat tail i.e. most nodes have a low degree, a few nodes have a significantly bigger degree. Real network also have varying sizes of clusters (especially true in our case). Furthermore, the GN benchmark don't take into account weights and directed edges.

3.4.2 LFR benchmark

The LFR benchmark [9] worked on the two limitations of the GN : it gives nodes with degrees following a power-law, therefore, it entails that the fat tail characteristic of real network is present. The communities are planted with heteregenous sizes and densities. They also give the option of directed and weighted graphs. Another pro of the LFR benchmark is that they are build rather quickly, allowing the researcher to work on bigger, more elaborate networks. However, the major drawback of the LFR benchmark is that it does not plant cluster with a hierarchical structure, so it won't be able to account for overlapping communities that are common to a lot of real networks.

In their paper *Community detection algorithms : A comparative analysis* Lancichinetti and Fortunato use the LFR benchmark they designed with their colleague Radicchi (the name of which is derivated from their three respective initials) on several methods ; namely, the method of Blondel et al., the RN method, Infomap, etc. Note that this study is rather old (2009), and its results should be taken with a grain of salt. For instance, since then the Louvain method has known change, see *From Louvain to Leiden: guaranteeing well-connected communities*[24].

Nonetheless, the results presented are taken into account in this master's thesis. While analysing the results the algorithms give us on the LFR benchmark, they also tried to give them a random graph in entry. They observe that method of Blondel et al. always find a few clusters on them anyway, this is a problem Traag address in his work on a new algorithm called Leiden which started from the observation that several clusters returned by Louvain were in fact badly connected. The Potts model approach by Ronhovde and Nussinov has been found to find non trivial partition for any size of the random graph and Infomap always finds a single cluster with all the nodes in it : that was the result desired on a random graph (note that in small graphs it does not always behave like that).

They then proceed to test the algorithm on the actual benchmark, on a version that is weighted and on a version that is directed. The comparative analysis conclude on Infomap being the best performing overall, it can be aplied to weighted and directed graphs and detect accurately communities on a large spectrum of cluster sizes. They also note the performances of the Louvain algorithm and the Potts model approach. We will come back to the two latter algorithms in section 3.5.

A final and important remark taken from the report of Lancichinetti and Fortunato 2009 is that using multiple methods for community detection might be a way to extract informations that are not dependent on the method selected.

3.4.3 RB-LFR benchmark

The RB-LFR benchmark graphs improve upon the LFR by introducing the lacking hierarchical component of cluster. [28] The study of method on that set of graphs has been made on Infomap, Louvain and the Hierachical Stochastic Block Model. We note that the clustering is hard for the three of them but Infomap is still reported to be the better performing one.

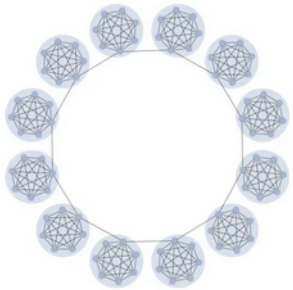


Figure 3.1: The ring of cliques network illustrates the problem of resolution limit

3.5 Algorithms of interest

3.5.1 Modularity based algorithms

Modularity is a pillar of community detection : the number of communities are determined without having to specify it, the popular method of Louvain optimizing it remains quite competitive, there is pros to this approach and therefore it has been studied extensively this past years. In surveys on the benchmarks presented, they often refer to the Fast modularty optimization by Blondel et al [1]. The algorithm optimizes the Newman-Girvan modularity in the neighborhood of each node. A partition is identified by this process and we can replace the clusters by node (supernodes) and iteratively repeat the process. The described process is stopped when the modularity do not increases further. The advantage of this technique is that it gives us a compromise between the modularity estimation (of greedy technique for example) and computational complexity. The complexity is linear in the links of the graph.

However modularity based algorithms including Louvain suffer from the resolution limit. [8] It can be shown that on a ring of cliques (see Figure 3.1), modularity tends to cluster the ring of cliques rather than the cliques separately. Therefore, there is a lower bound such that communties smaller than this bound will be invisible to the clustering method.

The modularity also suffers from the field of view limit, i.e. it partitions clusters that should be merged. This can be observed on a ring of rings (see Figure 3.2), where the probability of remaining in a ring and not moving to another is high. Long cycles in graph might be detected into several different clusters instead of one. [24]

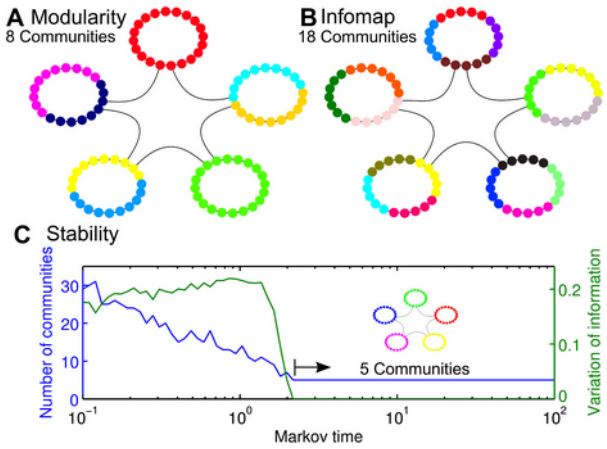


Figure 3.2: The ring of rings network illustrate the problem of field of view, taken from [20]

3.5.2 Random walker based algorithms

Random walks are useful in the dynamical perspective described above. It is a process where a random walker starts at a node and take random decision to travel between nodes based on a certain probability of transition defined on each edges. This probability of transition is the transition matrix P we defined in chapter 2, given by equation 2.5. As a reminder, P_{ij} is defined as the proportion of particles travelling from box i to box j, it is a normalized version of the adjacency matrix. The random walker uses the probabilities contained in P to make a decision but the walker is not to be confused with the Lagrangian particles that defined the transition matrix.

Random walks are considered to be *Markov chains* [12] : the next step of the random walker is chosen from a set of states : the nodes that have an in going edge connected to its initial node. The possible states in time $t + \tau$ only depend on the current state t.

Infomap and Markov stability both use random walk to study where the flow persists and identify communities from this information. [2]

Infomap

Infomap [18] is categorized to be the best method on several paper testing different methods on benchmark. [9, 28] Aside from that fact, the underlying intuition on

community is well suited to the domain we have : a community is viewed as a subgraph where the random walker is likely to be trapped in. This method is based on information theory and has the goal to represent the information of the random walker as concisely as possible. Namely, the information to compress is the *description length*, a network with communities where the walker stays longer has a description of its path that can be compressed. In chapter 4 we will give element of the information theory used and explain how the methods compress the information contained in the random walks in order to find communities.

We should mention that the map equation is subjected to resolution and field-of-view limits just like the modularity. We chose it to compare it to a method that is not subjected to those problems and to be able to quantify those limitations. Another argument for this choice is its performances on the benchmarks and the frequency of its use on similar work.

Stability

The clustered autocovariance of the Markov process described above is used to compute the stability of a partition. Indeed the autocovariance of the clustered Markov process wheighs the persistence of a cluster in time. The stability evolve with time, then we can compare the partitions given by the algorithm at a given time scale and determine in this way the optimal one. To briefly conclude, this method returns several solution over different time scales, it is up to the researcher to select the solution best suited to his application. [4]

3.5.3 CPM

Constant Potts model is based on the Potts model by Ronhovde and Nussinov (RN), it is not based on a null model in contrast to modularity based approaches.

It compares the network to a constant parameter γ_{CPM} : we want to detect community with density greater than that criterion and the connectivity between communities should be lower than that parameter, called the *resolution parameter*. This approach is interesting in that sense, we hypothesize that we should be able to detect communities smaller than in the solutions provided by Infomap and be free of the resolution limit.

3.6 Conclusion

Taking into account the special needs of our Lagrangian flow network, we explored the clustering methods available to us. This exploration is not exhaustive, there is a lot of interesting methods we did not mention : Label Propagation, Stochastic Block model, Auto Information State Aggregation, etc. It becomes apparent that there is no superior method on all types of graphs, just different definition of what constitutes a community. Here we are in need of a pattern-based method that would take into account the directed nature of the flow network while simultaneously detecting clusters of greatly varying sizes.

Considering the use of Infomap in several similar studies [17, 15] and CPM in Thomas et al. [22] while keeping in sight the performances of Infomap despite its resolution and field-of-view limits, we want to compare those two methods and report their shortcomings. It is interesting to analyze two different clustering methods and try to find some structures that may be algorithm-independent.

Chapter 4

Infomap - *Finding the Markov time*

4.1 Principle

Infomap base its principle into information theory, the field of representation and quantification of information. Its goal is to optimize the Map equation, namely find a binary code for each node of the graph, that uniquely defines the position of a random walker on the graph. [20]. We want to compress the description of the random walker path by reusing codewords that are in separate communities.

As an analogy, we can take phone number : *"Compare it to calling somebody on a land line. If you need to call someone within the same village (or even organisation) you usually only need a few numbers. For example, you dial your best friend with the phone number "1105". Now if you want to call somebody in another village (with number "38"), you will first have to dial out (using the code "0"), then dial the access code and then the phone number again. For example, your other friend lives in another town and you dial "0-38-1105". Notice that the actual phone number can be the same for both friends: "1105". This is the same idea for the random walker: nodes in different communities can reuse the same code."* [24]

4.1.1 Resolution limit of the map equation

In [2], Bohlin et al. stated that, even though the map equation is subjected to the resolution limit, it is less likely to suffer from it than the modularity maximization because the formulation (here we refer to the 2-level formulation, the one we use) does not depend on the total weight of all links in the graph but only on the total weight of links between communities.

4.1.2 Field-of-view limit of the map equation

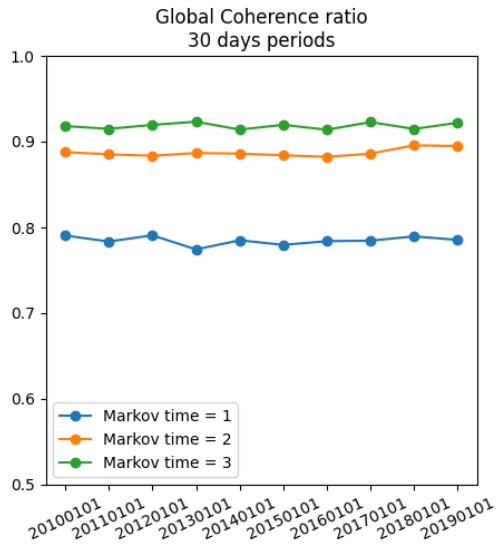
However, the design that makes Infomap less prone to resolution limit : the fact that it is greedy with respect to locally dense subgraphs, makes it prone to the field-of-view limit. It has been found that Infomap overpartitions even more than the modularity. [20].

4.1.3 Markov time

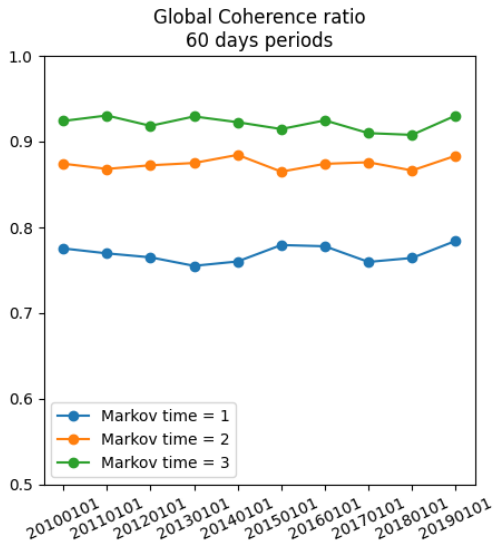
In this section, we want to determine a good Markov time parameter for the clustering. For that we applied Infomap with a two-level formulation and Markov times ranging from 1 to 3, the method was applied on the 1st January of 10 years (as a way of avoiding seasonal variability). In Figure 4.1, the average global coherence ratio for 30 days and 60 days (time step in advection of the particles) is given. The coherence ratio will be discussed further in Chapter 6. For now, we just need to know that it measures how "weak" is the connectivity between cluster. In Figure 4.2, the mixing parameter, measuring how well the communities interior are mixed, is displayed. Observe that having the best coherence (Markov time of 3) leads to the worst mixing parameter and having the best mixing parameter (Markov time of 1) leads to the worst coherence. Because the coherence is still good and stable between time steps for a Markov time of 1 (around 0.8), we will choose it as our parameter of choice for the rest of this dissertation.

4.2 Results

We illustrate in this section the clustering solutions for 30 days and 60 days time step with initial time 1/1/2019 directly on the map. The colors are chosen to indicate the cluster size as you can see on Figures 4.3 and 4.5. Another visualization is given on Figures 4.4 and 4.6 where the nodes are colored by the mean size of cluster they find themselves in for 10 clustering solutions on years 2010 - 2019. Nodes with a high value are consistently in big cluster and coincide with main currents. Nodes that have a low value are in the center of the gyre.

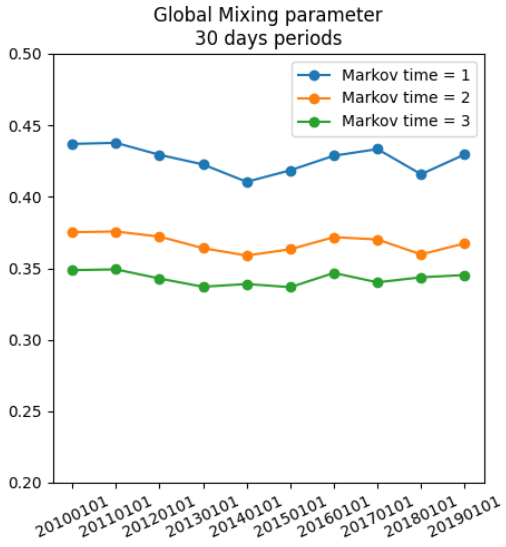


(a) Global coherence ratio for 30 days

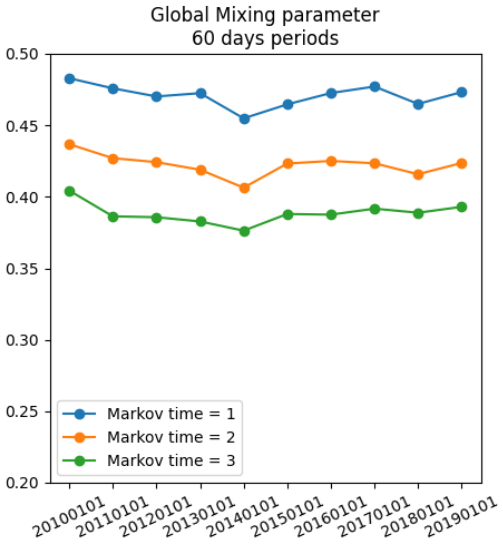


(b) Global coherence ratio for 60 days

Figure 4.1



(a) Global mixing parameter for 30 days



(b) Global mixing parameter for 60 days

Figure 4.2

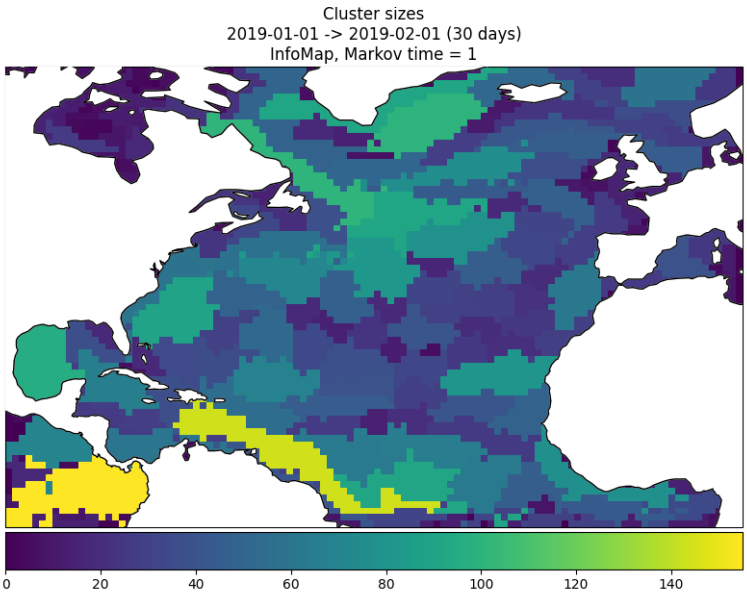


Figure 4.3: Clustering for a Markov time of 1, by the sizes of cluster, 2019, 30 days

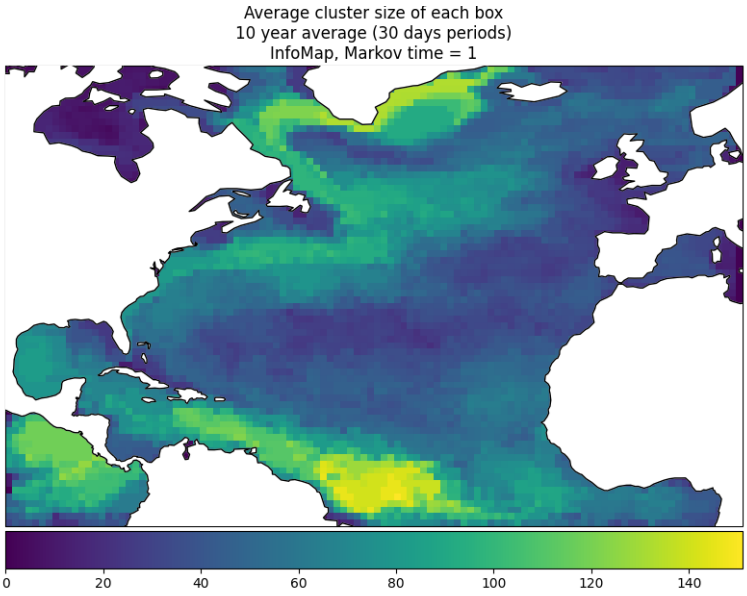


Figure 4.4: Clustering for a Markov time of 1, by the sizes of cluster, 2010-2019, 30 days

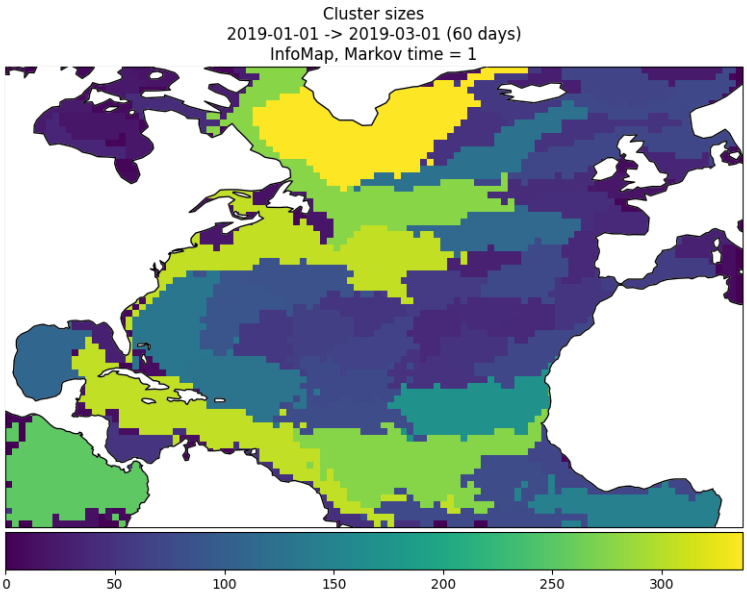


Figure 4.5: Clustering for a Markov time of 1, by the sizes of cluster, 2019, 60 days

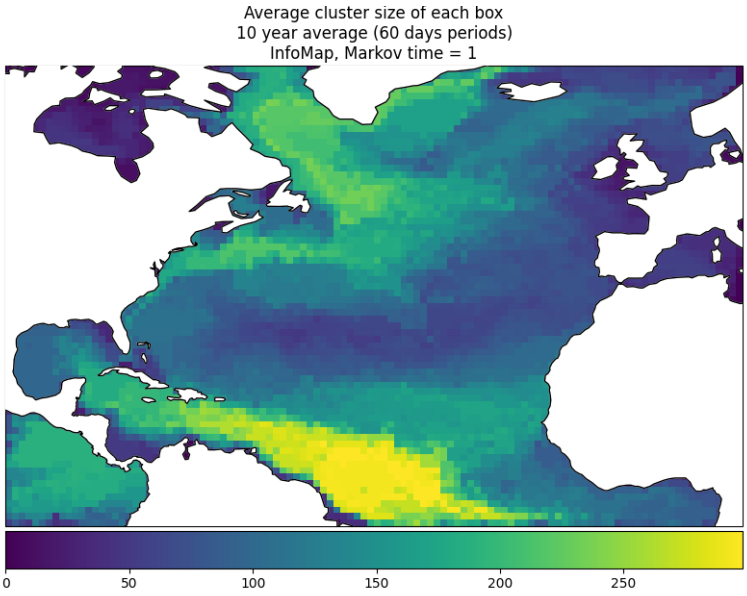


Figure 4.6: Clustering for a Markov time of 1, by the sizes of cluster, 2010-2019, 60 days

4.2.1 Quality metrics

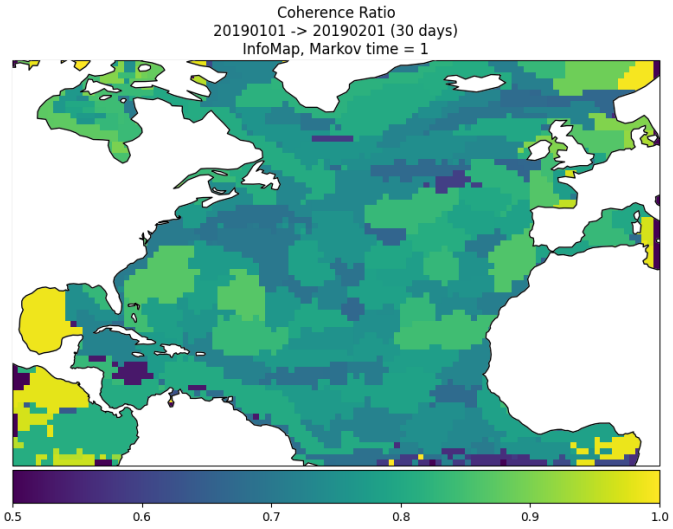


Figure 4.7: Local coherence ratio, 2019, 30 days

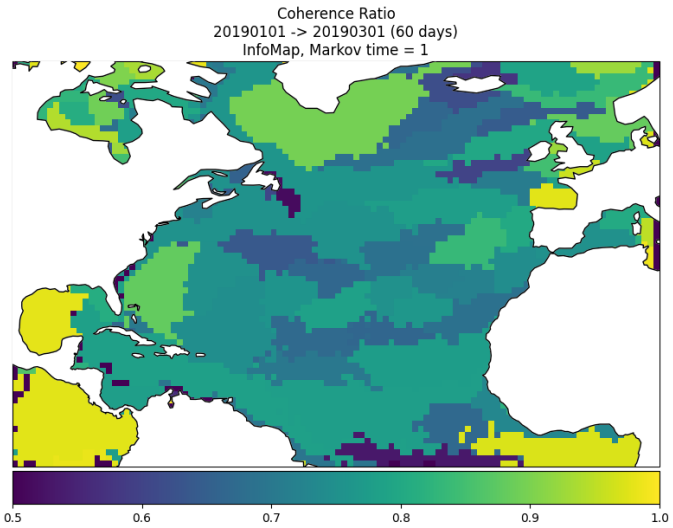


Figure 4.8: Local coherence ratio, 2019, 60 days

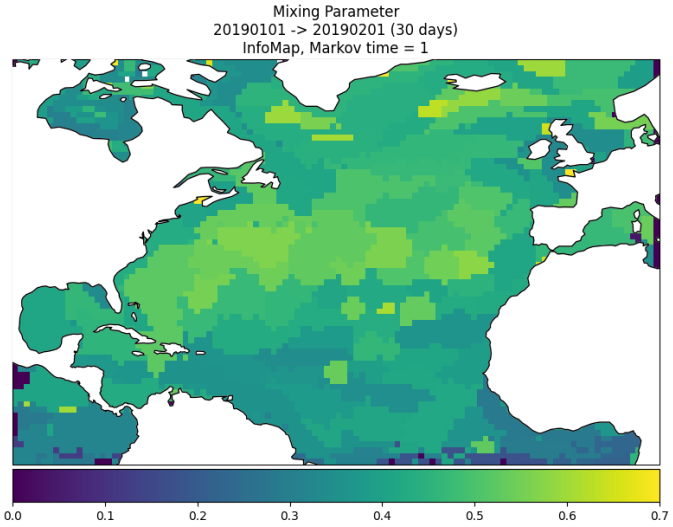


Figure 4.9: Local mixing parameter, 2019, 30 days

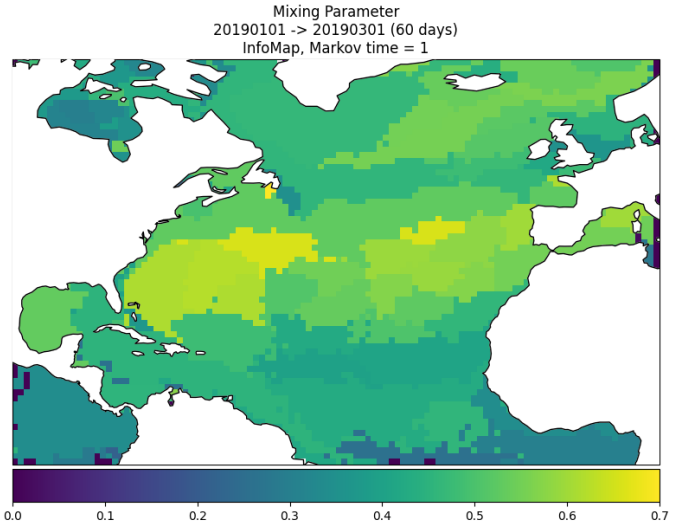


Figure 4.10: Local mixing parameter, 2019, 60 days

The local quality metrics are in Figures 4.7, 4.8, 4.9 and 4.10: respectively the coherence ratio for 30 and 60 days and local mixing parameter for 30 and 60 days for initial time being the 1st January 2019. For the coherence, we can see that the global coherence ratio will stay roughly the same from 30 days to 60 days as coherent regions grow but so do incoherent one.

For the mixing parameter however, the area taken by well mixed clusters increases.

Chapter 5

Constant Potts model - *Finding the resolution parameter*

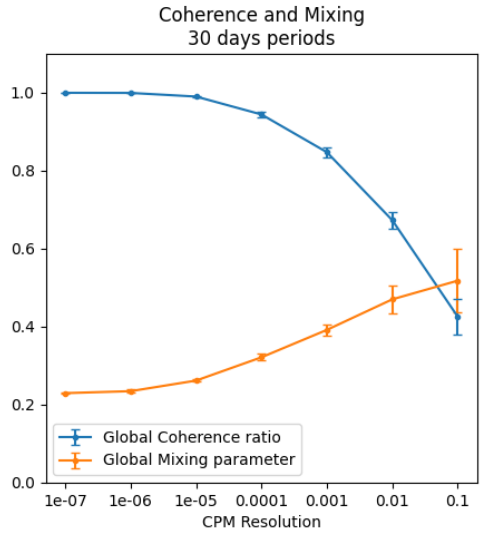
5.1 Principle

In this chapter we will describe the Constant Potts model, it is based on the Potts model but with a constant value. This method is said to not suffer from the resolution limit. The constant is called the resolution parameter and is used to split the domain wherever the density is bellow it. Simultaneously, we want community were the density is above the resolution parameter. [23]

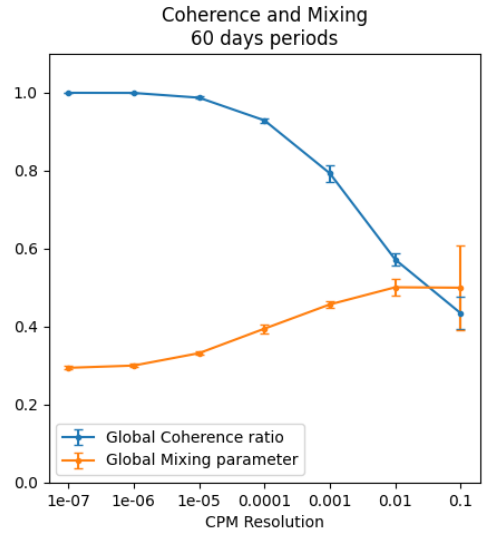
This method allows us to detect small communities as long as we set the parameter accordingly. Indeed, a high constant will lead to bigger communities and fewer of them. Those results are found because we do not focus on node density inside a cluster but rather on the gap in edge-density between subgraphs. With the Constant Potts model (or CPM), we could analyze the network at increasingly bigger levels of community density, corresponding to smaller and smaller communities.

5.2 Resolution parameter

The same way we chose a Markov time in the previous chapter, we have to find a suitable resolution parameter. For that we use the evolution of mean global coherence ratios and mean global mixing parameter over 10 years in Figure 5.1. We want to chose a resolution that have a good coherence and mixing with a cluster count (see Figure 5.2) comparable to the Infomap clustering solutions. For all of those reasons, we chose a resolution parameter of $8 \cdot 10^{-4}$.

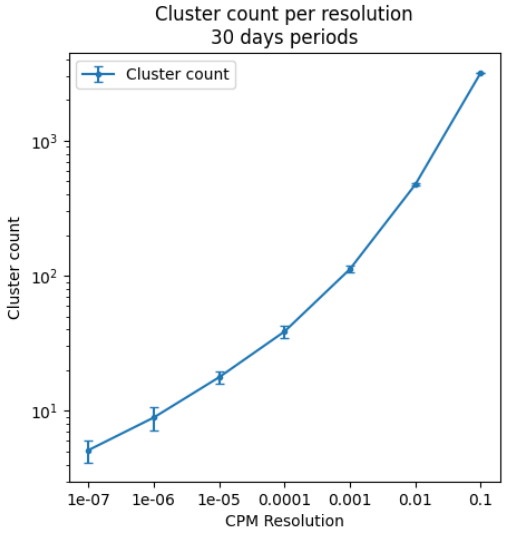


(a) Global coherence ratio and mixing parameter for 30 days

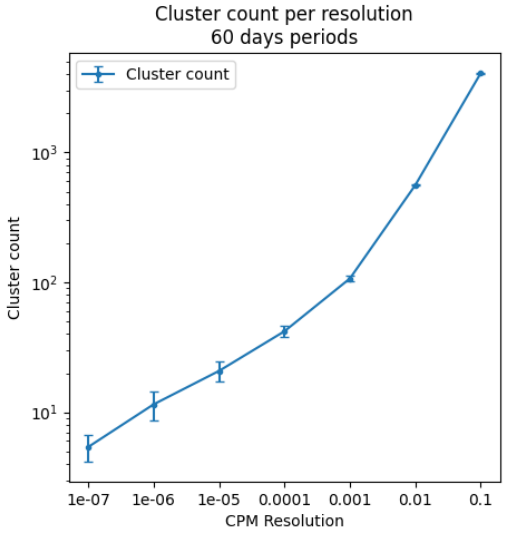


(b) Global coherence ratio and mixing parameter for 60 days

Figure 5.1



(a) Cluster count by resolution parameter for 30 days



(b) Cluster count by resolution parameter for 60 days

Figure 5.2

5.3 Results

5.3.1 Clustering solutions for a resolution parameter of $\gamma = 8 \cdot 10^{-4}$

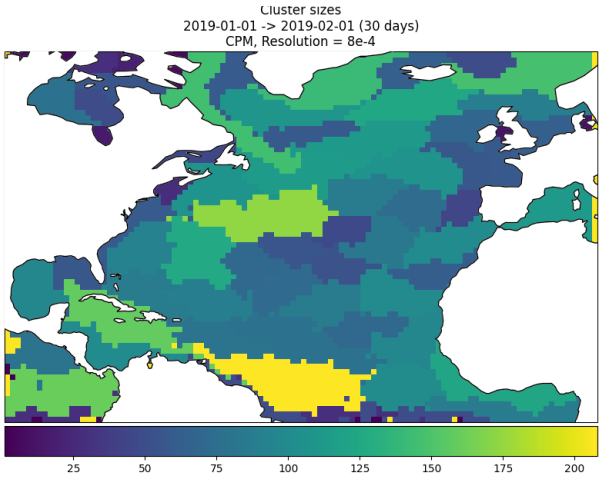


Figure 5.3: Clustering for resolution of $8 \cdot 10^{-4}$, by the sizes of cluster, 2019, 30 days

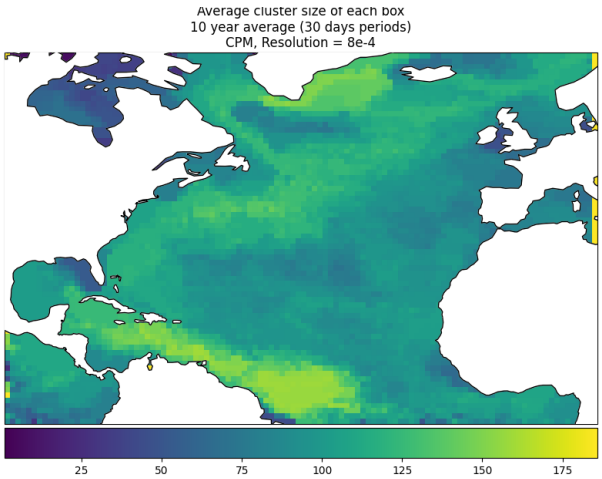


Figure 5.4: Clustering for resolution of $8 \cdot 10^{-4}$, by the sizes of cluster, 2010-2019, 30 days

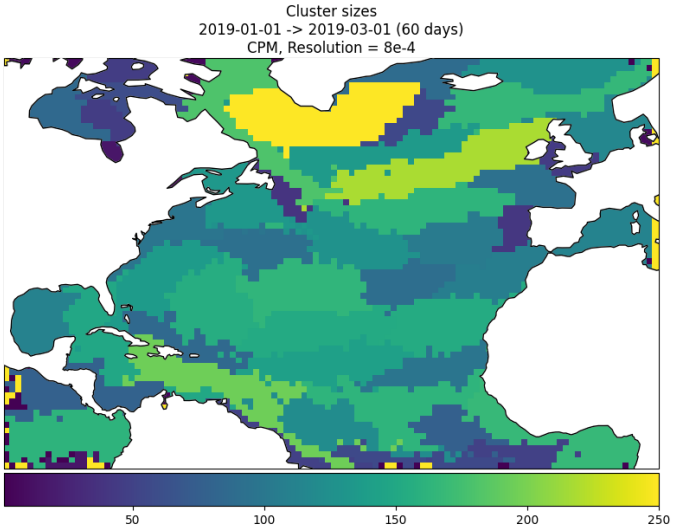


Figure 5.5: Clustering for resolution of $8 \cdot 10^{-4}$, by the sizes of cluster, 2019, 60 days

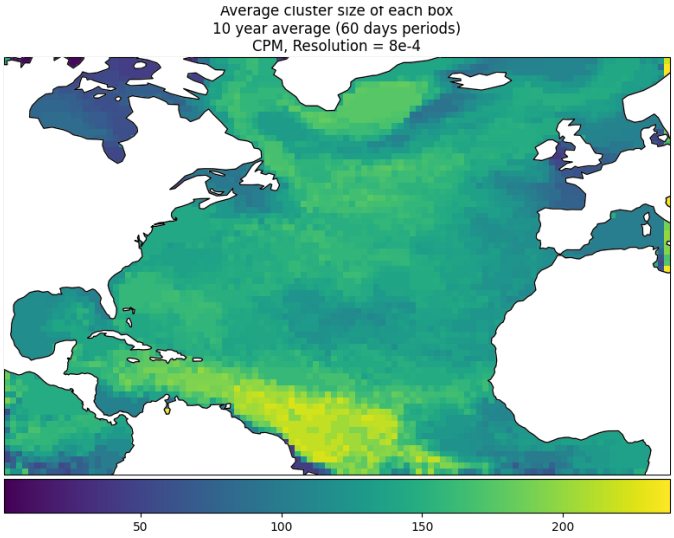


Figure 5.6: Clustering for resolution of $8 \cdot 10^{-4}$, by the sizes of cluster, 2010-2019, 60 days

5.3.2 Quality metrics

Local coherence ratio

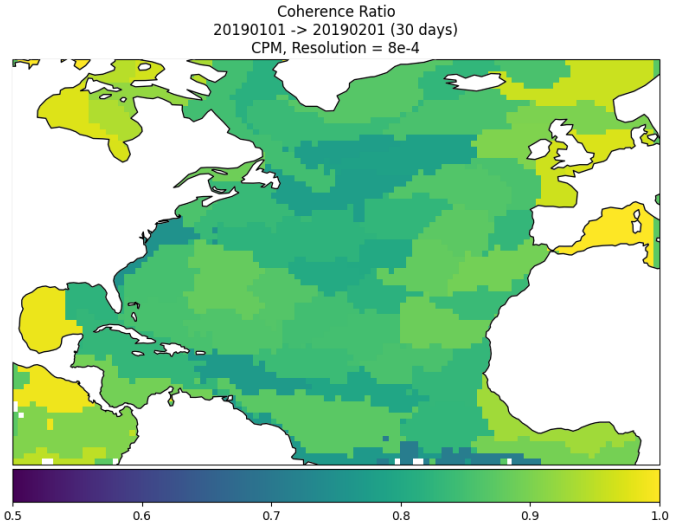


Figure 5.7: Local coherence ratio, 2019, 30 days

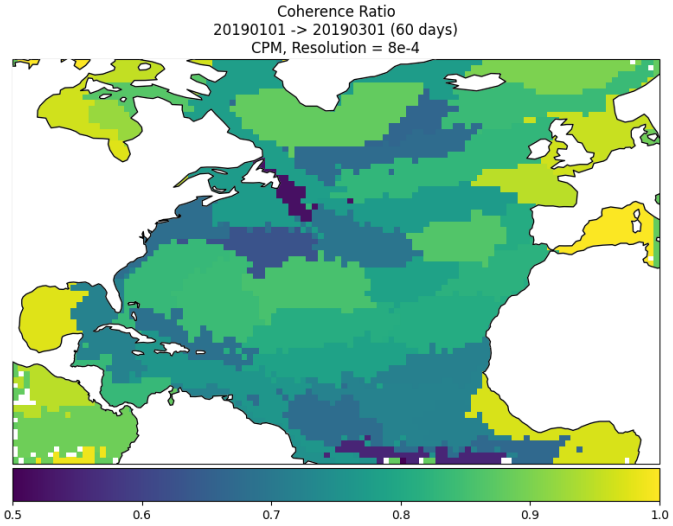


Figure 5.8: Local coherence ratio, 2019, 60 days

Local mixing parameter

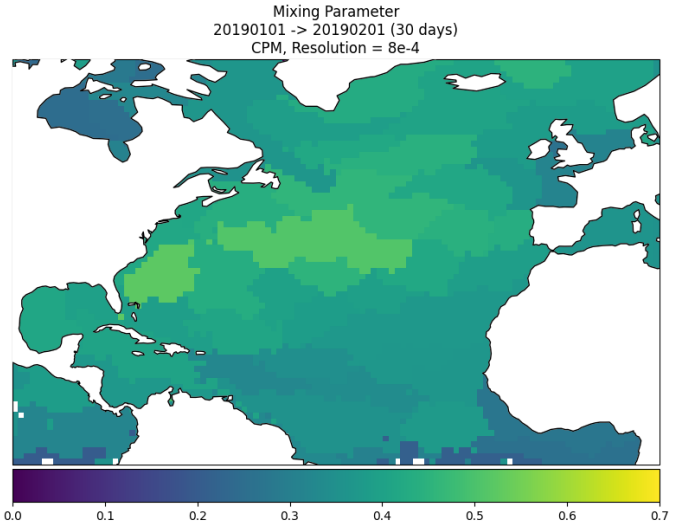


Figure 5.9: Local mixing parameter, 2019, 30 days

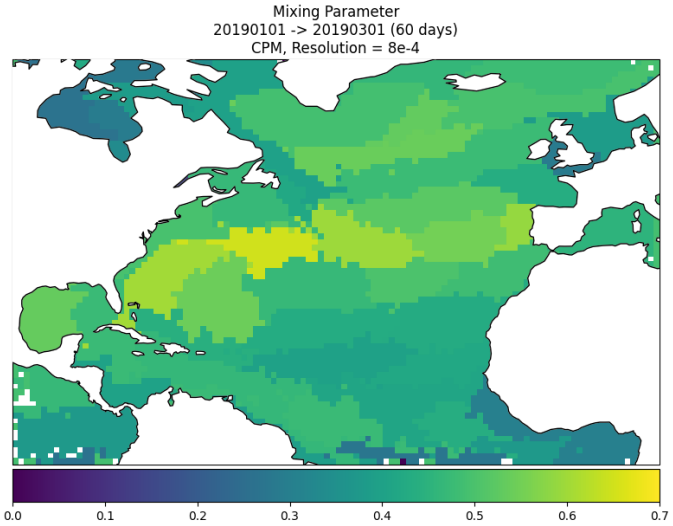


Figure 5.10: Local mixing parameter, 2019, 60 days

Chapter 6

Quality Measures - *Assessing the clustering solutions*

In order to assess the quality of our clusters, we define in this section different metrics. Namely two of them, the mixing parameter and the coherence ratio, intended to measure if partitions are well-mixed internally and sparsely connected to the rest of the graph.

Two other ones are handy : we illustrate the persistence of boundaries and the average distribution of the sizes of the cluster for 10 years. The latter has the intent to assess the limitations of our two techniques, in other words, we want to know what on average is the smallest size of cluster and the biggest too, respectively in an effort to measure resolution limit and field of view limit. The former give a clear illustration on where the boundaries of cluster are persistently found, note it corresponds to known oceanic features.
As a final word of introduction, it's important to keep in mind the ocean currents while comparing the results. We include Figure 6.1 for easy comparison.

The additional notation used in this section are :

- the ensemble of clusters K of size p, designated by the indices k such as : $k \in K = \{1, 2, ..., p\}$;
- for each cluster k, the number of boxes included is M_k ;
- the total number of nodes on the network is $M = \sum_{k \in K} M_k$;
- for each node or box B_i with $i \in I = 1, 2, ..., M$ there is a certain number of particles $N_i = m(B_i)$;
- the total number of particles is $N = \sum_{i \in I} N_i$.

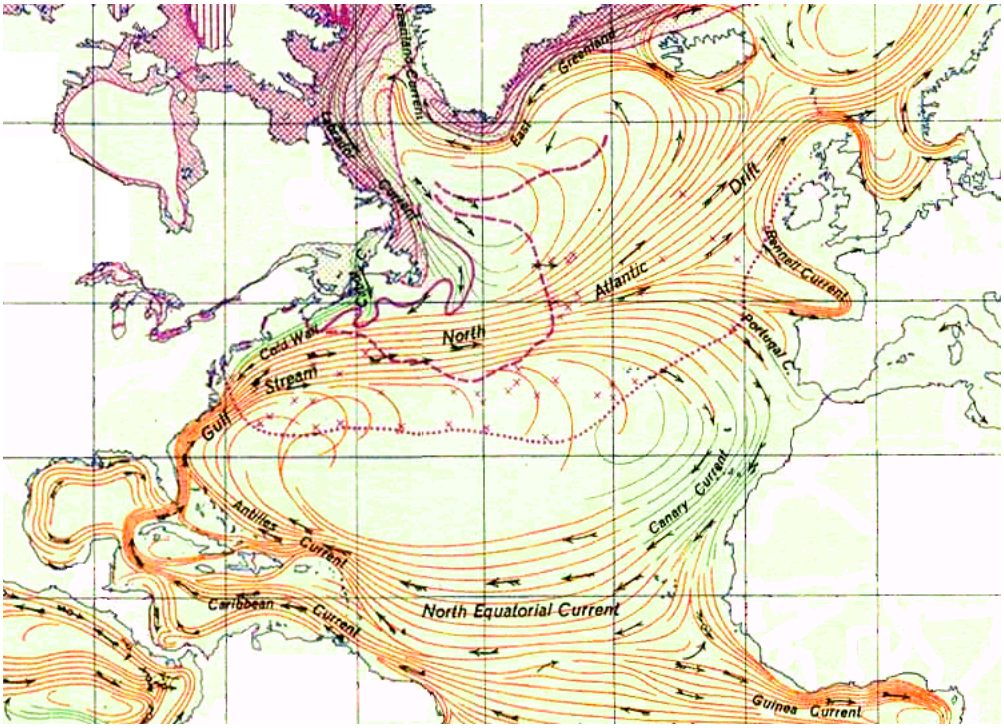


Figure 6.1: North Atlantic Gyre

6.1 (Global) Coherence ratio

Ser-Giacomi et al. 2015 [21] used a *coherence ratio*, written ρ_k for each community found k. The goal of the metric is to measure the fraction of particles that are initially released in a given community and stayed in that community at the end of the time step. A coherence ratio is calculated on every cluster found and then we can compute a *global coherence ratio* for all the clustering solution by doing a weighted sum on the local coherence ratio. This allows us to have a direct measure on how little the boundaries of clusters are crossed by particles. As an example, if the coherence ratio is of 0.76 on a given cluster in a flow network with a time step of 2 months, 76% of the particles that were on this province are still there after 2 months. The complement of ρ_k, $\bar{\rho_k} = 1 - \rho_k$, gives the proportion of particles leaking from the cluster after the time step. Intuitively, we understand that the coherence ratio can be used as a measure on how weakly connected the clusters are with each other. [17]

The equation to compute the coherence ratio for each cluster found k is given by 6.1, with I_k representing the nodes inside the cluster and $N_i = m(B_i)$ the number of particles inside each node B_i of the cluster. The numerator is the number of particles that stays in the cluster and the denominator the total number of particles.

$$\rho_k = \frac{\sum_{(i,j)\in I_k} N_i \mathbf{P}_{ij}^{t_0,\tau}}{\sum_{i\in I_k} N_i} \tag{6.1}$$

We expect to see high value of coherence for good clustering solution. Of course, since it measures a probability to stay within a cluster, its value is between 0 and 1.

A global view of the coherence ratio on the domain can be given by the *global coherence ratio* :

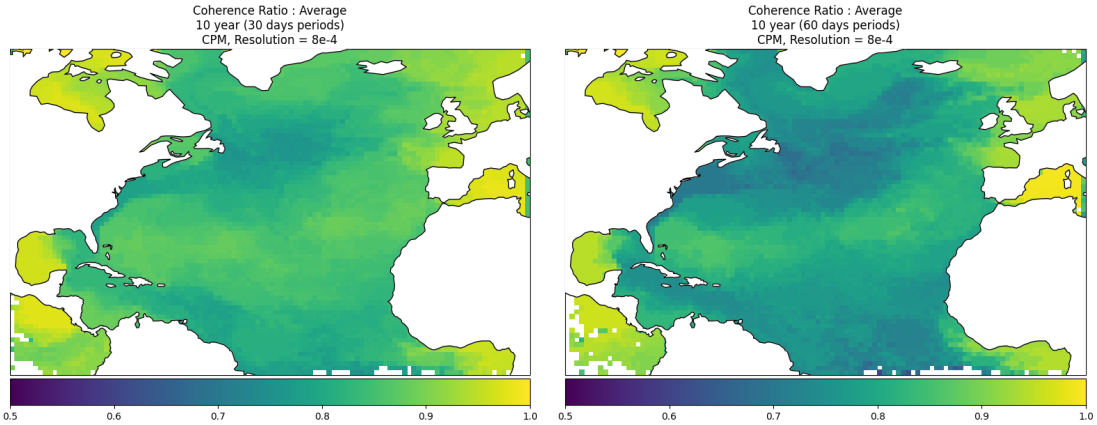
$$\rho = \frac{\sum_{k=1}^{p} \rho_k M_k}{M} \tag{6.2}$$

M_k being the number of nodes B_i inside the cluster numbered k, in other words the size of the set I_k. While M designate the total number of number in the p cluster. In this way the global coherence is weighted by the relative sizes of the clusters.

6.1.1 Comparison of coherence ratios

CPM has an average global coherence ratio of $\overline{\rho^{30}} = 0.856$ on a flow network with time step of 30 days with a standard deviation that is low, that is 0.0072. For a time step of 60 days, it is not surprising to see the average global coherence ratio drop slightly to : $\overline{\rho^{60}} = 0.811$ as the clusters become bigger the ratio of particles leaving them is slightly increased, the boundaries of clusters becoming longer. But the decrease is not significant, this represent about a 5% drop. The average does deviate more, as illustrated by the standard deviation that is now 0.0118, again this is to be expected : the particles of the flow network are allowed to be advected for a longer time resulting in bigger differences between the clustering solutions found for each year. The local average of coherence ratio is represented in Figure 6.2, you can observe that the region of higher coherence are the one with slower flow. Areas like the Gulf Stream, Labrador Current, North Atlantic Drift and North Equatorial Current are subjected to smaller coherence ratio and smaller standard deviation as illustrated by Figure 6.3

Infomap has a lower global coherence ratio in both time steps chosen : $\overline{\rho^{30}} = 0.785$ and $\overline{\rho^{60}} = 0.776$ but with slightly lower value for the standard deviation : 0.0048 and 0.0092, this difference is not significant enough to make a difference. However, notice that just like in CPM's case, when the time step is multiplied by 2, the standard deviation is roughly multiplied by 2 as well, indicating the greater variability in the pathways chosen by the released particles.
CPM performs better for the coherence metric. Illustration of average local coherence ratio for Infomap is also provided : 6.4 and the standard deviation of the



(a) 30 days (b) 60 days

Figure 6.2: Average local coherence ratio on 10 years - CPM

coherence ratios is in Figure 6.5.

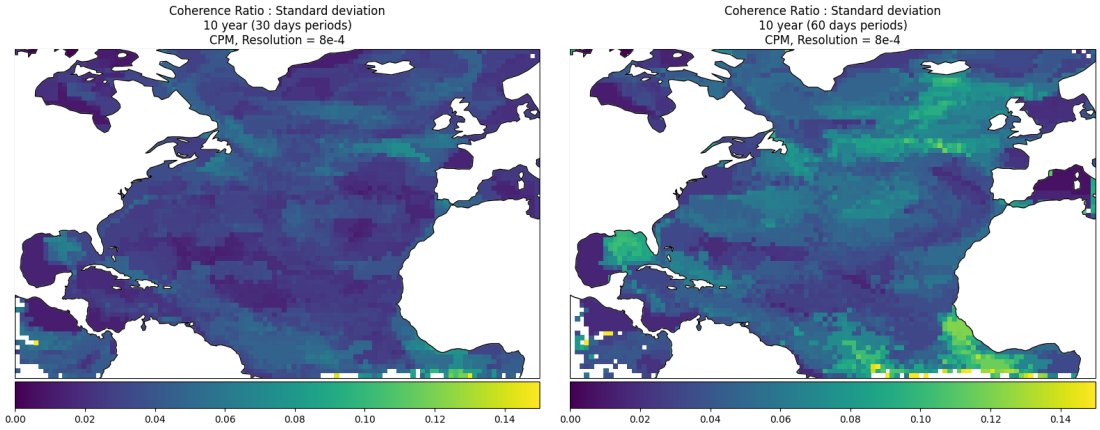
6.2 (Global) Mixing parameter

We introduced a measure on how low is the connectivity between different cluster, now we should comment on how well the clusters are mixed internally. A metric for that is the mixing parameter. [21]

The first step to determine this metric is to define a new transport matrix that describes the flow starting and ending in each cluster. Such a matrix is given by Equation 6.3, it adapts the weight of the previous transition matrix : the boxes not in the clustering solution are not considered and the new transition matrix $\mathbf{R}$ is also normalized.

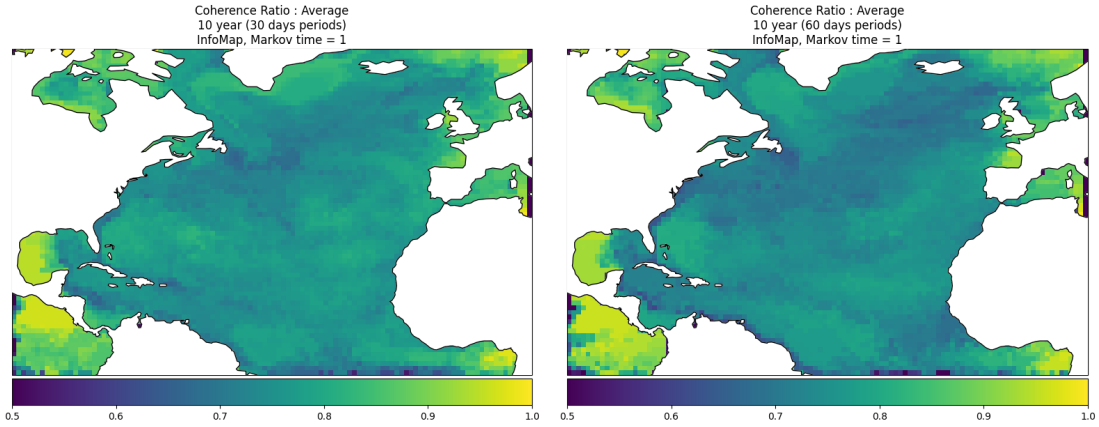
$$\mathbf{R}(t_0, \tau|k) = \frac{\mathbf{P}(t_0, \tau)_{ij}}{\sum_{k \in I_k} \mathbf{P}(t_0, \tau)_{ik},\ i, j \in I_k} \tag{6.3}$$

From that we can define the actual mixing parameter ν_k: the sum on all nodes included in the cluster k of the entropies associated to the transition probabilities : $-\sum_{ij \in I_k} \mathbf{R}(t_0, \tau|k)_{ij} \log \mathbf{R}(t_0, \tau|k)_{ij}$, normalized by $M_k \log M_k$, see Equation 6.4.



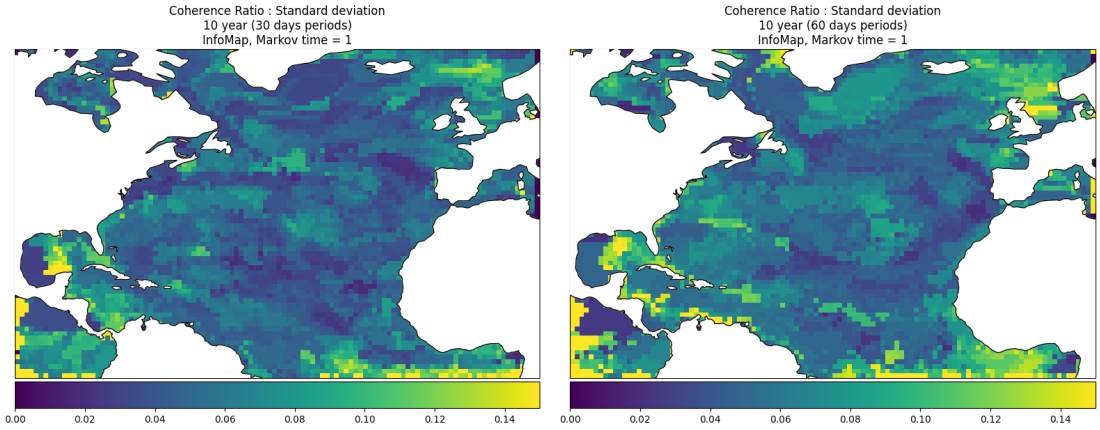
(a) 30 days (b) 60 days

Figure 6.3: Standard deviation of local coherence ratio on 10 years - CPM



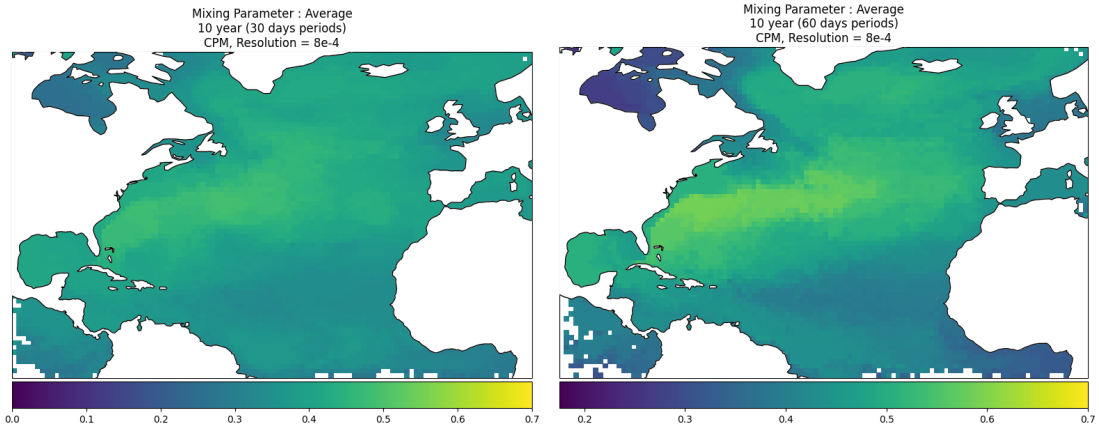
(a) 30 days (b) 60 days

Figure 6.4: Average local coherence ratio on 10 years - Infomap



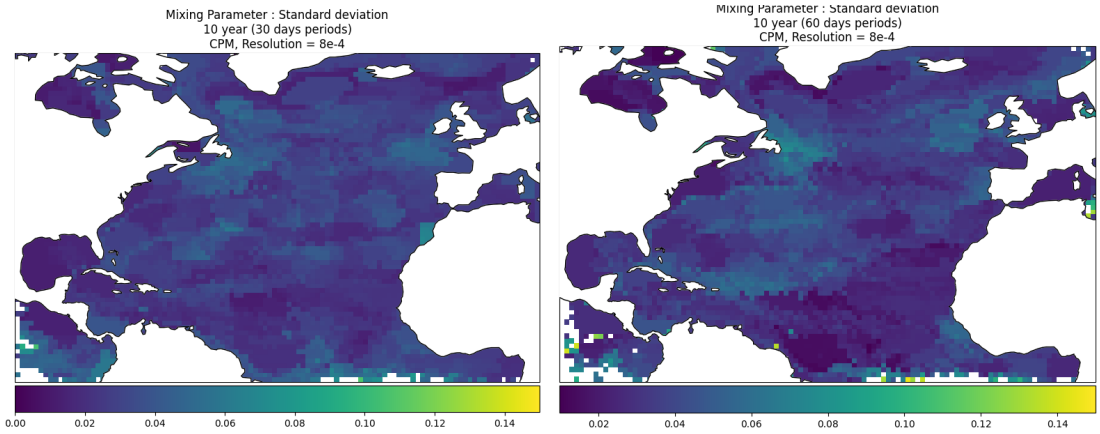
(a) 30 days (b) 60 days

Figure 6.5: Standard deviation of local coherence ratio on 10 years - Infomap



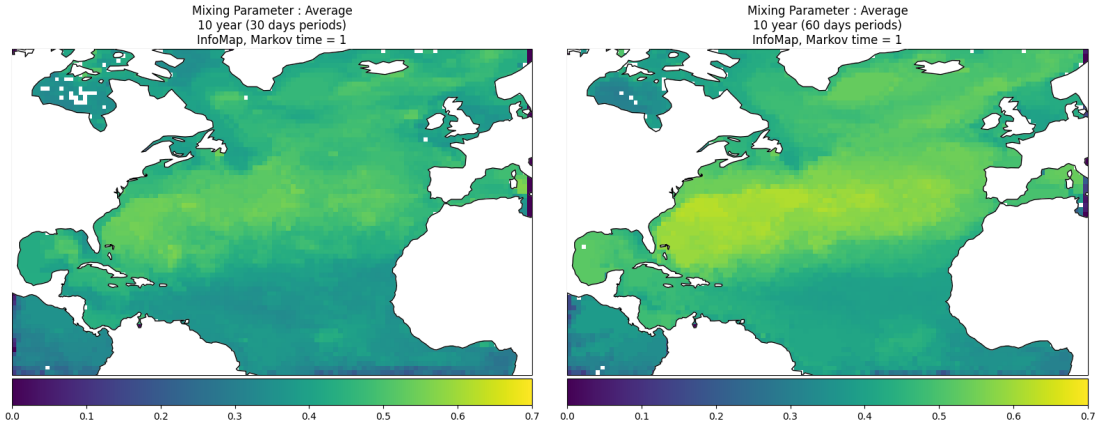
(a) 30 days (b) 60 days

Figure 6.6: Average local mixing parameter on 10 years - CPM



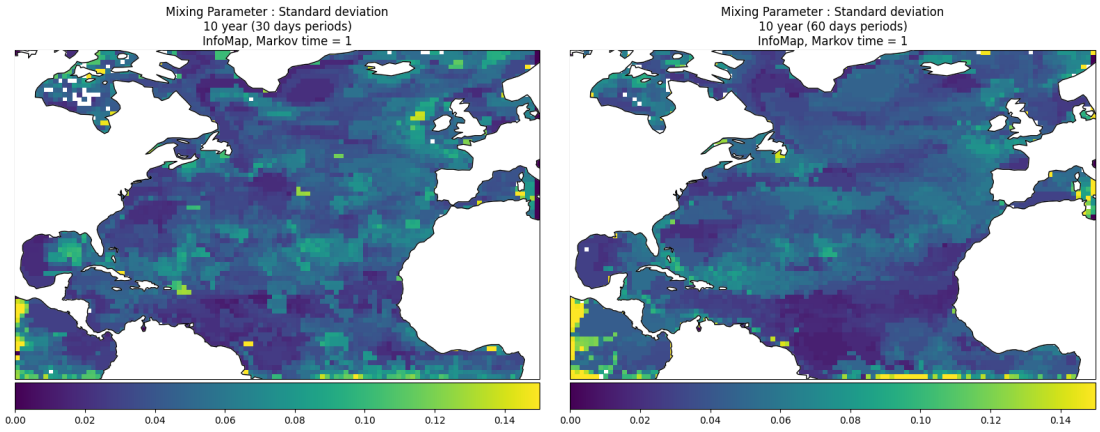
(a) 30 days (b) 60 days

Figure 6.7: Standard deviation of local mixing parameter on 10 years - CPM



(a) 30 days (b) 60 days

Figure 6.8: Average local mixing parameter on 10 years - Infomap



(a) 30 days

(b) 60 days

Figure 6.9: Standard deviation of local mixing parameter on 10 years - Infomap

$$\nu_k = \frac{-\sum_{ij \in I_k} \mathbf{R}(t_0, \tau|k)_{ij} \log \mathbf{R}(t_0, \tau|k)_{ij}}{M_k \log M_k} \tag{6.4}$$

A global version of this metric is computed in the same way of the coherence ratio Eq 6.5:

$$\nu = \frac{\sum_{k=1}^{p} \nu_k M_k}{M} \tag{6.5}$$

6.2.1 Comparison of the mixing parameters

For a time step of 30 days, CPM has a global mixing parameter of $\nu = 0.382$ with a standard deviation of 0.0084 for which Infomap scores $\nu = 0.426$ and 0.0087. And for the time step of 60 days, we have $\nu = 0.453$ with standard deviation 0.015 for CPM and $\nu = 0.471$ and standard deviation 0.0074 for Infomap.

We can conclude that Infomap scores better for the metric of mixing. In illustrations 6.6 and 6.8, you can see the local mixing parameter for CPM and Infomap. We notice that with bigger time step for the advection, the clusters are more mixed as the particles travel more between boxes. We also notice that the order of the maximum mixing parameter is 0.7 for both CPM and Infomap but Infomap detects larger well-mixed areas, explaining the better global metric.

Illustrations 6.7 and 6.9 are of the standard deviation of the local mixing parameter of the two methods. We can see that areas where the standard deviation

is particularly small are the areas of passage of the stronger currents. Because they have more impact on the Lagrangian flow network, those structure are often found in the different clustering solutions.

6.3 Persistence of boundaries

In this section we want to analyze the persistence of the clusters boundaries. In other words, giving clustering solutions at initial time t_0, the 1st January of years going from 2009 to 2019, we want to illustrate where the boundaries of clusters fall more often as they are supposed to coincide with well-known oceanic features. [17].

Those illustrations are computed by giving the nodes (or boxes) of the domain a value of 1 if they are at the boundary of a cluster or 0 otherwise, we do that for each of the solutions for a given method and a given time step of advection and sum the results and normalize them. The illustrations for a time step of 30 days are in Figure 6.10 for the CPM method and Figure 6.11 for the Infomap method. For 60 days, they are in Figure 6.12 for the CPM method and Figure 6.13 for the Infomap method.

For a time step of 30 days, we notice Infomap seems to cluster more small communities than CPM (this will be comfirmed in the next section). For oceanic features Infomap demarks more the Gulf Stream, the North Atlantic Drift and the Labrador Current for instance. That is to say CPM doesn't have bad performances as the difference in persistence is small.

For the longer time step of 60 days, note that communities are getting bigger and boundaries around notable oceanic feature are even more persistent. When we give more time for the particles to move, they are going further and thus highlighting more *barriers to flow*.

6.4 Distributions of cluster size

Throughout this dissertation, we spoke of scales and the limitations of both of our methods. In this section, we tried to quantify the resolutions the two methods are able to capture. For a time step of 30 days, you can see on Figure 6.14 the proportions of nodes (y-axis) falling into each size of clusters (x-axis). We observe that Infomap, which is subjected to the resolution limit, in our case is capable of detecting communities as small as CPM detects. Furthermore, more small communities are detected with the Infomap method, leading to believe that the

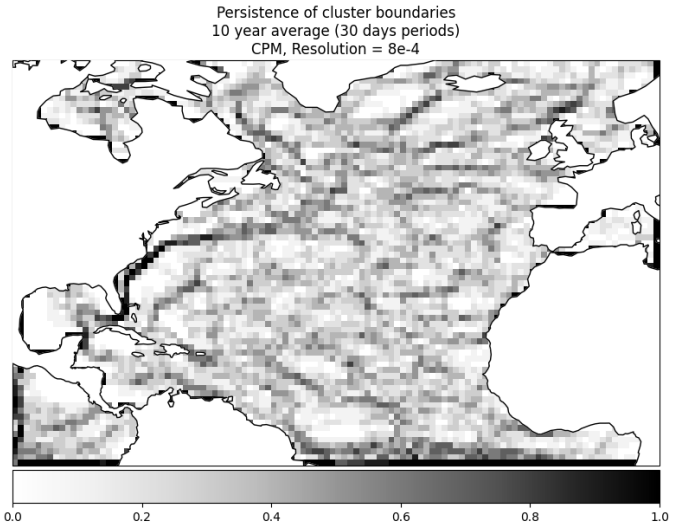


Figure 6.10: Persistence of clusters boundaries over 10 years, 30 days - CPM

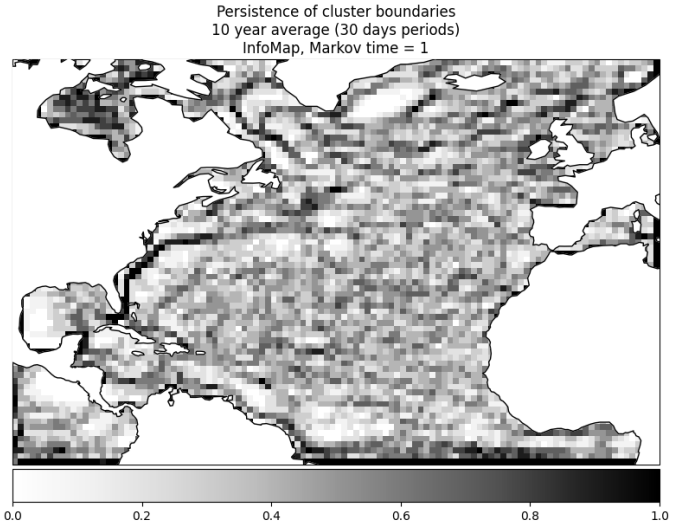


Figure 6.11: Persistence of clusters boundaries over 10 years, 30 days - Infomap

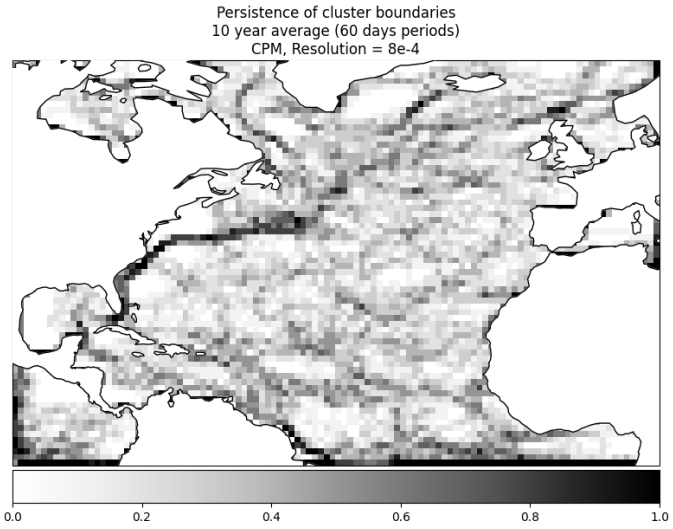


Figure 6.12: Persistence of clusters boundaries over 10 years, 60 days - CPM

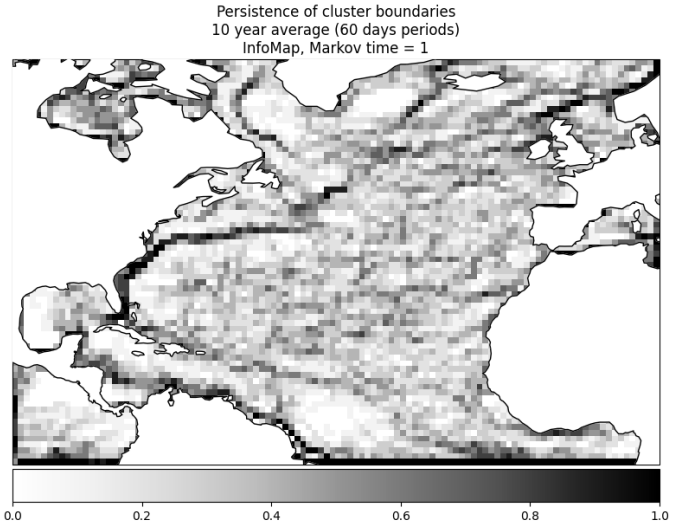


Figure 6.13: Persistence of clusters boundaries over 10 years, 60 days - Infomap

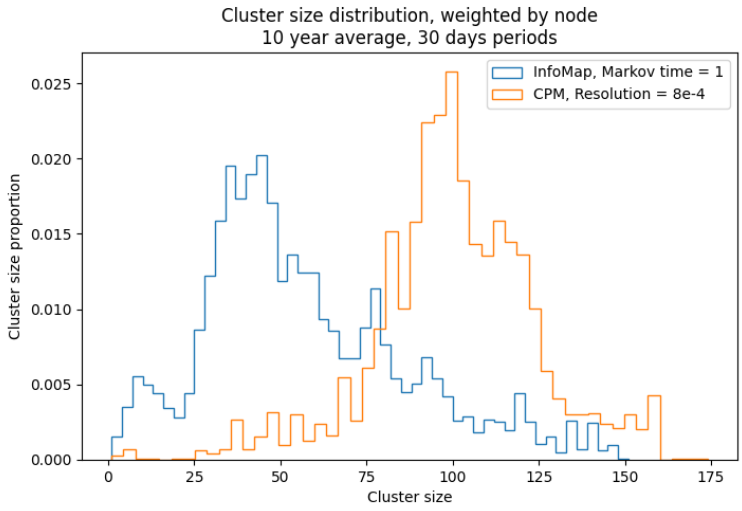


Figure 6.14: Cluster size distribution for 30 days

structure of the graph is such that Infomap escape the resolution limit in this case.

However, on the same figure, you'll see that CPM is able to detect larger communities than Infomap. Is it proof that Infomap suffers from a notable field-of-view limit ? In further work this should be investigated. However, in Figure 6.15 representing the same metric but with a time step of 60 days, we observe that it is Infomap who is able to detect larger communities. We conclude that the field of view limit is not measurable in this way.

A last observation is about the shape of the distribution. While CPM keeps a relatively similar distribution, Infomap goes toward a more uniform distribution when comparing 30 days with 60, reminding us of a diffusion process.

6.5 Reliability of the metrics at the border

We finish this chapter by mentioning that at the border, you'll often find missing values, reoccurring structures, noticeably high or low metrics, etc. This should not be taken into account because those are the areas where the particles escape the domain and are deleted from the analysis. This is also the place of many boundaries simply by construction, when seeing persistent boundaries at the border, note that it does not have any physical meaning.

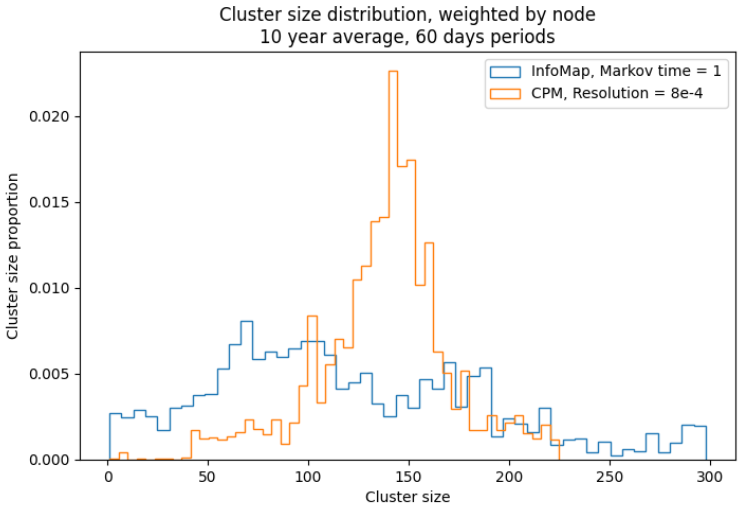


Figure 6.15: Cluster size distribution for 60 days

Chapter 7

Conclusions and perspectives - *For further work...*

This dissertation aimed at clustering the surface of the ocean. We learned the challenge of clustering real networks. In order to tackle this challenge, an effort has been made on modelling the flow accordingly. Of course you could argue that it is not ideal and you'll be right. As stated in Chapter 2 on Lagrangian flow network, the model as it is, is not divergent-free. Also, the particles are able to leave the domain and we do not take into account the ones entering the domain or reentering the domain. We are aware of the limitations of the modelling but as evidenced by the clusters obtained, while the model is not ideal it is good enough to be able to recover some interesting oceanic features.

We reported the main takeaways of the research done on community detection. Because there is several definition of communities, it was important to find methods that were well suited to our application and could uncover subgraphs properly. We had to find methods that were pattern-based (in order to take into account the directed edges) and capable of detecting largely varying scales.

Once this overview on community detection was done, we decided on comparing the Constant Potts model with Infomap. For further work, the study of how to combine the knowledge mined with the 2 techniques would be insightful. We left the discussion of their comparison on an open question : how do we quantify the field-of-view limit ? Indeed, while both of the method do not suffer from resolution limit on this particular case, we are not able to assess if the methods tend to overpartition or not.

All the clustering solutions have been found with an initial time being the first of January throughout the year 2010 until 2019, this was a conscious choice, in order to avoid seasonal variability. In further work, it seems interesting to use the

methods of clustering on several initial time and try to uncover a cyclic pattern. Analyzing trends might give us clustering solutions that are closer to the ground truth.

Even tough we were not able to distinguish a clear winner in the two methods used, one being more coherent, the other more mixed, we are satisfied by the results as they uncover main oceanic features well and have really good quality metrics.

Bibliography

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, oct 2008.

[2] Ludvig Bohlin, Daniel Edler, Andrea Lancichinetti, and Martin Rosvall. Community detection and visualization of networks with the map equation framework. In Ying Ding, Ronald Rousseau, and Dietmar Wolfram, editors, *Measuring Scholarly Impact: Methods and Practice*, pages 3–34. Springer International Publishing.

[3] Mark Carr, C. Woodson, Olivia Cheriton, Daniel Malone, Margaret Mcmanus, and Peter Raimondi. Knowledge through partnerships: Integrating marine protected area monitoring and ocean observing systems. 9.

[4] J.-C. Delvenne, S. N. Yaliraki, and M. Barahona. Stability of graph communities across time scales.

[5] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, feb 2010.

[6] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[7] Mikael L. A. Kaandorp, Stefanie L. Ypma, Marijke Boonstra, Henk A. Dijkstra, and Erik van Sebille. Using machine learning and beach cleanup data to explain litter quantities along the dutch north sea coast. 18(1):269–293. Publisher: Copernicus GmbH.

[8] Gautier Krings and Vincent D. Blondel. An upper bound on community size in scalable community detection, 2011.

[9] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. 80(5):056117. Publisher: American Physical Society.

[10] Diane Z. M. Le Gouvello, Michael G. Hart-Davis, Björn C. Backeberg, and Ronel Nel. Effects of swimming behaviour and oceanography on sea turtle hatchling dispersal at the intersection of two ocean current systems. 431:109130.

[11] L. C. M. Lebreton, S. D. Greer, and J. C. Borrero. Numerical modelling of floating debris in the world's oceans. 64(3):653–661.

[12] Fragkiskos D. Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. 533(4):95–142.

[13] Günce Keziban Orman, Vincent Labatut, and Hocine Cherifi. Towards realistic artificial benchmark for community detection algorithms evaluation. 9(3):349.

[14] Claire B. Paris, Matthieu Le Hénaff, Zachary M. Aman, Ajit Subramaniam, Judith Helgers, Dong-Ping Wang, Vassiliki H. Kourafalou, and Ashwanth Srinivasan. Evolution of the macondo well blowout: Simulating the effects of the circulation and synthetic dispersants on the subsea oil transport. 46(24):13293–13302. Publisher: American Chemical Society.

[15] Daan Reijnders, Erik Jan van Leeuwen, and Erik van Sebille. Ocean surface connectivity in the arctic: Capabilities and caveats of community detection in lagrangian flow networks. 126(1):e2020JC016416. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2020JC016416.

[16] Marcel Ricker, Jens Meyerjürgens, Thomas H. Badewien, and Emil V. Stanev. Lagrangian methods for visualizing and assessing frontal dynamics of floating marine litter with a focus on tidal basins. The Handbook of Environmental Chemistry, pages 1–36. Springer.

[17] Vincent Rossi, Enrico Ser-Giacomi, Cristóbal López, and Emilio Hernández-García. Hydrodynamic provinces and oceanic connectivity from a transport network help designing marine reserves. 41(8):2883–2891. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/2014GL059540.

[18] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, jan 2008.

[19] Martin Rosvall, Jean-Charles Delvenne, Michael T. Schaub, and Renaud Lambiotte. Different approaches to community detection. In *Advances in Network Clustering and Blockmodeling*, pages 105–119. John Wiley & Sons, Ltd. Section: 4 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119483298.ch4.

[20] Michael T. Schaub, Jean-Charles Delvenne, Sophia N. Yaliraki, and Mauricio Barahona. Markov dynamics as a zooming lens for multiscale community detection: Non clique-like communities and the field-of-view limit. 7(2):e32210. Publisher: Public Library of Science.

[21] Enrico Ser-Giacomi, Vincent Rossi, Cristobal Lopez, and Emilio Hernandez-Garcia. Flow networks: A characterization of geophysical fluid transport. 25(3):036404.

[22] Christopher Thomas. Modelling marine connectivity in the great barrier reef and exploring its ecological implications.

[23] V. A. Traag, P. Van Dooren, and Y. Nesterov. Narrow scope for resolution-limit-free community detection. 84(1):016114.

[24] V. A. Traag, L. Waltman, and N. J. van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1), mar 2019.

[25] V.A. Traag. Algorithms and dynamical models for communities and reputation in social networks.

[26] Erik van Sebille, Stephen M. Griffies, Ryan Abernathey, Thomas P. Adams, Pavel Berloff, Arne Biastoch, Bruno Blanke, Eric P. Chassignet, Yu Cheng, Colin J. Cotter, Eric Deleersnijder, Kristofer Döös, Henri F. Drake, Sybren Drijfhout, Stefan F. Gary, Arnold W. Heemink, Joakim Kjellsson, Inga Monika Koszalka, Michael Lange, Camille Lique, Graeme A. MacGilchrist, Robert Marsh, C. Gabriela Mayorga Adame, Ronan McAdam, Francesco Nencioli, Claire B. Paris, Matthew D. Piggott, Jeff A. Polton, Siren Rühs, Syed H. A. M. Shah, Matthew D. Thomas, Jinbo Wang, Phillip J. Wolfram, Laure Zanna, and Jan D. Zika. Lagrangian ocean analysis: Fundamentals and practices. 121:49–75.

[27] A. C. Vaz, K. J. Richards, Y. Jia, and C. B. Paris. Mesoscale flow variability and its impact on connectivity for the island of hawai'i. 40(2):332–337. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2012GL054519.

[28] Zhao Yang, Juan I. Perotti, and Claudio J. Tessone. Hierarchical benchmark graphs for testing community detection algorithms. 96(5):052311.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**