UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Master's degree course in Quantitative Biology

Calculation of long-term synergies of PFTs from OLCI and TROPOMI measurements

Interal supervisor:

Prof. Mattia Monga

External supervisor:

Dr. Leonardo Alvarado

Candidate:

Daniele Di Bella

Matr. 08376A

Academic Year 2022/2023

Contents

1. Introduction

1.1 Phytoplankton: a brief foreword

When we walk in the countryside and admire vast expanses of herbaceous plants, we are focusing our attention on a set of species that fulfil a specific ecological role: they are producers, meaning they appropriate atmospheric carbon dioxide (CO_2) molecules and, through photosynthesis, combine them into longer molecules, such as glucose ($C_6H_{12}O_6$), which they use to obtain the energy necessary for survival. The production of the molecules necessary for survival by the same organism that uses them is a process called autotrophy, and not all organisms are able to perform this process: this is the case for consumers, which must obtain the long carbon chains from other living beings. Humans, for example, are consumers. If we were to find some fruit during our walk, we would pluck it from the producer and eat it. Through this process, called heterotrophy, we would obtain the molecules we need to survive.

Consumers are not all the same: some feed on producers and are therefore called primary consumers, while others feed on primary consumers, and are thus called secondary consumers, and so on, until we reach a species of consumers that is not consumed by anyone else. These latter are called apex predators.

Both producers and consumers split the carbon molecules they procure to obtain energy: in the process, new carbon dioxide molecules are created and released into the environment, which will be used by producers for new synthesis processes. This all sounds like a complete cycle, but it is not. With only producers and consumers, the existence of an ecosystem would become quite difficult. In fact, to operate the synthesis processes, producers need other inorganic molecules besides carbon dioxide, and that is taken care of by detritivores and decomposers. The waste products of consumers (such as feces, or their bodies) along with those of producers, represent the food source for detritivores and decomposers, which transform organic matter into inorganic matter available to producers for their synthesis operations.

If during our walk in the countryside we focused on a specific producer species and were able to reconstruct its fate, that is, understand which species are part of one of the patterns in Figure 1 – in which decomposers and detritivores are missing –, we would have reconstructed

one of the food chains that characterize the ecosystem in which we are immersed. By intersecting all the identifiable food chains in the ecosystem, we would obtain a directed graph called a food web (Pimm *et al.*, 1991) – the center of Figure 1 represent a very small food web). This tool is essential for understanding an ecosystem as it allows us to understand in which direction energy flows inside the system, in other words, "who depends on whom" for their survival.

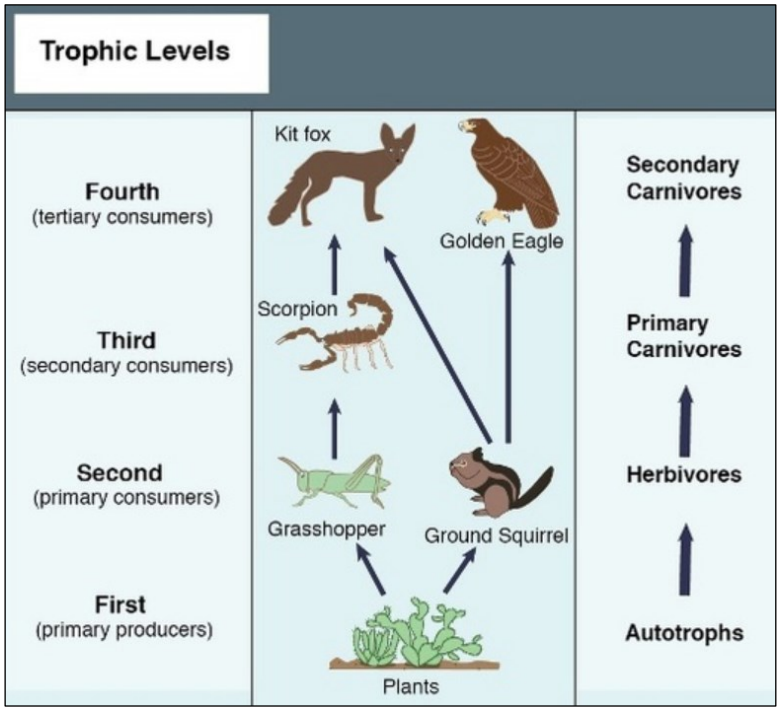

Figure 1 | **An hypothetical small food web.** On the left are the trophic levels at which the organisms shown in the center of the figure are placed: these are categories used to classify the type of nodes present in the food web graph. On the right, there are other terms to define the above nodes, based on the type of diet defined by their trophic relationships. (Scitable, n.d.)

As terrestrial animals, we can immediately identify the producers of a terrestrial ecosystem: plants. But which are the primary producers in aquatic ecosystems? As for terrestrial ecosystems, the producers of an aquatic ecosystem are photosynthetic organisms of many sizes (Sigman and Hain, 2012). Some are benthonic, which means that they are fixed to the bottom of the body of water in which they reside, but in most cases, they are mobile microscopic organisms which are not fixed to any support, but at the same time they cannot swim. This characteristic puts them in the plankton category, which name comes from the Greek word meaning "drifter" or "wanderer": unable to swim, these organisms are at the mercy of the currents.

Not all plankton are producers, since the term plankton is an umbrella term that encompasses all organisms transported by tides and currents, often including organisms such as crustaceans and jellyfish, which are consumers. Moreover, numerous species of consumers may have planktonic juvenile stages, while the adult form can resist currents (Mansur *et al.*, 2014). Thus, there is a clear need to possess more stringent classification tools to navigate the vast variability that the term "plankton" implies.

Traditionally, to overcome this problem, the microscopic organisms that make up plankton have been divided into two major groups: phytoplankton (or microalgae) which are autotrophic organisms capable of producing the molecules they need to sustain themselves through photosynthesis, and zooplankton (in which, based on size, we can distinguish mesozooplankton and protozooplankton), heterotrophic organisms that feed on phytoplankton. In recent years this classification has been modified by the identification of a new group of plankton, the mixoplankton, which can both operate autotrophy and heterotrophy (Glibert and Mitra, 2022). A schematic representation of the relationships between the different types of plankton and the rest of the oceanic food web is offered by Figure 2 below.

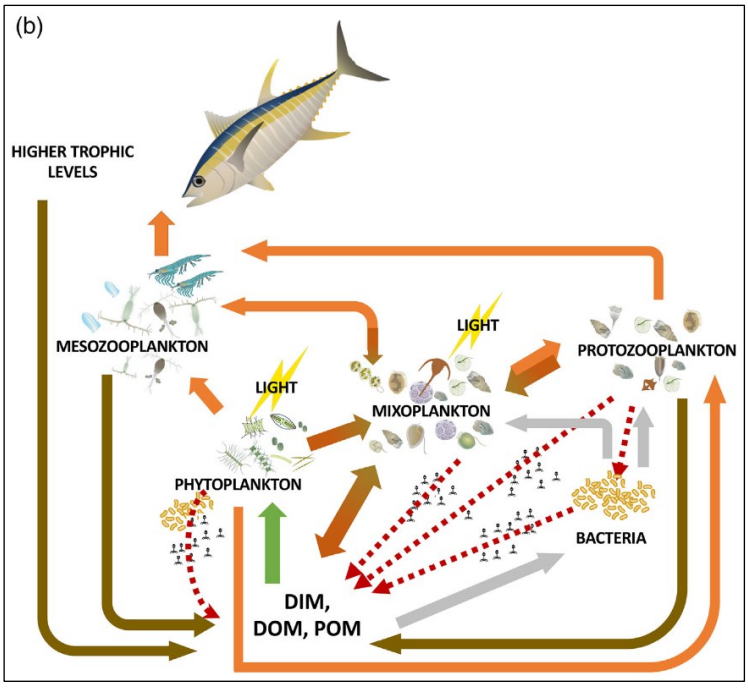

Figure 2 | **Model of interactions between the different types of plankton and the rest of the oceanic food web.** This scheme envisions phytoplankton as the primary producers using dissolved inorganic matter (DIM, green arrow) and light for carbon fixation. Orange arrows indicate grazing activities while brown arrows, excretion, and defecation. Orange-green gradient arrows depict the photo-phago-mixotrophic feeding strategies of mixoplankton. The microbial loop, depicted using gray arrows, portrays the multi-role of the bacteria: dissolved inorganic and organic matter (DIM and DOM, respectively) are taken up by bacteria that are in turn consumed by protozooplankton and mixoplankton. Not only bacteria are important in this scheme, but also viruses are (viral

processes are represented by the dashed red arrows). They infect bacteria, mixoplankton, phytoplankton and protozooplankton, in turn releasing particulate and dissolved organic matter. (Glibert and Mitra, 2022)

This thesis work focuses on phytoplankton, the study of which is fundamental because of their many prominent roles in marine ecosystems and the Earth system: the next chapter will provide a brief overview of these ecological functions and it will try to explain why our species should invest time and resources in the observation of such organisms.

1.2 Ecological roles of phytoplankton

As Figure 3 shows, several measurements conducted since the second half of the 20th century on different samples – like Antarctic ice samples (Neftel *et al.*, 1985) – clearly indicate how the era in which we live is characterized by a steady increase in the concentration of carbon dioxide in the atmosphere.

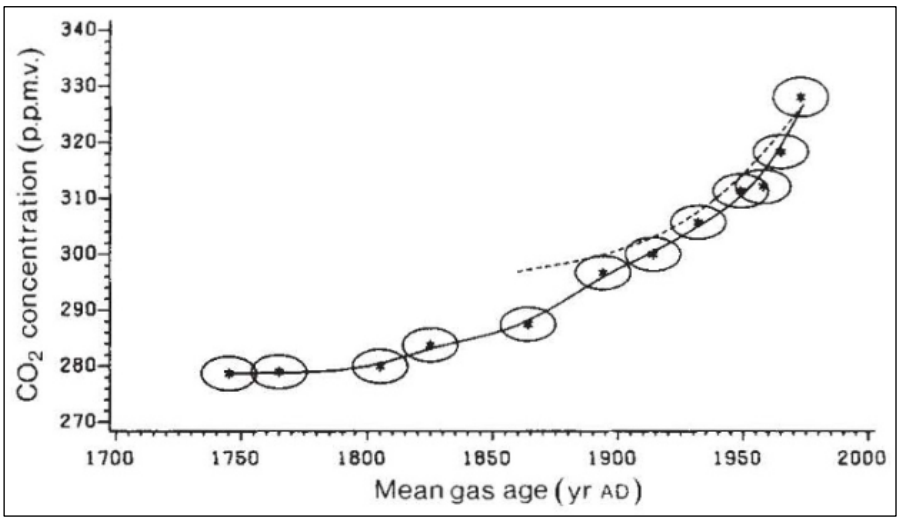

Figure 3 | **Measured mean CO_2 concentration in ice air bubbles plotted against the estimated mean gas age.** The dots depicted in this graph represent the concentration of CO_2 in gas bubbles trapped inside Antarctic ice samples analyzed by Neftel and colleagues: ice deposits are formed by gradual deposition of snow layers that compact into ice. During this process, small air bubbles, identical in composition to the rest of the air on the planet at the time of deposition, become trapped in the ice. By being able to access these bubbles, the evolution of the composition of Earth's atmosphere can be estimated. Neftel and his team analyzed samples obtained by ice coring, in other words, cylinders of ice extracted from the Antarctic ice sheet by drilling: in this way they gained access to a series of air bubbles deposited over several hundred years. They dated them and measured the average CO_2 concentration within them. From this they obtained the graph in the figure, where the horizontal axis of the ellipses indicates a close-off time interval of 22 years, and the vertical one reports the uncertainties of the concentration measurements. The dotted line represents the atmospheric CO_2 concentration calculated with the model they developed, assuming only CO_2 input from fossil fuel. (Neftel *et al.*, 1985)

This is due to various types of human activities that rely on energy production through the burning of fossil biomass, and which, increasing dramatically in recent decades, have greatly increased the amounts of CO_2 in the atmosphere compared to the pre-industrial era. According to measurements conducted at the Mauna Loa Observatory by staff of the U.S. National Ocean and Atmospheric Administration (NOAA) agency, the current concentration of carbon dioxide in the atmosphere is close to the frightening value of 424.55 ppm (Figure 4).

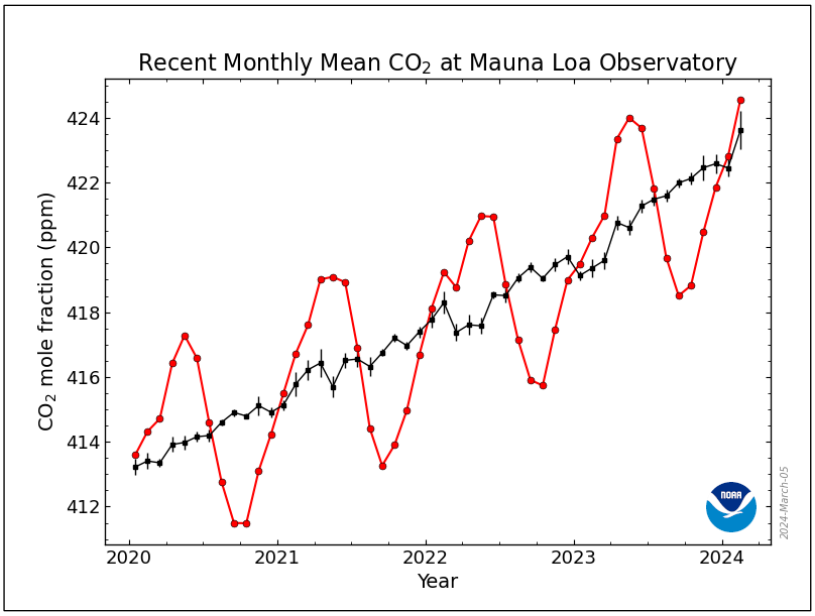

Figure 4 | **Monthly mean atmospheric CO_2 concentration measured at Mauna Loa observatory.** The red lines and symbols represent the monthly mean values, centered on the middle of each month, while the black lines and symbols represent the same, after correction for the average seasonal cycle. Seasonal cyclic variation is corrected by averaging the seasonal mean over a moving window with width equal to seven seasonal cycles and centered on the month of interest. The vertical lines protruding from the black squares are an indication of the uncertainty associated with the measurement. (Global Monitoring Laboratory, n.d.)

This increase does not occur without consequences; in fact, it involves several changes in numerous components of the Earth system that, as they change, trigger a series of domino effects that affect the entire rest of the planet, in many cases affecting the well-being of our species. One of the many examples that could be brought to support this assertion are the serious consequences that the increase in the concentration of CO_2 in the atmosphere has on the global temperature, and the consequences that this increase has on crops yields. In a beautiful study published in 2019 it is shown that the increasing amount of carbon dioxide in the atmosphere influences the rain frequency and intensity in West Africa, giving more water to the crops and thus increasing their yield. Up to here, it would almost seem that the increase in carbon dioxide leads to positive phenomena. The problem is that it also correlates with an increase in global temperatures due to the greenhouse effect (Zhong and Haigh, 2013), and this increase prevents water from being trapped in the soil, thus available to plants, and places plants under heat stress. These effects exceed the benefit reported above, causing a net decrease in the yield of the West African crops, worsening the continent food insecurity situation (Sultan *et al.*, 2019). Although the study just mentioned focused on the West African region, these kinds of phenomena also affect other regions of the planet, in fact, similar studies have also been conducted in other regions such as the United States' Midwest (Jin *et al.*, 2018) and Europe (Supit *et al.*, 2012).

For these reasons, an overview of the roles that phytoplankton play within the earth system must necessarily begin with an account of the close link that, as photosynthetic organisms, they have with atmospheric carbon dioxide.

The difference in the concentration of carbon dioxide that exists between the Earth's atmosphere and the oceans causes diffusion phenomena by which CO_2 molecules reach the surface of the oceans and dissolve among the waves. Although interaction with water does not always leave carbon dioxide molecules unchanged – for example, due to hydration reactions, they can take forms such as that of the bicarbonate ion ($CO_2 + H_2O \leftrightarrow H_2CO_3 \leftrightarrow H^+ + HCO_3^-$) – phytoplankton are able to use many types of inorganic carbon dissolved in water as building blocks for organic molecules. Once fixed in molecules of biological origin (biogenic molecules), carbon atoms are subject to the fate that such molecules incur; for example, if several atmospheric carbon atoms are fixed in a molecule used for structural purposes by the phytoplankton that performed photosynthesis, and the cell dies without being subjected to predation processes, and precipitates to the bottom of the water basin in which it is contained, the carbon atoms are said to have been "sequestered" from the atmosphere and deposited in sediments in which they will remain for several hundred years.

Collectively, all aquatic processes involving the sequestration of atmospheric carbon through the fixation of it into molecules of biological origin and the transport of the molecules to the bottom of the water basin are identified through the concept of biological carbon pump (BCP), which first appeared in 1985 (Volk and Hoffert, 1985). Although above only one example of how a biogenic molecule containing atmospheric carbon atoms can be transported to the depths of the oceans was shown, many more have been identified over the years, and until a few years ago, at least three different types of BCP were recognized (Le Moigne, 2019):

- Gravitational BCP, i.e., the totality of all the phenomena that include the precipitation of particulate organic carbon – which is the ensemble of suspended and sinking organic particles with size ≥ 0.2 μm (Kharbush *et al.*, 2020)– of planktonic origin: the example proposed above about the dead phytoplankton cells falls into this category.
- Mixing BCP, i.e., the totality of physical mechanisms that, through the movement of water masses, transport downward the carbon fixed at the surface.
- Migration BCP, i.e., the transport of carbon to the depths by migrating fish and plankton: different species of plankton and fish move to the surface at night – to feed by limiting the likelihood of being subject to predation – and return to deeper areas

during the day. Here, by defecating, they release some of the carbon they have recovered at the surface.

The different types of BCPs do not contribute equally to the total amount of carbon sequestered by the oceans but, as Nowicki and colleagues show (Nowicki *et al.*, 2022), the amount of carbon sequestered by different BCPs, as well as the amount of time the sequestered carbon spends in sediments, varies depending on the parts of the planet being considered (Figure 5).

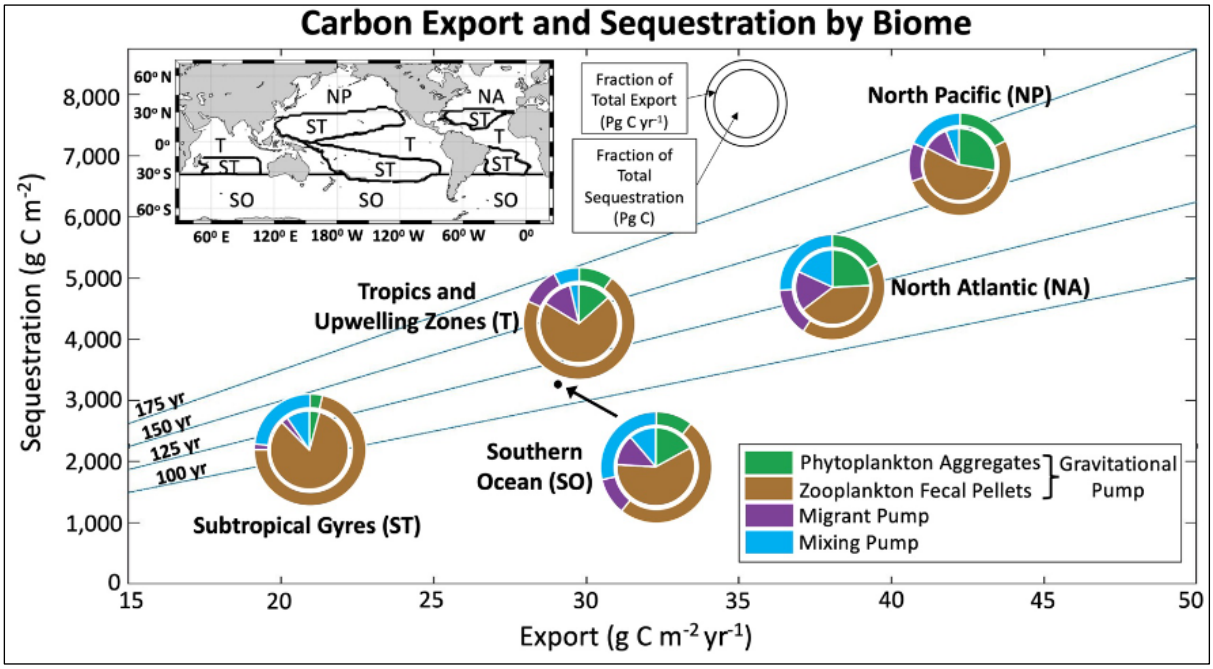

Figure 5 | **Different contributions of different BCPs to the total carbon sequestered in different parts of the planet.** The total number of grams of carbon per square meter sequestered by each of the five ocean biomes defined in the figure is shown on the y-axis depicted in the figure, while the x-axis shows the total number of grams of carbon per square meter exported each year from each of the above ocean biomes. Carbon is exported when it is respired by organisms, typically zooplankton, in the lower layers of the water column, where it can remain from several years to several hundred years. Outer rings represent the fractional contribution of each type of BCP to total export in the corresponding region, with inner pies representing the fractional contribution of each type of BCP to total sequestration in that region. The location of each pie chart on the trend lines shows the average sequestration time (which is the total carbon sequestration divided by total carbon export) for each biome. (Nowicki *et al.*, 2022)

In recent years, new types of biological carbon pumps have been recognized (Claustre *et al.*, 2021), but in each of these what is at the center is the photosynthetic activity of phytoplankton. It is the fixation of atmospheric carbon through photosynthesis that gives rise to BCPs, through the activity of which the oceans and seas are transformed into what is defined as carbon "sinks". To provide some indicative figures of the magnitude of these phenomena, it is good to mention that about 50 percent of the primary production (i.e., in simpler terms, photosynthesis) that takes place each year on planet Earth is operated by phytoplankton (Losa *et al.*, 2017), and

BCPs that are grafted onto this phenomenon sequester 11 Gt of carbon each year from the atmosphere (Sanders *et al.*, 2014; Basu and Mackey, 2018).

The impact of this on the earth system is enormous: the constant removal of such amounts of CO_2 from the atmosphere affects global temperatures and climate phenomena to such an extent that phytoplankton are a consideration when proposing hindcast and forecast simulations (Bracher *et al.*, 2022). However, the climate changes to which our planet is subjected can seriously threaten the stability of mechanisms such as BCPs that are dependent on the physiology of phytoplankton and their community structure and distribution (Basu and Mackey, 2018), which is why, constantly having accurate information regarding these phenomena is of crucial importance, especially in these times.

The reasons that lend importance to the observation of phytoplankton communities' structure and distribution, however, do not end with their relationship to carbon in the atmosphere, but extend by virtue of numerous other characteristics. As the basis of aquatic trophic networks, for example, phytoplankton is the limiting factor for the growth and abundance of all aquatic organisms. In fact, biomass production is not only dependent on atmospheric carbon fixation, but also on the production of fatty acids by phytoplankton (Budge *et al.*, 2014). Fatty acids (FA) are essential for the growth of all vertebrates, both marine and terrestrial, and since the only source of fatty acids in the oceans is phytoplankton, these organisms must obtain essential FA either directly, by feeding on phytoplankton, or indirectly, through the food web.

The direct correlation between the primary production and synthesis of fatty acids by phytoplankton and the size of fish stocks lends great commercial interest to these organisms: knowing their distribution and community characteristics is crucial for the fishing industry, but not only. Public health administrators are also interested in this kind of information since terrestrial vertebrates such as *Homo sapiens* have a need to assimilate certain fatty acids, such as eicosapentaenoic acid (EPA), represented in Figure 6, which they access by consuming organisms that have fed on phytoplankton. However, as shown by Budge and colleagues, due to the steady growth of human population, global production of EPA is in danger of no longer meeting the demand for this molecule, and this would prove to be a problem because of the many benefits it provides as hypotriglyceridemic and anti-inflammatory to prevent cardiovascular disease (Siriwardhana *et al.*, 2012).

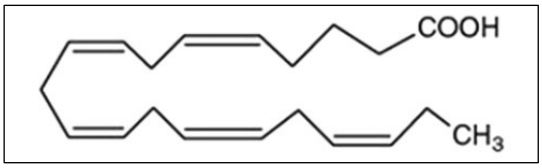

Figure 6 | **Carbon skeletal structure of eicosapentaenoic acid.** (Peter *et al.*, 2013)

Budge and colleagues point out that, so far, we have had only very limited knowledge of the amount, distribution, and rate of fatty acid synthesis in the oceans, and given the impact of these molecules on human health, gaps of this kind need to be covered as soon as possible.

When it comes to phytoplankton, public health administrators have much to look out for, not just fatty acid production. In fact, phytoplankton proliferations, known as algal blooms (Figure 7), can prove to be dangerous for humans living near the coasts because of the toxins produced by the proliferating organisms: in this case the phenomenon falls into the category of what is called harmful algal bloom (HAB), while in the case where the algae should not produce toxins we speak of eutrophication, a phenomenon with serious ecological consequences, but not worrying from a point of view strictly related to public health.

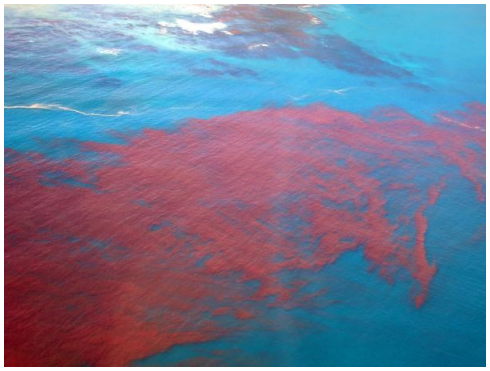

Figure 7 | **Portrait of an algal bloom.** (Intergovernmental Oceanographic Commission, n.d.)

Until 2016, the year of publication of the article from which the following information was taken (Grattan *et al.*, 2016), five types of diseases directly related to HABs were recognized: Ciguatera Fish Poisoning, Paralytic Shellfish Poisoning, Neurotoxic Shellfish Poisoning, Amnesic Shellfish Poisoning and Diarrheic Shellfish Poisoning. Each of these diseases arises after contact with poisoned fish or seafood, and for none of the toxins underlying these diseases were antidotes available, therefore, disease prevention was, and still is, of paramount importance in managing the risks associated with HABs. Over the years, several datasets have been compiled in which the characteristics and frequency of HAB events are recorded (Hallegraeff *et al.*, 2021), and these tools have proven to be very useful in producing time series

that allow policy makers to get a clear idea of the past and present trends of these phenomena (Figure 8) and hypothesize a future one.

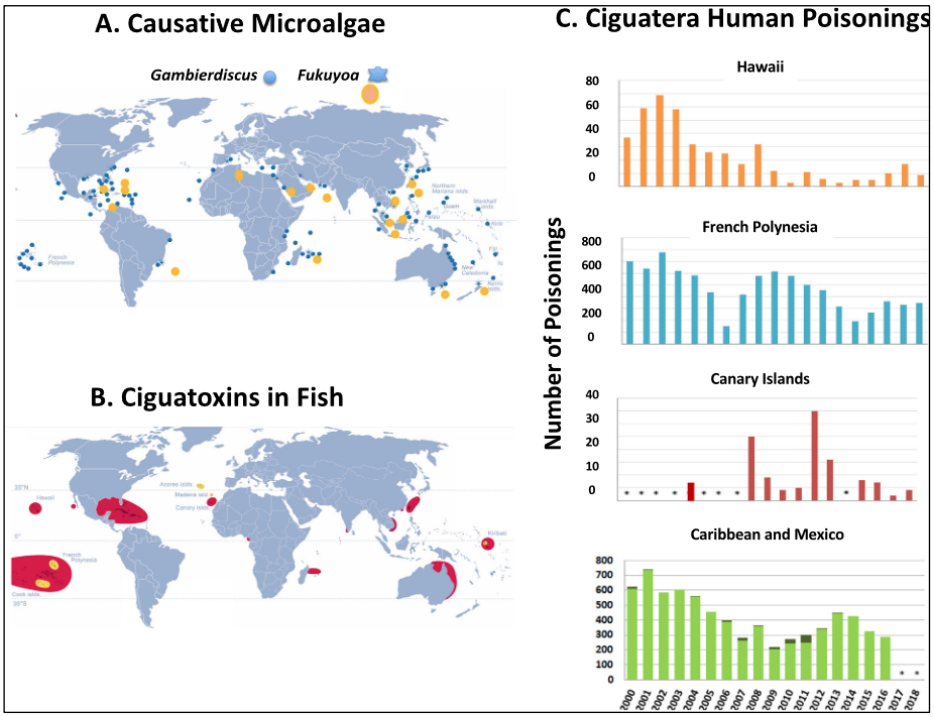

Figure 8 | **Time series representing the occurrence of cases from Ciguatera Fish Poisoning in different places on the planet.** Thanks to the creation of new datasets, it is possible to map the proliferation sites of certain toxic phytoplankton species – such as dinoflagellates of the genus *Gambierdiscus* and *Fukuyoa* that produce the toxins responsible for Ciguatera poisonings represented, respectively, by the blue dots and yellow dots in part A of the figure – and to record the presence of toxins in fish and shellfish (red dots and yellow dots in part B of the figure). Not only that, but time series can also be created to keep track of the occurrence of these poisoning cases in different parts of the planet, part C of the figure, and to get an idea of the trend of these phenomena over time. (Hallegraeff *et al.*, 2021)

Phenomena and studies such as those just mentioned make it clear that the study of phytoplankton communities' structure and distribution is an issue that not only interests basic research, but which is of immense applicative importance. It should be part of the agendas of public administration, which must ensure the health of citizens and the availability of ecosystem services for them. The necessary condition for this to happen, however, is the possession of accurate information on the distribution and composition of phytoplanktonic communities, but, unfortunately, "the large-scale patterns of [phytoplankton] diversity are not well understood and are often poorly characterized in terms of statistical relationships with factors such as latitude, temperature, and productivity" (Dutkiewicz *et al.*, 2020). Is it possible to fill this gap?

1.3 Phytoplankton Functional Types

The attempt to address the question expressed in the previous chapter has, over the years, brought to the surface the concept of phytoplankton functional type (PFT): when studying phytoplankton and their involvement in certain biogeochemical mechanisms occurring on the Planet, it is necessary to orient ourselves in the multitude of species included in that grouping. One strategy to do this is to identify groups of species based on the biologically mediated biogeochemical transformations they are able to perform (IOCCG, 2014). Such groups are called phytoplankton functional types.

Since the "functional type approach" recognizes that we cannot keep track of the evolution of the distribution of every phytoplankton species in the ocean, and that some dimensions of variation will not matter for some research questions (Irwin and Finkel, 2017), the use of the PFT concept allows for the conduct of inquiry processes with categories that are both manageable and more stringent than those generated by some definitions used in more general ecological contexts.

It is important to emphasize that these categories do not reflect the taxonomic divisions to which the species contained within them belong, as the use of taxonomy in these situations can be counterproductive: taxonomically related species may exhibit very different ecological adaptations, and taxonomically distant species may have evolved similar ecological roles (Salmaso *et al.*, 2015).

This thesis work is aimed at developing tools that can facilitate the understanding of phenomena concerning two PFTs, commonly referred to as nitrogen fixers – or cyanobacteria – and silicifiers – or diatoms – (IOCCG, 2014):

- Nitrogen fixers get their name from their ability to use nitrogen dissolved in water to operate photosynthesis processes. Normally, phytoplankton are only able to utilize nitrogen in the form of nitrate, nitrite, or ammonium, and since these compounds are present at low concentrations in the more superficial layers of water bodies, they are a limiting factor for phytoplankton growth. Nitrogen fixers can overcome this problem through the use of nitrogen dissolved in water. Generally, the species that are included in this PFT belong to the class of cyanobacteria (an example of which is given in Figure 9).

- On the other hand, the name of silicifiers derives from the silicon exoskeletons that surround them: to build such structures, silicifiers – a set to which species belonging to the class of diatoms are commonly ascribed – have a constant need for silicon. An example of a silicifier, a diatom, is portrayed in Figure 10.

Figure 9 | **Image of filamentous cyanobacterium.** The filaments in the picture are composed of cells of the cyanobacterium species *Arthronema africanum*, which image was obtained through 100-fold magnification operated by light microscope. (Damatac and Cao, 2022)

Figure 10 | **Image of diatom.** The black rectangle in the upper right of the image represents a scale bar equal to 10 μm, and the diatom species portrayed is *Diploneis smithii*. (Bonomo *et al.*, 2009)

The concept of PFT allows studies to be conducted with a more manageable set of categories than the one that the broad term phytoplankton offers but, pragmatically, how is it possible to get an idea of the distribution of different PFTs on our planet?

1.4 Satellite technology

Studying the distribution of PFTs on our planet is a complex matter. One could attempt to address it through a series of *in situ* measurements, that is, by going in person to different points on the planet and collecting water samples that would then be analyzed in the laboratory to determine how many and what kind of phytoplankton functional groups are contained there. However, this kind of approach would yield a very limited number of measurements, which would provide accurate information only relative to the sampling points considered, but without giving an idea of the global distribution of the PFTs of interest. Not to mention the temporal resolution: it would be constrained to the days on which the measurements were taken.

To overcome this problem, remote sensing, a set of techniques that can be defined as "noncontact recording of information from the electromagnetic spectrum by means of mechanical, photographic, numeric, or visual sensors located on mobile platforms" (Fussell & Rundquist, 1986), is used. This means, in other words, that instead of coming into direct contact with the object of interest, the interactions it has with electromagnetic radiation of various kinds are recorded, and from this the needed information is derived.

In the case of PFTs, since phytoplankton are photosynthetic organisms, thus rich in pigments (such as chlorophyll-a) that absorb light in the visible spectrum, the interaction of pigments with electromagnetic waves from the sun is recorded through the use of sensors mounted on satellites orbiting the Earth. The sensors "divide" the planet's surface into pixels, varying in area depending on the instrument considered, and, as they orbit the Earth, acquire values from each of these. In this way, the problem of *in situ* measurements is overcome because information is obtained for the entire planet, and with much better temporal resolution than the one that could be obtained by going to the field in person.

For several years now, ESA, the European Space Agency, has had a number of sensors mounted on satellites that, as they orbit the planet, record the electromagnetic radiation from the sun that is reflected and scattered by the surface of the seas and oceans: these radiance values are collected in datasets that are subsequently processed and analyzed from the ground, in order to obtain several types of information regarding our planet. As suggested by Bracher and colleagues (Bracher *et al.*, 2022), different information about the Earth's surface can be derived by retrieving different types of electromagnetic waves:

- Visible wavelength radiation allows for observing vegetation coverage extent and analyzing the surface of water bodies.
- Near-infrared wavelength radiation ($\lambda \sim 0.7$ to 1.3 μm) enable the assessment of vegetation health and other surface properties.
- Shortwave infrared (SWIR) wavelength radiation is useful for investigating the mineral composition and moisture content of the Earth's surface.
- Thermal infrared wavelength radiation ($\lambda \sim 3$ to 14 μm) is employed to measure the surface temperature and heat distribution of the planet.
- Radio waves with longer wavelengths than those in the thermal infrared range are ideal for exploring soil moisture content and ocean currents.

Visible wavelength radiation is the kind of radiation that can be useful for the studying of phytoplankton since chlorophyll-a absorbs it. This pigment is of particular importance because, given its presence in all phytoplankton species (and thus in all PFTs, clearly), chlorophyll-a concentration values can be used as a proxy to determine the concentration values of the totality of phytoplankton in an area of interest. However, if one is interested in the distribution of specific PFTs in an area of interest, obtaining the map of distribution of chlorophyll-a in that area is not sufficient: further analysis should be conducted to understand how much the different PFTs of interest contribute to the formation of these values. From there, distribution maps of phytoplankton functional groups can be obtained.

Retrievals of the visible light backscattered and reflected by the surface of Earth's water basins in order to gain insight on the distribution of different PFTs has been done along the years through the use of several instruments mounted on board of several satellites. This thesis work looked at the retrievals by OLCI and TROPOMI instrument mounted, respectively, on the Sentinel-3 satellite, Figure 11, and Sentinel-5P satellite, Figure 12, of ESA's Copernicus system.

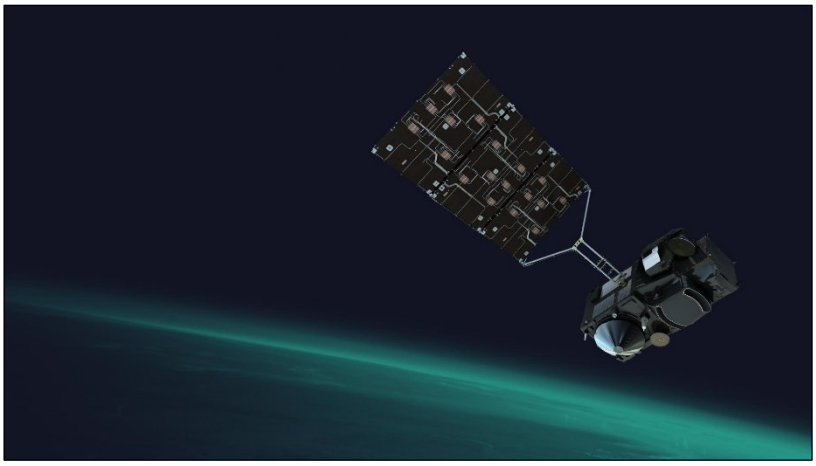

Figure 11 | **The Copernicus system's Sentinel-3 satellite.** (EUMETSAT, 2020)

Figure 12 | **The Copernicus system's Sentinel-5p satellite.** (ESA, n.d.)

Launched by ESA in 2016, OLCI (Ocean and Land Color Instrument) can capture ten bands within the visible spectrum, which are ten wavelength windows ranging from 0.4 to 0.7 µm. The number of bands that OLCI can observe is greater compared to its predecessor, ENVISAT MERIS (Sentinel Online, n.d.), representing a significant advancement in remote observation of phytoplankton distributions. Indeed, having an instrument that can capture multiple wavelength windows allows for better identification of PFTs: although distinguished through the distinction of certain characteristic pigments, these groups contain many molecules that absorb in a similar manner and can only be discerned by increasing the spectral resolution of the measuring instrument.

Despite these characteristics, the strength of an instrument such as OLCI, called multispectral, is not the amount of wavelengths that it can retrieve, which, as will be shown below, is limited compared to instruments like TROPOMI, but rather, the size of the pixels into

which it divides the planet: in the case where one wanted to use the instrument in full resolution mode, FR, one would be able to have pixels of 300 m × 300 m. Alternatively, in reduced resolution mode, RR, it detects pixels with sides equal to 1.2 km.

On the other hand, the TROPOMI instrument (TROPOspheric Monitoring Instrument) was launched on 13 October 2017 on board the Sentinel-5 Precursor, also known as Sentinel-5P, as the first Copernicus mission dedicated to monitoring our atmosphere. In fact, covering the global atmosphere every day with a spatial resolution as high as 7 km × 3.5 km for almost all spectral bands (Sentinel Online, n.d.), and a spectral resolution of 0.5 nm (Guanter *et al.*, 2015), TROPOMI is designed mainly to map a multitude of trace gasses, which affect the air we breathe and therefore our health and our climate. However, this instrument can also be employed for other purposes. In fact, it is not solely useful to measure the reflected and transmitted light from the atmosphere but, like OLCI, it can be used to obtain data from all the spheres of the planet: atmosphere, lithosphere, hydrosphere… The reason for its usefulness for the purposes of this thesis lies precisely in the possibility to employ it for measuring the concentration of PFTs in Earth's waters. In fact, TROPOMI is what it is called a hyperspectral instrument, and its characteristics exactly mirror those of OLCI: it can sample the wavelength range of 675 nm to 775 nm every 0.5 nanometer, which means that it has a very high spectral resolution, but the area of pixels it detects does not allow for as fine a spatial resolution.

Both OLCI and TROPOMI are two spectrometers that measure the reflected and backscattered solar radiation from various components of planet Earth. This means that the data obtained from these instruments undergo several conversions in order to transform them into values indicating the concentrations of PFTs in specific areas. Following a conversion, it is said that data "go through a level". Below, I report a simple scheme of these operations:

$$\text{OLCI} \rightarrow \text{L1A} \rightarrow \text{L1B} \rightarrow \text{L2: Chl-a} \rightarrow \text{L2A: [PFTs]}$$

$$\text{TROPOMI} \rightarrow \text{L1A} \rightarrow \text{L1B} \rightarrow \text{L2A: [PFTs]}$$

The instruments provide data referred to as level 1A (L1A) data, which immediately undergo a "dark current" correction, which is the subtraction of the noise generated by the electronic components of the satellites. In fact, despite orbiting in a vacuum, the satellites hosting the instruments are subject to unstable conditions. For instance, depending on their position relative to the Sun, they may experience temperature variations that affect the functioning of the electrical circuits, thereby compromising the accuracy of the data recorded by the onboard

instruments. Fortunately, this “internal noise” can be subtracted from the datasets, bringing them to level 1B (L1B). From here, the conversions applied to the data from OLCI and TROPOMI take slightly different paths.

Regarding OLCI, an algorithm called Polymer is used for atmospheric correction, which involves subtracting signals originating from all atmospheric components of the planet. This ensures that we only have signals coming from the Earth’s surface, classified as level 2 (L2) data. At this level, the data represents the concentrations of chlorophyll-a in the planet’s water bodies and needs to be converted into PFTs concentration values using a special algorithm that is called OC-PFT (Ocean Color-Phytoplankton Functional Types). OC-PFT belongs to a category of algorithms based on what is known as an “abundance-based approach”, which utilizes empirical knowledge of the relationships between phytoplankton abundance in a certain area and the percentages of PFTs that generate them to convert level 2 data to level 2A data (this level contains the PFTs concentration values). The empirical relationships underlying an abundance-based algorithm are investigated through the analysis of samples from various sites using high pressure liquid chromatography (HPLC), a technique that allows quantification of the concentrations of characteristic pigments of various PFTs in a specific area. Clearly, abundance-based algorithms are not effective in describing scenarios where the quantities of PFTs deviate from established relationships, such as situations where climate change affects phytoplankton populations. Fortunately, these algorithms are not the only viable option for transitioning from level 2 to level 2A data: ecological-based algorithms and spectral-based algorithms also exist. The former utilize level 1B data and other environmental variables derived from satellite measurements to determine the ecological niches represented by different areas of interest and the associated phytoplankton communities. However, ecological-based algorithms have similar limitations to abundance-based algorithms: “Since there can be deviations (natural or anthropogenically driven) from the tuned ecological relationships, we must be careful in interpreting time-series using these approaches” (Bracher *et al.*, 2022). On the other hand, spectral-based algorithms analyze variations in the optical properties of the marine/oceanic surface, which vary with the concentration of phytoplankton pigments and the size of the cells present in it.

Once we understand the steps by which L2A data can be obtained from OLCI, understanding the steps by which we can do the same with TROPOMI is much easier: in fact, level 1B data from TROPOMI is transformed directly into level 2A data through a spectroscopic method called DOAS (Differential Optical Absorption Spectroscopy).

The L2A data by themselves do not represent the maps that one looks for when interested in understanding the distributions of phytoplankton, since they relate to the orbits that satellites make around the planet. To obtain a set of information that refers, for example, to the entire Atlantic Ocean, it is necessary to merge L2A data from different orbits creating a grid representative of the mentioned area. This operation is called gridding, and it is necessary to perform it in order to obtain datasets that can be used in the graphical representation of maps showing the global distribution of different PFTs.

1.5 Aim of the project

As shown in Section 1.2, the times in which we live makes clear the need to possess quality information on the global distribution of different phytoplankton functional types. Numerous attempts to obtain this information have been conducted using both datasets from various multispectral satellites and datasets collected from different hyperspectral satellites.

In both cases, the results obtained had some strengths but also some defect aspects: in the case of the maps obtained through the use of multispectral instruments, the strengths lay in the great spatial resolution shown by the maps, while the defect aspects consisted of their low spectral resolution. Conversely, the maps obtained through the use of the datasets formed through the activity of hyperspectral instruments represented more accurate PFT concentration values but related to much larger pixels. Thus, these were maps with high spectral resolution but low spatial resolution.

In recent years an approach, called synergistic, has been experimented with, which aims at obtaining maps that possess the best characteristics of both types of instruments. Through data assimilation techniques, datasets from hyperspectral and multispectral instruments are fused in such a way that maps that have both good spectral resolution and good spatial resolution can be produced. An example of this type of approach can be found in the work published in 2017 by Losa and colleagues, in which an algorithm was developed, called SynSenPFT, that allowed them to fuse data from OC-CCI and SCIAMACHY instruments to obtain maps with the above characteristics. An excerpt of the results achieved during the span of this work can be seen in Figure 13.

This thesis work is in the vein of studies such as the one just mentioned and aims to produce a synergistic method that can be applied to datasets from instruments of more recent conception than those considered by Losa and colleagues, namely, the OLCI and TROPOMI instruments of the European Space Agency's Copernicus system. Such a method can be used to produce a set of datasets representative of a time span spanning several months, on which time series analyses can be conducted to understand spatiotemporal variations in diatoms and cyanobacteria distributions. In addition, such datasets can be used to deepen the understanding of important biological information about these two PFTs, such as, for example, their phenology.

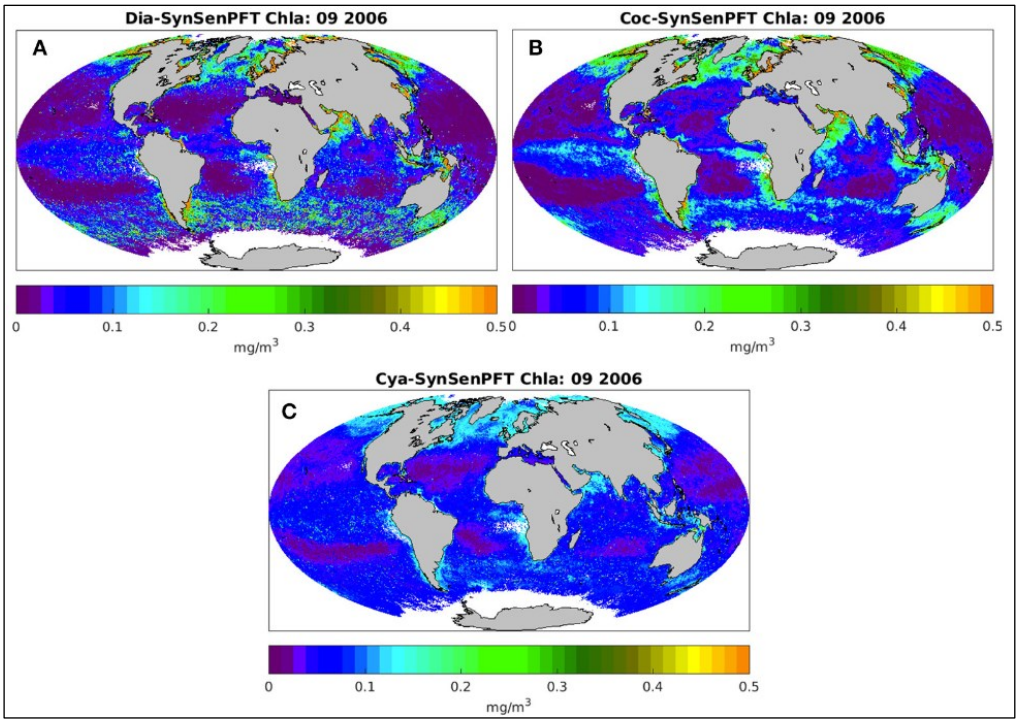

Figure 13 | **Distribution maps of three different PFTs obtained through the SynSenPFT algorithm.** These maps represent the global average PFT distribution of diatoms, coccolithophores, and cyanobacteria during September 2006, and were obtained by applying the SynSenPFT algorithm to data obtained from SCIAMACHY and OC-CCI instruments during the indicated period. The concentration of the different PFTs is expressed in milligrams per cubic meter of water. (Losa *et al.*, 2017)

2. Theoretical background

As seen in the previous chapter, this thesis work focuses on the development of a synergistic method to combine the concentration values of two different phytoplankton functional types obtained using the OLCI and TROPOMI instruments. This chapter aim to answer in detail three questions that are necessary to proceed in that direction:

- Chapter 2.1 – How can plankton concentration values be obtained from TROPOMI?
- Chapter 2.2 – How do we obtain plankton concentration values from OLCI?
- Chapter 2.3 – How do we merge two datasets?

2.1 Differential Optical Absorption Spectroscopy (DOAS)

Imagine that during a field expedition you recovered a seawater sample containing cyanobacteria and, hypothetically, you could extract all other organic absorbing species from the sample. How could the concentration of cyanobacteria in the sample be measured? By taking advantage of the Beer-Lambert's law – or Beer-Lambert-Bouguer's law – (Gold, 2019).

In fact, if we poured the sample into a laboratory cuvette, and irradiated it with a monochromatic beam of electromagnetic waves, that is, with a set of waves characterized by the same wavelength (λ), we would notice a difference between the intensity of the beam before and after interaction with the sample. This is due to the absorption of electromagnetic radiation by the pigments contained in cyanobacteria, and this phenomenon can be described through the equation

$$A(\lambda) = \ln\left(\frac{I^0(\lambda)}{I(\lambda)}\right) = \epsilon\, c\, l \quad ,$$

where A represents the amount of electromagnetic radiation of wavelength λ that the solution absorbs, I^0 the intensity of the incident wave beam, and I the intensity of the wave beam after interaction with the sample. In addition to the ratio between the intensities of the wave beam, the absorbance (A) can also be determined by the multiplication of three parameters:

- ϵ, the molar absorption coefficient, which is a constant of proportionality.
- c, the concentration of the absorbing species within the sample.
- l, the length of the optical path traveled by light, that is, the length of the path within which the light interacts with the absorbing species. In the hypothetical case introduced above, the depth of the cuvette.

It is evident how, thanks to this relationship, it is possible to calculate the concentration of cyanobacteria in the cuvette: simply irradiate it with a monochromatic beam of electromagnetic waves, measure its intensity before and after interaction with the cuvette, and calculate the absorbance of the sample. From there, knowing the optical path length and the molar absorption coefficient of cyanobacteria, the desired value can be derived. But how does Beer-Lambert's law relate to a satellite instrument like TROPOMI?

Starting with the intensity of electromagnetic radiation coming from the sun with a certain λ, $I^0(\lambda)$, and the measured top-of-atmosphere radiance, $I(\lambda, s)$, satellite sensors such as TROPOMI exploit the Beer-Lambert law to calculate the concentration of various inorganic compounds in the atmosphere (Losa *et al.*, 2017). However, they can also be used to measure the concentration of absorbing objects residing on the planet's surface. In fact, the radiation that converges in the top-of-atmosphere radiance values comes from different paths: in some cases, radiation incident on planet Earth has penetrated the atmosphere, come in contact with the planet's surface (for those interested in phytoplankton, it has come in contact with the surface of the seas and oceans), and from there has been reflected or backscattered to the atmosphere. From this radiation, we can derive the concentration of phytoplankton on the surface layer of terrestrial waters.

The DOAS technique involves calculating the optical density (τ, a synonym for absorbance) of a certain optical path as a function of a certain wavelength (λ) and a certain zenith angle (s), which is the angle comprised between the direct line from the center of the Sun to the center of the Earth and the normal to the Earth's surface at a specific point on the planet. In other words, it is the angle between the incident solar rays and the local vertical. Once the value of τ has been determined according to the equation

$$\tau(\lambda, s) = \ln\left(\frac{I^0(\lambda)}{I(\lambda, s)}\right),$$

the components that generate it are identified, so that a new equality can be developed. To this end, Astrid Bracher and her group (Bracher *et al.*, 2009), based on the study by Vountas and colleagues (Vountas *et al.*, 2007), identified a series of components of τ that they summarized in the following equation:

$$\begin{aligned}\tau(\lambda, s) &= \ln\left(\frac{I^0(\lambda)}{I(\lambda, s)}\right) \\ &= \sum_i \sigma_i(\lambda) \times S_i(s) + \sigma_{ring}(\lambda) \times S_{ring}(s) + \sigma_{VRS}(\lambda) \times S_{VRS}(s) \\ &\quad + \sigma_{phyto}(\lambda) \times S_{phyto}(s) - \sum_{k=0}^{n} a_k \lambda^k\end{aligned}$$

To best understand it, let us carefully consider each component of this equality:

- σ_i (λ) represents the cross-section of the generic atmospheric trace gas *i* at a certain λ. The term "cross-section" refers to a virtual area representing the probability of interaction between a photon of light at wavelength λ and the molecule *i*. This value is multiplied by $S_i(s)$ which, roughly, can be defined as the number of elements *i* present along the optical path. The expression

$$\sum_i \sigma_i(\lambda) \times S_i(s)$$

  thus represents the absorption contribution of all trace gases present in the atmosphere.
- $\sigma_{ring}(\lambda)$ indicates the cross-section related to the inelastic interactions that an electromagnetic radiation may undergo when it impinges against N_2 and O_2 molecules in the atmosphere: this phenomenon is called the Ring effect. According to the same logic expressed in the previous point, $\sigma_{ring}(\lambda)$ is multiplied by $S_{ring}(s)$.
- Inelastic interactions like those occurring in the atmosphere can also involve the water molecules with which electromagnetic radiation interacts. In this case we speak of a phenomenon known as Vibrational Raman Scattering, which is considered by the factors $\sigma_{VRS}(\lambda)$ and $S_{VRS}(s)$.

  Both the Ring effect and the VRS are referred to as pseudo-absorbers: the equation of Bracher and colleagues takes into account only these two pseudo-absorbers, while Vountas and colleagues consider more through the inclusion of additional factors in the equation.

- $\sigma_{phyto}(\lambda)$ and $S_{phyto}(s)$ represent the contribution of phytoplankton to absorption processes. To be able to derive $\sigma_{phyto}(\lambda)$, as well as the rest of the cross-sectional values, it is necessary to consider absorption spectra of the absorbing species of interest. Figure 14 shows an example of such spectra:

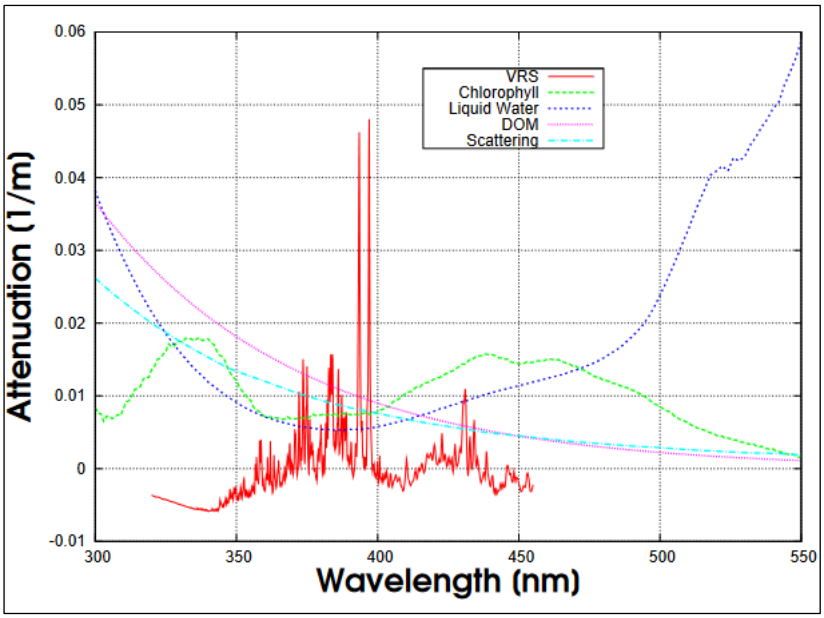

Figure 14 | **Example of spectra used for determining cross-sectional values.** (Vountas *et al.*, 2007)

The DOAS method makes it possible to derive both the concentration values of chlorophyll-a, and from these to understand what the concentration levels of phytoplankton on the water surface are, and to directly derive the concentration values of the different phytoplankton groups (the PFTs): instead of using the cross-section value of chlorophyll-a as σ_{phyto}, it is possible to assign to this parameter the cross-section value of a specific pigment of the phytoplankton group in which one is interested. Pigments typical of phytoplankton groups have different absorption spectra, which means that they absorb the same λ differently, however, often the differences are small: for this reason, it is important to have hyperspectral instruments such as TROPOMI: given the large number of wavelengths they can detect, it is possible to apply the DOAS method to a wavelength at which a certain pigment responds well. On the other hand, multispectral satellites such as OLCI do not give this possibility, since they cannot detect a large number of wavelengths: they usually detect wavelengths around 443 nm, and therefore they allow to measure the concentration of chlorophyll-a on the water surface, but then it is necessary to process this kind of data to get an idea of the concentrations of the different phytoplankton groups.

- a_k is instead the coefficient of a low-degree polynomial – typically of a degree of three (Vountas *et al.*, 2007) – which is included in the equation to separate spectrally dynamic features from the slowly varying attenuation.

The "*S*" parameters are called fit factors, and they are at the heart of the DOAS method: once the cross-section and optical density values have been derived, an initial estimate of the value of the fit factors is advanced, the optical density value that corresponds to the estimate is calculated, and the difference with that obtained through measurements is derived (and it is squared). From this point, we proceed to iteratively reduce that difference by least-square minimization.

Having finished the process, one can derive the phytoplankton concentration on the water surface from the S_{phyto} fit factor. In fact, the concentration of phytoplankton, c, expressed as mg of chlorophyll-a per cubic meter, is obtained by the formula

$$c = \frac{S_{phyto}}{\delta},$$

where δ represents the penetration depth into water of electromagnetic waves of wavelength λ.

In conclusion, DOAS is thus a spectroscopic technique that allows us to derive, from TROPOMI retrievals of radiances at different top-of-atmosphere points and a formula, the concentration of phytoplankton at corresponding points on the planet's surface. As will be seen in the next chapter, the method that allows us to derive phytoplankton concentrations from measurements made by OLCI stands on a quite different logic.

2.2 POLYnomial-based algorithm applied to MERIS (Polymer) and Ocean Color – Phytoplankton Functional Types (OC-PFT)

To derive data from OLCI without the help of a spectroscopic technique such as DOAS, it is necessary to rely on two algorithms, Polymer and OC-PFT. The former, deals with what is commonly referred to as atmospheric correction, i.e., it removes the absorption contributions of anything that is not phytoplankton, while OC-PFT allows one to distinguish the contribution of the PFTs of interest within the set of absorbers that Polymer identifies as phytoplankton. Let's delve deeper into these algorithms.

Polymer's process of operation can be broken down into three steps (Soppa *et al.*, 2021): pre-correction of the top-of-atmosphere radiance (L_{TOA}), fitting process, and retrieval of the chlorophyll-a concentration values.

1. **Pre-correction of the top-of-atmosphere radiance**: after the conversion of L_{TOA} measured by the sensor into reflectance (ρ_{TOA}), ρ_{TOA} is defined as a function of several variables:
    a. Gaseous transmittance estimated for O_3 and for NO_2.
    b. Reflectance due to scattering by air molecules (ρ_{mol}).
    c. Sunglint reflectance (ρ_{gli} with transmission factor T). The phenomenon of sunglint occurs when sunlight is reflected off the surface of water at the same angle that a sensor views it.
    d. Aerosol reflectance (ρ_{aer}).
    e. The coupling between Sun glint, molecules, and aerosols (ρ_{coup}).
    f. The water-leaving reflectance just above the surface (ρ_w^+) with direct and diffuse atmospheric transmission (t).

    The equation that sum up all these variables is the following:

$$\rho_{TOA}(\lambda) = t_{oz}(\lambda)t_{NO2}(\lambda)[\rho_{mol}(\lambda) + T\rho_{gli} + \rho_{aer}(\lambda) + \rho_{coup}(\lambda) + t(\lambda)\rho_w^+(\lambda)]$$

    ρ_{TOA} undergoes initial corrections for gaseous transmittance, absorption by air molecules, Rayleigh scattering, and sunglint, resulting in a pre-corrected reflectance (ρ'), which still contains a residual sun glint ($\Delta\rho_{gli}$) and it's defined as it follows:

$$\rho'(\lambda) = \Delta\rho_{gli}(\lambda) + \rho_{aer}(\lambda) + \rho_{coup}(\lambda) + t(\lambda)\rho_w^+(\lambda)$$

Then, the variables composing ρ' can be grouped into atmospheric and sun glint contribution (ρ_{ag}) and water components contribution ($t\rho_w^+$):

$$\rho'(\lambda) = \rho_{ag}(\lambda) + t(\lambda)\rho_w^+(\lambda)$$

2. **Fitting process**: ρ_{ag} and $t\rho_w^+$ are optimized in such a way as to obtain the best fit of ρ'.
3. **Chlorophyll-a values retrieval**: the water-leaving reflectance is modelled as a function of two variables (taking for granted λ): chlorophyll-a concentration (*chl*) and a coefficient, f_b, that scales the backscattering coefficient of particles in the water body:

$$\rho_w^+(\lambda) = f(chl, f_b)$$

The fitting process provided the value of ρ_w^+, and this allows, through the way it is modeled, to obtain the chlorophyll-a concentration values (proxy of phytoplankton concentration) that will be used as the input of OC-PFT.

To understand what OC-PFT is and how it works, it is necessary to delve into the study that sanctioned its birth. OC-PFT was developed by Takafumi Hirata and colleagues in 2011 (Hirata *et al.*, 2011) to build a bridge between the information obtained from studies using high pressure liquid chromatography (HPLC) and that derived from space-borne ocean color sensors.

Indeed, HPLC analyses allow accurate characterization of the composition of planktonic communities starting from the total concentration of chlorophyll-a (TChla) contained in samples obtained from *in situ* measurements, but unfortunately, they have a very low spatiotemporal resolution: the one of the researchers going on the field. On the other hand, space-borne ocean color sensors can provide TChla concentration values at a much higher spatiotemporal resolution (unlike *in situ* observations, satellites can cover the entire planet in few days), but at the time Hirata and colleagues developed OC-PFT, the existing algorithms for processing such data did not provide accurate information about the composition of the phytoplanktonic communities: they could only determine the dominant PFT, or distinguish a very limited number of PFTs, from each TChla value. Thus, the goal that Hirata and colleagues desired to achieve through their algorithm was to be able to combine the information from HPLC analyses with the data from satellites, to obtain maps that could help researchers gain insight into the composition of phytoplankton communities on a global scale (Hirata *et al.*, 2011).

To do so, they collected a series of *in situ* measurements taken over the course of several expeditions (a picture of which is shown in Figure 15), eliminated outliers, and divided the remaining 3966 observations in the following way: 70 percent were used for the model development and 30 percent for validation operations.

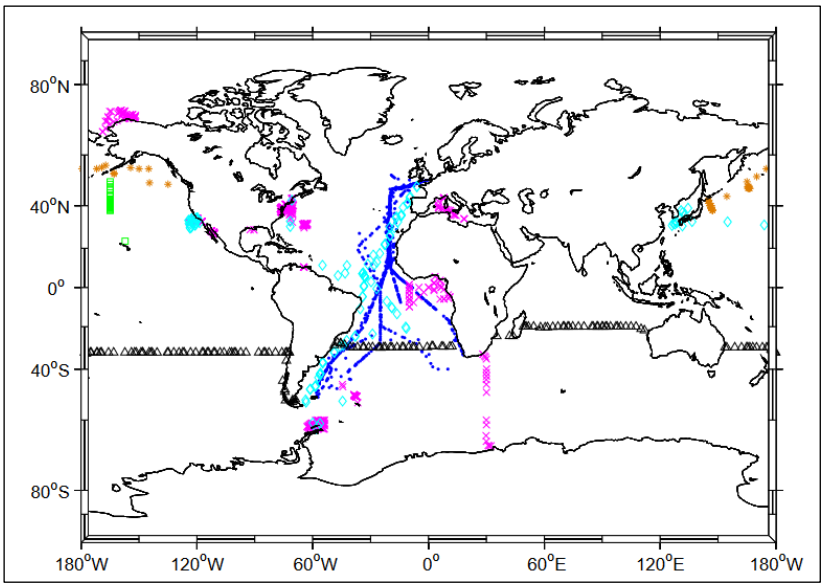

Figure 15 | **Geographical locations of the places from which *in situ* observations used by Hirata and colleagues were taken.** (Hirata *et al.*, 2011) The blue dots have been explored by the NERC AMT cruise (Aiken *et al.*, 2009), while the black triangles by the JAMSTEC BEAGLE cruise (Barlow *et al.*, 2007). The cyan diamonds are relative to the NASA NOMAD dataset (Werdell and Bailey, 2005), while the magenta crosses to the NASA SeaBASS dataset. The brown stars are both the points explored by the SEEDS II cruise (Suzuki *et al.*, 2005) and the sampling stations located along the northwest Pacific Ocean monitoring "A-line" (Isada *et al.*, 2009). Lastly, the green squares are the points explored by the HU Oshoro-maru cruise.

To the data chosen for model development, Hirata and colleagues applied a procedure called Diagnostic Pigment Analysis (DPA), developed by Vidussi and her team (Vidussi *et al.*, 2001) – and subsequently refined (Uitz *et al.*, 2006)– for the purpose of distinguishing phytoplankton functional types from other elements detected during an HPLC procedure.

DPA involves identifying one or more characteristic pigments for each PFT that one wishes to recognize, so that by subjecting a sample derived from an *in situ* observation to HPLC it is possible to quantify the relative abundance of each PFT (f-PFT) based on the ratio between the concentration of the PFT's diagnostic pigments ([PFT-DPs])[1] and the sum of the concentrations of all the other diagnostic pigments:

[1] [X] is a commonly used notation to indicate the concentration of compound X.

$$f\text{-}PFT = \frac{[PFT\text{-}DPs]}{\sum[DPs]}$$

The list of diagnostic pigments that Hirata and colleagues have associated with the different PFTs of their interest is shown in Figure 16, on which two points need to be made:

1. The developers of OC-PFT were not only interested in phytoplankton functional types, but also in phytoplankton size classes, PSCs, which are used to classify phytoplankton based on their size. PSCs are not in the interest of this thesis, as well as 5 of the 7 PFTs that Hirata and his team also focused on: diatoms, dinoflagellates, green algae, prymnesiophytes (haptophytes), pico-eukaryotes, prokaryotes, and Prochlorococcus sp. Nevertheless, to better understand Hirata's study, it is appropriate to report all the results that came out of it.
2. As can be seen from Figure 16, some diagnostic pigments are shared among functional groups, and this can cause several problems in estimating the relative abundances of PFTs. For example, fucoxanthin (Fuco) was chosen as the diagnostic pigment for the PFT "diatoms", however, Fuco is also a precursor of 19'-Hexanoyloxyfucoxanthin (Hex), the diagnostic pigment for haptophytes, and may be present in this group. Hirata writes that during the development of OC-PFT it was necessary to attend to this issue to avoid overestimating the relative abundance of diatoms (Hirata *et al.*, 2011).

PSCs/PFTs	Diagnostic Pigments	Estimation Formula
Microplankton (>20 μm)*2	Fucoxanthin (Fuco), Peridinin (Perid)	1.41 (Fuco + Perid)/ΣDP*2
Diatoms	Fuco	1.41 Fuco/ΣDP*2
Dinoflagellates	Perid	1.41 Perid/ΣDP*2
Nanoplankton (2–20 μm)*1	19'-Hexanoyloxyfucoxanthin (Hex) Chlorophyll-*b* (Chl-*b*) Butanoyloxyfucoxanthin (But) Alloxanthin (Allo)	(X_{n*}1.27 Hex + 1.01 Chl-*b* + 0.35 But + 0.60 Allo)/ΣDP*3
Green algae	Chl-*b*	1.01 Chl-*b*/ΣDP*2
Prymnesiophytes*4 (Haptophytes)	Hex, But	
Picoplankton (0.2–2 μm)*1	Zeaxanthin (Zea), Hex, Chl-*b*	(0.86 Zea + Y_p1.27 Hex)/ΣDP*3
Prokaryotes	Zea	0.86 Zea/ΣDP*2
Pico-eukaryotes*5	Hex, Chl-*b*	
Prochlorococcus sp.	Divinyl Chlorophyll-*a* (DVChl-*a*)	0.74 DVChl-*a*/Chl-*a*

*1 Sieburth et al. (1978)
*2 ΣDP = 1.41 Fuco + 1.41 Perid + 1.27 Hex + 0.6 Allo + 0.35 But + 1.01 Chl-*b* + 0.86 Zea = Chl-*a* (Uitz et al., 2006)
*3 X_n indicates a proportion of nanoplankton contribution in Hex. Similarly Y_p indicates a proportion of picoplankton in Hex, (Brewin et al., 2010)
*4 Given that contributions of Allo to nanoplankton were only a few percent in our data set, haptophytes were approximated to Nano minus Green Algae (see also Fig. 2 caption)
*5 Pico-eukaryotes can be determined from picoplankton minus prokaryotes (see also Fig. 2 caption).

Figure 16 | **List of diagnostic pigments that Hirata and colleagues have associated with different PFTs.** (Hirata *et al.*, 2011)

By applying DPA to data from *in situ* observations, Hirata and colleagues were able to produce graphs, shown in Figure 17, in which they correlate, to different values of TChla, the relative abundance of different PFTs. These data were then fit by the method of least squares in such a way as to produce continuous curves that were used during the development of the algorithm: in other words, the models over which OC-PFT was developed.

The accuracy of these relationships was evaluated by calculating the error generated by the difference between a value produced from the fitting curves and the corresponding value obtained from the *in situ* observations. All these errors were then summarized in an average value, the root mean square error (RMSE), which is obtained through the following formula,

$$RMSE = \sqrt{\frac{\sum(y_{obs} - y_{fit})^2}{N}},$$

where y_{obs} is a generic value associated to an observation, y_{fit} is the corresponding (meaning relative to the same x) value derived from the fitting curve, and N is the number of observations. This evaluation is summarized in Figure 18.

At this point the empirical model, i.e., the curves in figure 17 below, was validated by comparing it with *in situ* observations that were not used for its development (Figure 19), the 30% of the *in situ* dataset that was left behind. These samples have already been subjected to HPLC to determine what is the total concentration of chlorophyll-a within them, and what are the percentages of PFTs contributing to that value: from the value of TChla they contain, the models created earlier are used to calculate the percentages of PFTs contributing to that value. The closer the results of this operation are to the values obtained through HPCL, the more accurate the models are.

After model development and validation operations are finished, an algorithm (OC-PFT, in fact) was created to apply the model to the mean chlorophyll-a concentrations detected by the SeaWiFS satellite sensor during the period 1998-2009 (O'Reilly *et al.*, 1998). The results obtained by Hirata and colleagues are shown in Figure 20 and the errors associated with them in Figure 21.

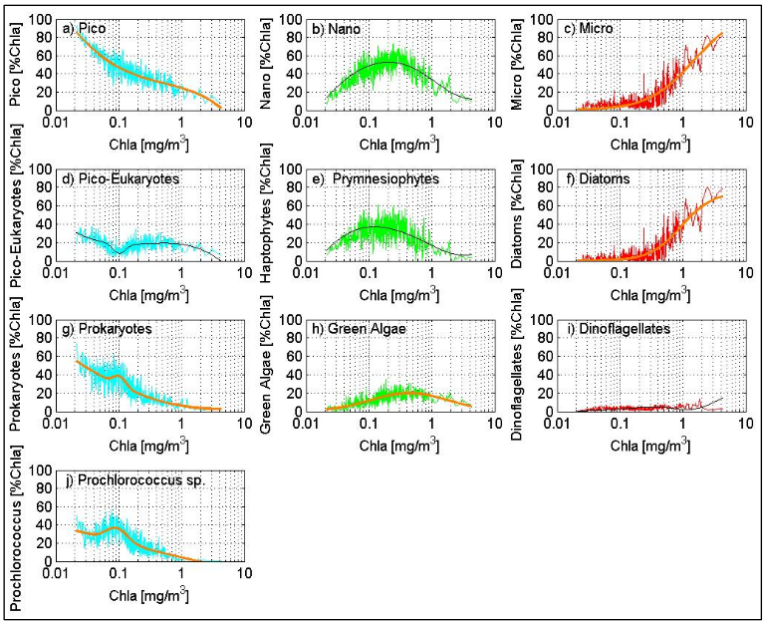

Figure 17 | **Relationships between total chlorophyll concentration and relative PFT concentration.** The first row of graphs relates to phytoplankton size classes, and determines the color of the graphs below: since both diatoms and dinoflagellates, for example, are microplankton, and since the curve relating to the "microplankton" PSC is colored red, the graphs of the "diatom" and "dinoflagellates" PFTs are colored red. The thick yellow fitting lines are those derived by the least squares method, mentre le linee di fitting nere sono ricavate per differenza. For example, if we consider the fitting lines in the first row of graphs: if picoplankton represents a percentage X of a certain value of chlorophyll-a concentration, and microplankton represents a percentage Y, then nanoplankton will be the Z = 100% − (X + Y) of the chlorophyll-a concentration. Applying this reasoning to the whole fitting curves, one can understand how the black curves were obtained. (Hirata *et al.*, 2011)

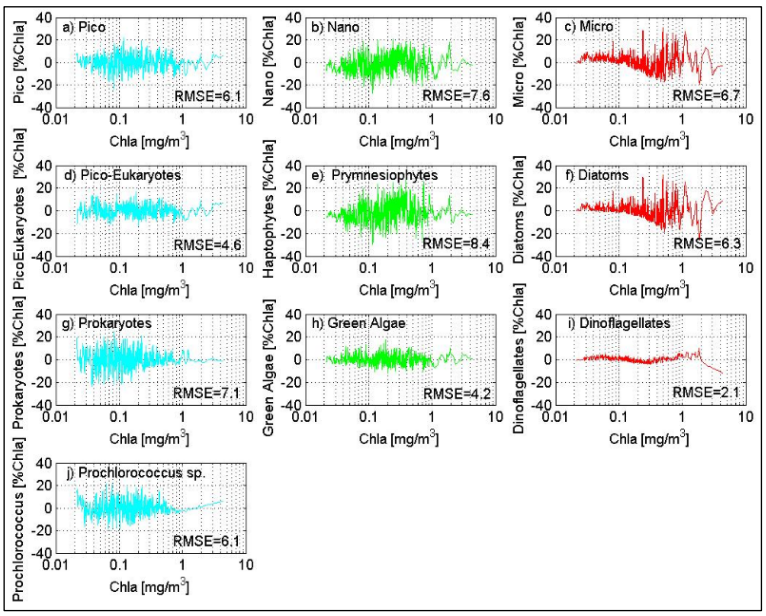

Figure 18 | **Uncertainties of the relationships between total chlorophyll concentration and relative PFT concentration.** The x-axis of each graph shows different values of chlorophyll-a concentration, reported in milligrams per cubic meter, while the y-axis shows the relative errors as percentages of the value estimated through the fitting curve. The root mean square error is reported in each plot, and it's calculated as described above. (Hirata *et al.*, 2011)

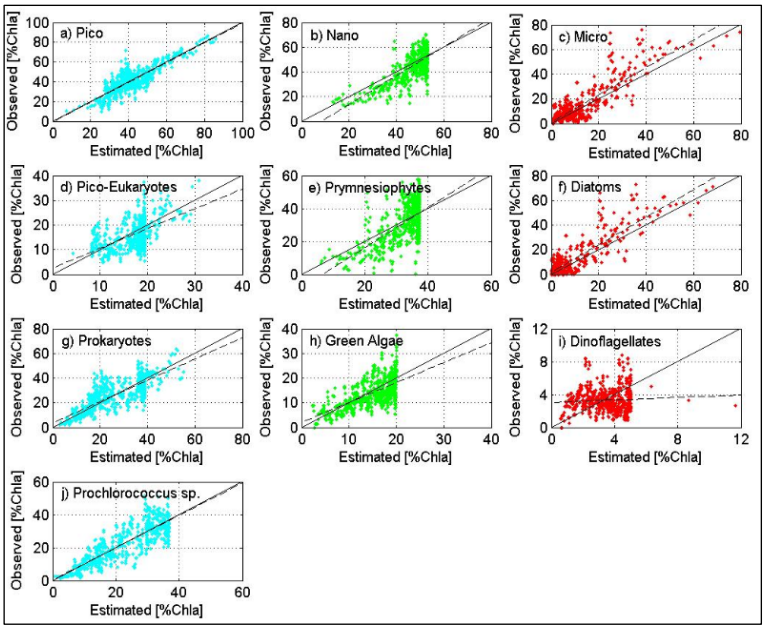

Figure 19 | **OC-PFT validation results.** Given the total concentration of chlorophyll-a in a series of samples, the percentages of phytoplankton comprising it are measured by HPLC and estimated using the models depicted in Figure 17. The results of both operations are plotted in the scatter plots depicted above and interpolated with a linear curve: the more the line overlaps the bisector of the Cartesian plane, the more the measurements and estimates resemble each other. *Ergo*, the more accurate the models are. (Hirata *et al.*, 2011)

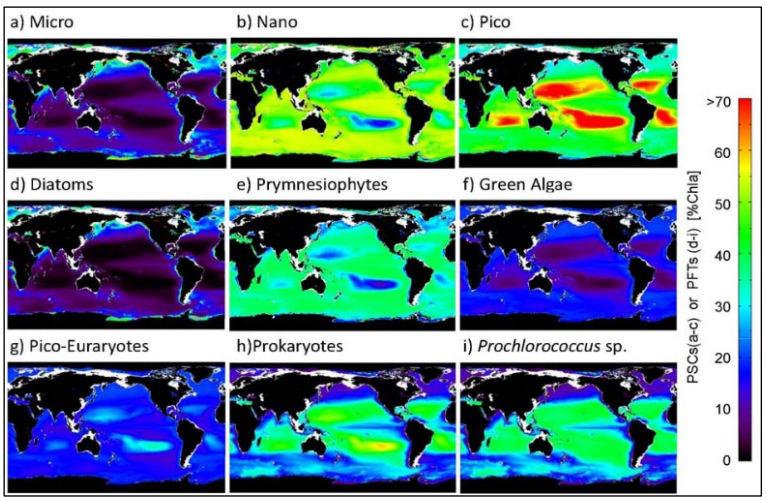

Figure 20 | **Distribution of surface PFTs derived from SeaWiFS data.** (Hirata *et al.*, 2011)

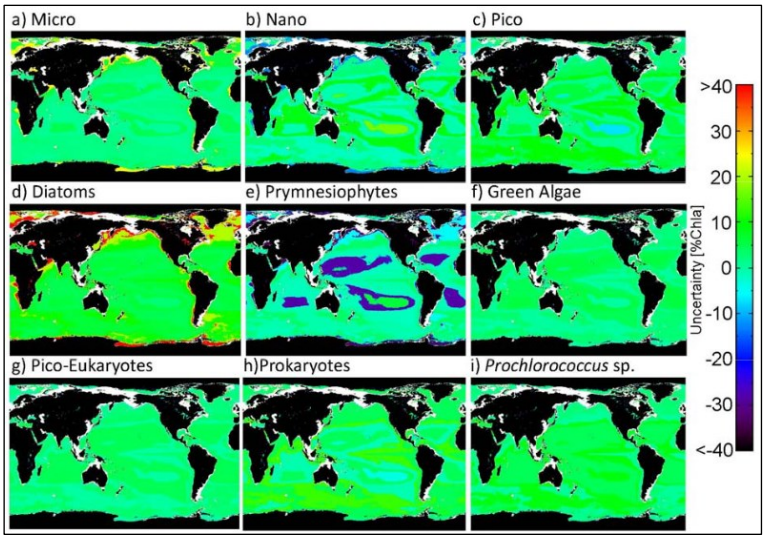

Figure 21 | **Uncertainties of the distributions of surface PFTs [%Chl-a] derived from SeaWiFS data.** (Hirata *et al.*, 2011)

After describing how OC-PFT was developed and showing what early results it produced, to conclude this chapter, it is crucial to emphasize the close link that an algorithm such as OC-PFT has with observations conducted in the field. As shown, models that link total chlorophyll-a concentrations to the relative abundance of PFTs are built based on empirical data, so they have several limitations:

- They cannot predict atypical associations: relationships derived from empirical data and taken for granted globally may have regional variations. Mariana Soppa and colleagues (Soppa *et al.*, 2014) have highlighted this critical issue excellently, showing that the OC-PFT algorithm, as developed by Hirata and colleagues, underestimates the diatom abundance in the Southern Ocean. Through this study, it was possible to advance the hypothesis that diatoms in the Southern Ocean might be more abundant than previously thought, and to tie that estimate to the lack of *in situ* phytoplankton pigment data, and to the fact that, probably, the relationship between the total chlorophyll-a concentration measured in one sampling point in the Southern Ocean and the relative concentration of diatoms in the same point is different from the rest of the world.
- They may vary with environmental conditions and thus the model parameters may change over time.
- They can lead to incorrect predictions of the structure of different phytoplanktonic communities if variations in the same occur without any change in TChla.

For these reasons, it is critically important to update the datasets with which the models underlying algorithms such as OC-PFT are developed as soon as there is availability of new data.

2.3 Data assimilation

Investigating a component of the Earth system provides two main sources of information: observations and models (Lahoz and Schneider, 2014). Observations are defined as measurements that are made in direct contact with the object of investigation, while models are defined as all instruments that describe relationships between variables, such as, for example, equations. Because of their close contact with the object of investigation, observations exhibit more pronounced spatiotemporal discontinuities than those exhibited by models, which, on the other hand, are less reliable because of their looser connection with the object of investigation. Fortunately, the information from observations and models can be interpolated through various data assimilation procedures that yield results that are spatiotemporally homogeneous and, at the same time, more reliable than models alone. Figure 22 graphically represents these concepts.

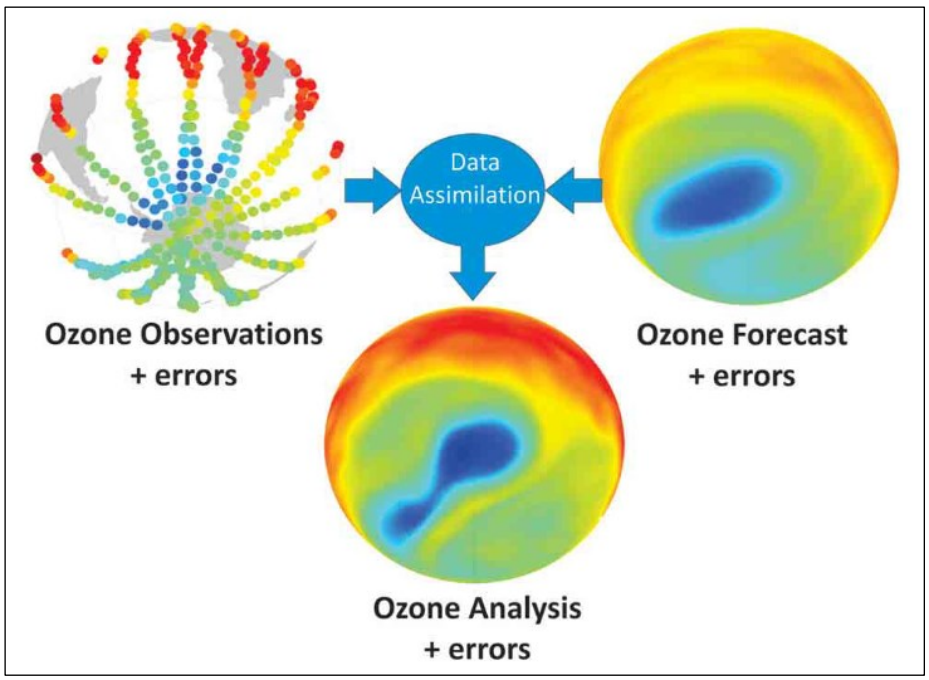

Figure 22 | **A schematic representation of a data assimilation process.** Observations are more dispersed in space and time than model data, but they derive from a more accurate interaction with the object: if assimilated with model data, they provide accurate information with less spatiotemporal discontinuity. In this picture the model data are reported as "Forecast" while the result of the assimilation process is called "Analysis". (Lahoz and Schneider, 2014)

Should it be desired to interpolate data from TROPOMI and OLCI, the characteristics of the two instruments would allow the data to be considered as observations in the former case, and models in the latter. For this reason, it was thought to combine the information from OLCI and

TROPOMI using a data assimilation procedure known as Optimal Interpolation (OI), the same as that used in the article published by Svetlana Losa and colleagues in 2017 (Losa *et al.*, 2017). This procedure consists of an equation for integrating information obtained from different measurements (y) to previously obtained data — that is, model data, in the data assimilation language — called background data ($\boldsymbol{x^b}$, where the superscript b is not an exponent, but stands to indicate that $\boldsymbol{x^b}$ refers to background data), and comes in this form:

$$\mathbf{x}^{\mathbf{a}} = \mathbf{x}^{\mathbf{b}} + \mathbf{K}(y - H(\mathbf{x}^{\mathbf{b}}))$$

The result of the first equation, $\boldsymbol{x^a}$ (the a at the exponent sums up the word "analysis"), is what is sought when operating an OI process: the "integrated" dataset that contains both information from y and from $\boldsymbol{x^b}$. The other components of the equation, $\mathbf{K}$ and H, are referred to as "Kalman filter" and "mapping function" and are critical to the success of the interpolation process (Lacey, n.d.).

While the mapping function is concerned with transforming the background data into the space of measurements (i.e., in other words, the mapping function makes the background data compatible with the measurements so that the two types of data can interact), the Kalman filter determines how much individual data should influence the formation of $\boldsymbol{x^a}$. Before showing how it was decided to define y, $\boldsymbol{x^b}$, $\mathbf{K}$ and H, it is necessary to explain how the general logic of the data assimilation process was imagined.

Imagine a geographical area bounded by four points, each of the four defined by its own latitude and longitude coordinates (lat, lon). Now imagine dividing the interior of this area into pixels whose size is dependent on the resolution of the instruments referred to: in the case where we wanted to place data from TROPOMI in this area, the pixels would be larger than in the case where we wanted to place data from OLCI. In other words, if we drew on the same geographic area first the pixels identified by the resolution of TROPOMI, and then those identified by the resolution of OLCI, we would realize that one pixel of TROPOMI "contains" several pixels of OLCI. For clarity, refer to Figure 23 below: the left matrix represents a generic grid of pixels containing data from TROPOMI, while the right matrix represents a generic grid of pixels containing data from OLCI.

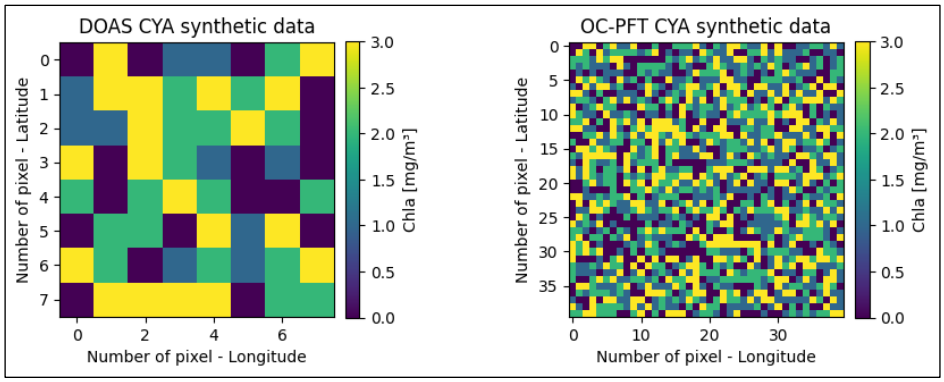

Figure 23 | **Example of the relationship between a generic TROPOMI grid and a generic OLCI grid.** Data mimicking the possible concentration values assumable by chlorophyll-a relative to the PFT cyanobacteria were generated and arranged in grids formed by pixels mimicking a possible size ratio between TROPOMI pixels and OLCI pixels. In this case, the ratio considered is 1:5.

The precise number (N) of OLCI pixels contained in a TROPOMI pixel can be determined by the formulas given below, where ξ represents the spatial resolution of the instruments: in the first case, that in which the pixels are square, the ξ of the instruments is the same on both the latitude and longitude dimensions, while in the case in which the pixels are rectangular, ξ varies depending on the dimension considered.

$$N = \left(\frac{\xi_{OLCI}}{\xi_{TROPOMI}}\right)^2$$

$$N = \left(\frac{\xi_{OLCIlat}}{\xi_{TROPOMIlat}}\right)\left(\frac{\xi_{OLCIlon}}{\xi_{TROPOMIlon}}\right)$$

So, in sight of the spatial resolution and spectral resolution characteristics, it was planned to proceed with data assimilation by considering data from TROPOMI as measures (i.e., y), and data from OLCI as background data (i.e., $\boldsymbol{x^b}$). Since the project's area of interest was decided to be the entire Atlantic Ocean, stowing all the data for such a large area in two variables turned out to be an infeasible route: the runtime of the algorithms would be incredibly long, and so it proved necessary to devise a strategy to deal with this problem.

We therefore thought, once the data from TROPOMI and OLCI were arranged on two grids, to operate a cycle in the TROPOMI grid, and consider a single pixel at a time. The choice to loop over the TROPOMI grid was almost forced: since this grid contains fewer pixels than the other, the cycles that affect it are shorter than those that are performed on the OLCI grid. So only one TROPOMI pixel is considered at a time, and the value in it, a scalar, is assigned to the variable y. The variable $\boldsymbol{x^b}$, on the other hand, is assigned to the values of the OLCI pixels

located within the considered TROPOMI pixel. Ergo, while y is a scalar, $\boldsymbol{x^b}$ is a vector containing N elements (also scalars).

Now that we have defined y and $\boldsymbol{x^b}$, it is necessary to understand how the other components of the equation

$$\mathbf{x^a} = \mathbf{x^b} + \mathbf{K}(y - H(\mathbf{x^b}))$$

can be obtained, so that it is possible to obtain $\boldsymbol{x^a}$. Let's start with H.

The purpose of H is to transform $\boldsymbol{x^b}$ in the space of y so that the two variables can interact; in this case, since y must be subtracted from $H(\boldsymbol{x^b})$, H must transform $\boldsymbol{x^b}$ into a scalar. How to do this? Following several days of reflection, a fruitful dialogue with Dr. Svetlana Losa highlighted the answer to this question. By defining H as the average of the values contained in the vector $\boldsymbol{x^b}$, one can easily move from vector space to scalar space.

$$H(\mathbf{x^b}) = \overline{\mathbf{x^b}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x_i^b}$$
$$H : R^{N\times 1} \rightarrow R^1$$

It now remains to understand how the so-called Kalman filter**,** **K**, defined by the following equation, can be obtained:

$$\mathbf{K} = \mathbf{P^b H^T}\left(\mathbf{H P^b H^T} + \mathbf{R}\right)^{-1}.$$

The three main components of this equality are:

- $\mathbf{P^b}$, that is, the matrix of covariances relative to the background data.
- **H** which is the matrix of the prime partial derivatives of $H(\boldsymbol{x^b})$ with respect to the variables that make up $\boldsymbol{x^b}$, calculated in $\boldsymbol{x^b}$.
- **R**, the matrix of covariances relative to the measured data, i.e., to y.

$\mathbf{H^T}$ was not mentioned in the previous bullet point list because it is simply the transposed matrix of **H**, that is, a matrix obtained by exchanging the rows and columns of H. And since H is a matrix with one row and N columns, $\mathbf{H^T}$ will be a matrix with N rows and one column. But specifically, what kind of matrix is **H**? What does it mean?

H indicates how much each of the variables that make up $\boldsymbol{x^b}$ affect $H(\boldsymbol{x^b})$, in other words, it allows us to understand how much "weight" a single OLCI pixel has in forming the scalar that is made to interact with y. For this reason, **H** is defined according to the following equation:

$$\mathbf{H} = \left.\frac{dH}{d\mathbf{x}}\right|_{\mathbf{x^b}} = \left[\frac{dH}{dx_1^b}(\mathbf{x^b}) \quad \frac{dH}{dx_2^b}(\mathbf{x^b}) \quad \dots \quad \frac{dH}{dx_N^b}(\mathbf{x^b})\right]$$

Since $H(\boldsymbol{x^b})$ is defined as the arithmetic mean of the values contained in $\boldsymbol{x^b}$, we can expect the contribution of the individual variables in the formation of $H(\boldsymbol{x^b})$ to be identical. In other words, if:

$$H(\mathbf{x^b}) = \overline{\mathbf{x^b}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x_i^b},$$

then:

$$\mathbf{H} = \left[\frac{1}{N} \quad \frac{1}{N} \quad \dots \quad \frac{1}{N}\right].$$

Having defined H, it remains to figure out how to get $\mathbf{P^b}$ and **R**, and to do so it is necessary to define what a covariance matrix is, starting from what the covariance is. The covariance of two variables, say X and Y, that are distributed over N value with mean, respectively, $\bar{x}$ and $\bar{y}$, is calculated according to the following formula:

$$\sigma_{X,Y} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{N}\sum_{i=1}^{N}x_i y_i - \left(\frac{1}{N}\sum_{i=1}^{N}x_i\right)\left(\frac{1}{N}\sum_{i=1}^{N}y_i\right).$$

Now, since we must imagine that, to form $\mathbf{P^b}$, we are calculating the covariance of two OLCI pixels, that is, two variables of $\boldsymbol{x^b}$, the formula is simplified, since the variables do not consist of populations of observations, but they consist of only one value. Thus:

$$\sigma_{\mathbf{x_i^b},\mathbf{x_j^b}} = \mathrm{Cov}(\mathbf{x_i^b}, \mathbf{x_j^b}) = (\mathbf{x_i^b} - \bar{\mathbf{x_i^b}})(\mathbf{x_j^b} - \bar{\mathbf{x_j^b}}).$$

From this, we can construct the matrix of covariances, which, in the case of a vector containing N values such as $\boldsymbol{x^b}$, will be an N × N matrix in which the row and column at which a value is placed indicate the two variables whose covariance is represented: at position $\mathbf{P^b}_{(2,3)}$, for example, Cov($\boldsymbol{x^b}_2$, $\boldsymbol{x^b}_3$) will be found.

$$\mathbf{P^b} = \begin{bmatrix} P^b_{1,1} & \dots & P^b_{1,N} \\ \vdots & \ddots & \vdots \\ P^b_{N,1} & \dots & P^b_{N,N} \end{bmatrix} = \begin{bmatrix} \mathrm{Cov}(\mathbf{x_1^b}, \mathbf{x_1^b}) & \dots & \mathrm{Cov}(\mathbf{x_1^b}, \mathbf{x_N^b}) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}(\mathbf{x_N^b}, \mathbf{x_1^b}) & \dots & \mathrm{Cov}(\mathbf{x_N^b}, \mathbf{x_N^b}) \end{bmatrix}$$

Since the variance of a variable is defined according to the equation

$$\sigma_X^2 = \mathrm{Var}(X) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \,,$$

and since the variables of $\boldsymbol{x^b}$ are assigned to single values, then:

$$\sigma^2_{\mathbf{x_i^b}} = \mathrm{Var}(\mathbf{x_i^b}) = (\mathbf{x_i^b} - \bar{\mathbf{x}^b})^2 \,.$$

This equation makes it very clear that the variance of a variable is equal to the covariance of the same variable with respect to itself,

$$\mathrm{Var}(X) = \mathrm{Cov}(X, X) \,,$$

from which it follows that:

$$\mathbf{P^b} = \begin{bmatrix} \mathrm{Var}(\mathbf{x_1^b}) & \dots & \mathrm{Cov}(\mathbf{x_1^b}, \mathbf{x_N^b}) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}(\mathbf{x_N^b}, \mathbf{x_1^b}) & \dots & \mathrm{Var}(\mathbf{x_N^b}) \end{bmatrix} .$$

Once the structure of covariance matrices is understood, it is necessary to find a convenient way to obtain them. A possible solution can involve noticing that the difference between the value of a variable and the expected value of that same variable is the error associated with that variable. It is therefore possible to obtain the matrix of covariances of a variable by multiplying, through outer product, the vector of errors associated with that variable by its transpose. Thus, if one could obtain the vector of errors associated with the values contained in $\boldsymbol{x^b}$, which we call $\boldsymbol{e^b}$, it is possible to obtain $\mathbf{P^b}$ by the multiplication of $\boldsymbol{e^b}$ with $\boldsymbol{e^{bT}}$.

To obtain $\boldsymbol{e^b}$, we define the absolute error associated with the values of $\boldsymbol{x^b}$ as a percentage of the values themselves – for the moment, let's imagine it between 0% and 50% – which we stow in the **err** vector. At this point, to obtain $\boldsymbol{e^b}$ we simply multiply element by element the $\boldsymbol{x^b}$ vector with the **err** vector, and then multiply via outer product $\boldsymbol{e^b}$ with $\boldsymbol{e^{bT}}$ to obtain $\mathbf{P^b}$.

$$\mathbf{x^b} = \begin{bmatrix} x_1^b \\ \vdots \\ x_N^b \end{bmatrix}, \ \mathbf{err} = \begin{bmatrix} n_1 \in [0, 0.5] \\ \vdots \\ n_N \in [0, 0.5] \end{bmatrix}$$

$$\mathbf{e^b} = \mathbf{x^b} \odot \mathbf{err}$$

$$\mathbf{P^b} = \mathbf{e^b} \otimes \mathbf{e^{b^T}} = \begin{bmatrix} P_{1,1}^b & \cdots & P_{1,N}^b \\ \vdots & \ddots & \vdots \\ P_{N,1}^b & \cdots & P_{N,N}^b \end{bmatrix}$$

It is very important to emphasize the difference between the type of product that is applied to the vectors $\boldsymbol{x^b}$ and **err** and that which is applied to the vectors $\boldsymbol{e^b}$ and $\boldsymbol{e^{bT}}$. In the former case it is an element-wise product, also called Hadamard's product (Styan, 1973), which is defined according to the equation:

$$(A \odot B)_{ij} = (A)_{ij}(B)_{ij}.$$

In the second case, however, it is an outer product (Golub and Van Loan, 2013) that, given two vectors

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix},$$

is defined as:

$$(\mathbf{u} \otimes \mathbf{v})_{ij} = u_i v_j.$$

Written this definition in a more expanse way we obtain:

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{A} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_n \\ u_2 v_1 & u_2 v_2 & \cdots & u_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_m v_1 & u_m v_2 & \cdots & u_m v_n \end{bmatrix}.$$

After understanding how to define a matrix of covariances, and applying the definition to $\mathbf{P^b}$, obtaining **R** is quite easy. In fact, while $\mathbf{P^b}$ is the matrix of covariances relative to the values of $\boldsymbol{x^b}$, **R** is the matrix of covariances relative to the values of y, but since y is a scalar, **R** will be

a scalar as well (and for this reason we will denote it as R instead of **R**), and can be easily obtained as:

$$R = \mathbf{e^y} \otimes \mathbf{e^{y^T}} = \mathbf{e^{y^2}}$$

.

At this point we possess all the components necessary to obtain **K**:

$$\mathbf{P^b} = \begin{bmatrix} P^b_{1,1} & \dots & P^b_{1,N} \\ \vdots & \ddots & \vdots \\ P^b_{N,1} & \dots & P^b_{N,N} \end{bmatrix}, \ \mathbf{H} = \left.\frac{dH}{d\mathbf{x}}\right|_{\mathbf{x^b}} = \begin{bmatrix} \frac{dH}{dx^b_1}(\mathbf{x^b}) & \frac{dH}{dx^b_2}(\mathbf{x^b}) & \dots & \frac{dH}{dx^b_N}(\mathbf{x^b}) \end{bmatrix}, \ R = \mathbf{e^{y^2}}$$

.

All that remains is to verify that there are no dimensional problems that prevent the interaction between them. Taking into consideration the equation defining **K**,

$$\mathbf{K} = \mathbf{P^b H^T} \left(\mathbf{H P^b H^T} + \mathbf{R}\right)^{-1},$$

we see that $\mathbf{P^b}$, being an N × N matrix, when multiplied with $\mathbf{H^T}$, an N × 1 vector, gives rise to an N × 1 vector, which, when multiplied to **H**, a 1 × N vector, forms a scalar:

$$\mathbf{P^b H^T} = \begin{bmatrix} P^b H_1^T \\ \vdots \\ P^b H_N^T \end{bmatrix}, \ \mathbf{H P^b H^T} = \begin{bmatrix} H P^b H_{1,1}^T \end{bmatrix} = H P^b H^T$$

.

The two addends in the round bracket are both scalars, therefore, the result of the addition involving them will itself be a scalar, which will be multiplied to $\mathbf{P^b H^T}$, that is a N × 1 vector, giving rise to a new N × 1 vector: **K**.

$$\mathbf{K} = \begin{bmatrix} P^b H_1^T \\ \vdots \\ P^b H_N^T \end{bmatrix} \cdot \left(H P^b H^T + R\right)^{-1}$$

$$\mathbf{K} = \begin{bmatrix} K_1 \\ \vdots \\ K_N \end{bmatrix}$$

$$\mathbf{K} : R^{N \times N} \times R^{N \times 1} \times R^1 \to R^{N \times 1}$$

Thus defined, the process of forming **K** works, and gives no dimension problems. Moreover, **K** fits perfectly into the optimal interpolation formula since, being an N × 1 vector, it can be easily multiplied with the scalar (y - $H(\boldsymbol{x^b})$), and added to $\boldsymbol{x^b}$, another N × 1 vector.

$$\mathbf{x^a} = \mathbf{x^b} + \mathbf{K} \cdot (y - H(\mathbf{x^b}))$$

$$\mathbf{x^a} = \begin{bmatrix} x_1^b \\ \vdots \\ x_N^b \end{bmatrix} + \begin{bmatrix} K\left(y - H(\mathbf{x^b})\right)_1 \\ \vdots \\ K\left(y - H(\mathbf{x^b})\right)_N \end{bmatrix} = \begin{bmatrix} x_1^a \\ \vdots \\ x_N^a \end{bmatrix}$$

$$\mathbf{x^a} : R^{N\times 1} \to R^{N\times 1}$$

We now have obtained $\boldsymbol{x^a}$, the result of the optimal interpolation process. Nonetheless, it is to be reminded that this optimal interpolation process is applied on two data sets that are not considered in their entirety. In fact, in order to avoid too much execution time, it was decided to operate a loop on the data set coming from TROPOMI, considering the individual elements of this set as y, and the corresponding elements in the data set coming from OLCI as $\boldsymbol{x^b}$. Thus, the result of the total assimilation process is obtained when the results of the individual assimilation processes that have been carried out during the loop are combined in a consistent manner. In other words, since the data from TROPOMI and OLCI are related to a geographic area of interest, the result of a single interpolation process must be placed in the same sub-area from which the TROPOMI data used as y and the OLCI data used as $\boldsymbol{x^b}$ originate.

3. Methods and instruments

3.1 The Alfred Wegener Institute and the PHYTOOPTICS group

The entire process aimed at acquiring the necessary knowledge to be able to develop the synergistic algorithm that is the focus of this thesis work, and the actual development of it, took place in Bremerhaven, Germany, at the offices owned by the PHYTOOPTICS group of the Alfred-Wegener-Institut Helmholtz-Zentrum für polar und meeresforschung.

The institute, whose name is commonly shortened to AWI, is dedicated to conducting research that can increase understanding of the mechanisms involving the seas, oceans, and polar regions of our planet. To do this, it makes use of various logistical tools such as icebreakers (like the Polarstern ship), aircrafts, observatories, field laboratories, and three permanent stations, two of which are located in Antarctica (Neumayer station III and Kohnen station) and one in the Svalbard Islands (AWIPEV arctic research base): the precise location of these stations is reported in the Figure 24 below.

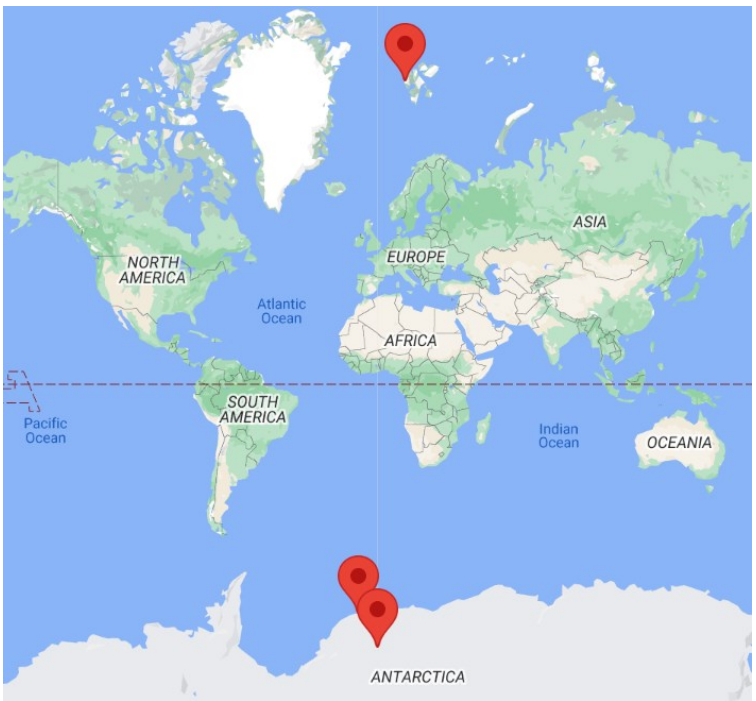

Figure 24 | **Location of AWI's permanent bases.** The marker at the top of the image represents the AWIPEV Arctic base, shared with the French polar institute Paul Emile Victor (IPEV), and located in Ny-Ålesund, Svalbard.

In contrast, the markings in the lower part of the map represent, the one at the top, Neumayer III station, and the one at the bottom, Kohnen station. (AWI, n.d.)

Scientific research within the institute is organized into three research divisions:

- Biosciences, directed by the Prof. Dr. Maarten Boersma.
- Geosciences, guided by Prof. Dr. Gesine Mollenhauer.
- Climate sciences, which refers to the Prof. Dr. Christian Haas.

The PHYTOOPTICS group, which welcomed me during my thesis course, is located in the research division that is interested in climate sciences, specifically, within the wing that deals with physical oceanography, under the supervision of Prof. Dr. Torsten Kanzow. The group has long been involved in various projects involving the use of tools to analyze the optical properties of our planet's waters in order to "contribute significantly to a better understanding of the attribution of anthropogenic and natural sources of climate change to the marine ecosystem and biogeochemical cycles" (AWI, n.d.). These projects are led by Prof. Dr. Astrid Bracher, to whom my sincere thanks go for allowing me access to the spaces and knowledge of her group.

I would also like to thank Dr. Leonardo Alvarado who, although he made a position transition in favor of DLR, the German space agency, found a way to follow me throughout my stay in Germany and in the developing of this work.

I would also like to take this opportunity to thank the members of the group – Moritz Zeising, Ehsan Mehdipour, Sonja Wiegmann, Christian Hohe, and Drs. Hongyan Xi and Mariana Altenburg Soppa – who gave me the pleasure of numerous edifying discussions through which I was able to better understand some of the concepts that were fundamental to the writing of this paper. In addition, thanks to the support of the group, I had the opportunity to experience a week-long field expedition at Lake Constance and to be able to present the progress of this project at an international conference for young researchers, in Oldenburg.

I am very grateful for the opportunities I received, and I am sure that the skills and knowledge I gained this year at the PHYTOOPTICS group have enriched and will enrich my academic career and personal growth path.

Figure 25 | **PHYTOOPTICS group logo.** (AWI, n.d.)

3.2 Polymer and OC-PFT application

Once the theoretical background needed to baste a process of assimilating two datasets was obtained, it was necessary to determine which datasets from OLCI and which datasets from TROPOMI would be used in this process. The choice fell on the datasets that the instruments produced from 01/05/2018 to 30/06/2018, as the research team has access to an *in situ* dataset, covering the same period, that could be used for the validation of the algorithm. In other words, once the drafting of the algorithm and the verification of its operation is completed, the results that the algorithm produces using real data as input could be compared with the data from the *in situ* measurements to have an element of evaluation of how well the codes work.

Once the datasets to be assimilated had been chosen, the first issue that needed to be addressed concerned the need to "prepare" the data coming from OLCI for the assimilation process: in other words, a way had to be found to apply the Polymer and OC-PFT algorithms to this kind of data. Remember that, thanks to the DOAS technique, data coming from TROPOMI can be easily brought to level 2A, while, to reach the same level, data coming from OLCI should be processed with Polymer and OC-PFT.

A necessary condition to proceed in the processing of OLCI data was to choose their resolution type and download them from the catalog of datasets held by ESA. It was decided to proceed with the downloading of reduced-resolution data, which means data related to pixels of about 1 km side, because, compared to the full-resolution OLCI data, there was the possibility of accessing the errors associated with these data, an aspect of fundamental importance for constructing the covariance matrices mentioned in Chapter 2.3. In addition to the desired period and resolution, during the downloading process it was necessary to specify the area of the Planet to which the data would refer: a rectangular area overlapping the Atlantic Ocean was chosen. More details regarding the characteristics of the downloaded OLCI data can be found in the Appendix at the end of the document.

The download operation was conducted through the adaptation of a Python code devised by Ehsan Mehdipour, a doctoral student in the PHYTOOPTICS group, which made it possible to store all the files of interest in a folder of Albedo, the high-performance computer (HPC) at AWI's disposal, in the possession of Dr. Leonardo Alvarado.

Once this point was reached, since for the application of both Polymer and OC-PFT it would have proved much more convenient to have data sorted by reference date, one of the first codes that was written was devoted to the solution of this problem.

3.2.1 OLCI data sorting

The purpose of the Python code named `OLCI_zip_sorting` was to move the compressed OLCI files from the folder in which they had been uploaded to a folder in my personal space in Albedo, in which they would then be automatically sorted according to the referenced date. We speak of compressed files because each of the OLCI files that `OLCI_zip_sorting` proposed to sort consisted of a folder containing several databases related to the date to which the folder referred.

The logic behind `OLCI_zip_sorting`, explained in detail in the Appendix, rests on the nomenclature conventions of OLCI files, which, following the European Space Agency's scheme (Sentinel Online, n.d.), have in their names a triplet of temporal information, the first element of which indicates the time when the measurement was started, called the "sensing start time". The files downloaded from the ESA catalog were in the form shown in the figure below, and to give an example of what was mentioned above, the first of them is a file containing databases whose data began to be retrieved on May 1, 2018.

```
S3A_OL_1_ERR____20180501T000437_20180501T004857_20180502T042807_2660_030_330______LN1_O_NT_002.zip
S3A_OL_1_ERR____20180501T014536_20180501T022956_20180502T060229_2660_030_331______LN1_O_NT_002.zip
S3A_OL_1_ERR____20180501T032635_20180501T041055_20180502T074359_2660_030_332______LN1_O_NT_002.zip
S3A_OL_1_ERR____20180501T050734_20180501T055154_20180502T092439_2660_030_333______LN1_O_NT_002.zip
S3A_OL_1_ERR____20180501T064833_20180501T073253_20180502T110058_2660_030_334______LN1_O_NT_002.zip
S3A_OL_1_ERR____20180501T082932_20180501T091352_20180502T124500_2660_030_335______LN1_O_NT_002.zip
S3A_OL_1_ERR____20180501T101030_20180501T105451_20180502T142414_2661_030_336______LN1_O_NT_002.zip
```

Figure 26 | **Some among the compressed OLCI files that were downloaded for this thesis work.**

Being fairly contained, the function that characterizes `OLCI_zip_sorting` is written and called in a single script, so that anyone who needs to use it can simply run the script, taking care to enter in the script the name of the folder in which the downloaded folders are randomly stored, and the name of the folder in which it is wanted to store them according to their reference date.

3.2.2 Application of atmospheric correction (Polymer) to OLCI data

Once the OLCI files were arranged according to the reference date, thought was given to how to subject them to atmospheric correction through Polymer. First, it was necessary to install Polymer through the appropriate channels, i.e., through the website of the HYGEOS

company (HYGEOS, n.d.), and then find a way to be able to circumvent a computational length problem. In fact, while a contained script such as `OLCI_zip_sorting`, which is used to conduct simple operations on a small amount of data, can be executed without problems through the resources of the supercomputer node in which it is located, applying an algorithm such as Polymer to a very conspicuous amount of data such as the OLCI data of our interest can be a problem.

HPCs such as Albedo possess a request management system that divides user-ordered work among its compute nodes (Figure 27), and this allows more onerous tasks to be completed than would be possible for a single compute node to perform. In the case of Albedo, the user is responsible for writing a file in Bash in which he enters some specifications about himself and the job he is requesting to be done, the maximum time he believes should be devoted to his request, and the instructions he wishes to be executed: once the command to execute that Bash file is run, Albedo's request management system will take care of allocating the resources of different computation nodes to fulfill the user's wishes.

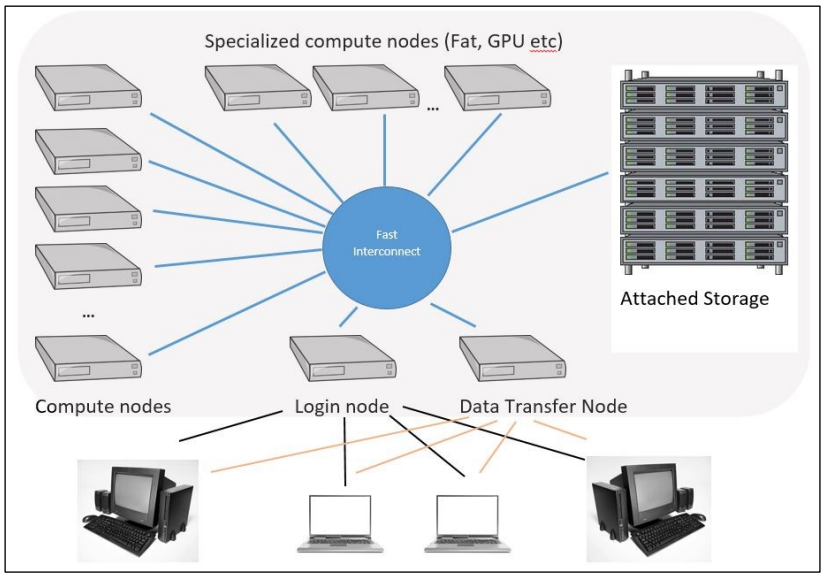

Figure 27 | **Diagram of the structure of a general HPC. (**Iowa State University, n.d.)

Unfortunately, even taking advantage of this job splitting system, applying Polymer to the entire OLCI data set would have been too time-consuming. Therefore, it was necessary to find a way to reduce the weight of the individual job that was being submitted to the request management system. Since the original intention was to request Albedo to apply Polymer to a 61-day dataset, it seemed a good idea to request the HPC to apply the algorithm to 61 one-day

datasets. In other words, the resource management system was required to handle many jobs in parallel, but with each job having a low specific weight.

In order to operationalize this idea, it was necessary to find a way to automate the parallelization process, and for this reason a Python code called `new_polymer_submission` was conceived. This code consists of two parts:

- **Creation and submission of requests to Albedo**

  This part of `new_polymer_submission` relies on a command line interface (CLI) developed by Dr. Leonardo Alvarado, which allows Polymer to be launched on a data set through the simple use of the command

  ```
  polymer_run   path-to-config_file
  ```

  where:

  - `polymer_run` is the command that refers to `path-to-config_file`.
  - `path-to-config_file` is the path that tells `polymer_run` where to find a configuration file called `config_file`. In that file are marked the specifics of the data to which Polymer should be applied, such as, for example, directions to find the folder in which they are stowed, the period of interest (in our case, the days from May 1 to June 30, 2018), the resolution of the data, etc…

  It is evident that being able to apply Polymer to a dataset through the use of these two simple elements (the command, and the configuration file) is a great facilitation for the work that was necessary to do. Hence, the first part of `new_polymer_submission` was structured in order to produce a desired number of Bash files (61, in our case) to be delivered to the HPC request handling system: each of these files would order the supercomputer to execute a command similar to the one given above. Further details about the functioning of this part of `new_polymer_submission` are available in the Appendix.

- **Error management**

  Each of the requests that are submitted to the supercomputer, however, may result in an error that, given their large number, would be difficult to detect without the proper tools. For this reason, while creating `new_polymer_submission`, it was thought it would be appropriate to devote a portion of the code to creating a folder in which all error messages coming from the HPC would be stowed in an orderly manner. To do

this, a folder was created, called `pol_sub`, and a text file that would act as a counter. When `new_polymer_submission` was started, the code would look for the counter file in a predefined path (represented by the variable `counter_loc`): in the case where `new_polymer_submission` was started for the first time, the counter file would automatically be created in `counter_loc` and a 0 would be written inside it, while in the case where the code had been run before, the counter file would simply be read.

In either case, the number found in the counter file would be used as the name for a folder to be created within `pol_sub`, where all supercomputer communications generated by the current execution of `new_polymer_submission` would be stored. For this process to happen, it is necessary to specify during the creation of the `.sh` scripts (the files delivered to the HPC's request handling system) that the communications related to that job submission should be stored in the numbered folder mentioned above. In this way, within `pol_sub` there will be a set of numbered folders within which there will be a series of files, named according to the job submission from which they were generated. In such a context, it is quite easy to keep track of errors: just run `new_polymer_submission` and, in case of unforeseen problems, move to `pol_sub`, look for the folder marked by the higher number and consult its content.

Clearly, for this mechanism to hold, it is necessary for `new_polymer_submission` to update by one unit the number marked in the counter file, after using it to create the folder in `pol_sub`.

`new_polymer_submission` proved quite efficient and accurate in doing the job it was designed to do, and once Polymer was applied to the OLCI data of interest, the next step could be tackled: applying OC-PFT to it.

3.2.3 OC-PFT CLI testing and application to OLCI data

The processing of OLCI data by OC-PFT, conceptually speaking, was quite simple. In fact, a mode of operation very similar to that used with Polymer was followed. Dr. Alvarado devised a command line interface that would allow, in the same way as `polymer_run`, OC-PFT to be applied to a data set by exploiting the command:

`ocpft path-to-config_file`.

Unlike `polymer_run`, whose use had become established over time, the development of the CLI for OC-PFT ended at the same time as the development of `new_polymer_submission`, which made it necessary to spend some time testing it.

Once the testing phase was successfully concluded, a Python code, `ocpft_submission`, was created, which functioned according to the same logic as `new_polymer_submission`, but changing the information contained in the configuration files and the names of the folders in which to store the Bash files, the configuration files themselves, and the error notifications.

The successful performances of `ocpft_submission` and the command line interface made it possible to apply OC-PFT to OLCI data, bringing them to level 2A. Since the TROPOMI level 2A data for the period 01/05/2018-30/06/2018 were already in the possession of the lab staff, we had everything we needed to begin the development of the assimilation algorithm. At the end of the development of the latter, and of the testing phase, the level 2A data from both satellites would be subjected to the gridding process, and finally processed through the algorithm itself.

3.3 Assimilation algorithm development and structure

Given that the studies by Dr. Losa and colleagues on the assimilation of data collected by OC-CCI and SCHIAMACHY had led to the production of the SynSenPFT algorithm (Losa *et al.*, 2017), the initial idea for the development of the algorithm underlying this thesis work was aimed at the simple translation of SynSenPFT from Fortran to Python. Once this quick, it was thought, translation work was completed, an algorithm would have been ready to be applied to data from OLCI and TROPOMI.

However, despite the implementation of several strategies, the process of translating the algorithm was very difficult, given the extent and complexity of the original code, and time was running out, as it was decided to present this project in Oldenburg, at the International Conference for Young Marine Researchers, to be held between September 18 and 22, 2023 (ICYMARE, n.d.).

In the face of these problems, it was decided to abandon the translation attempt, and embark on a different endeavor: creating an algorithm from scratch. Of course, compared to a translation process it would certainly prove more challenging, but this perspective offered total

control over the logic behind the algorithm and, therefore, a better understanding of it. It was therefore decided to structure the algorithm in three parts:

1. file handler script – in this script would be concentrated all the functions necessary for data input in the assimilation process, and for the management of the output data, which includes their saving and eventual graphical representation.
2. `processor` script – this script would contain the functions dedicated to the actual assimilation process.
3. main script – would serve as a management point for the previous two scripts, fitting their fragments into a congruent workflow.

More detailed descriptions of these scripts will be given in Sections 3.3.1, 3.3.2 and 3.3.3.

3.3.1 File handler script

This script contains four functions devoted to the following tasks:

- **Produce a list of files that satisfy a certain characteristic**
  The `list_files` function, once the folder where the files of interest are contained, their extension, and the dates to which the files should refer are specified, is responsible for returning a list containing all files that satisfy the imposed conditions.
- **Extracting essential information from the files**
  The level 2A data files of interest to are `DataArray` objects of Xarray (Xarray, n.d), stored in netCDF4 format (Rew *et al.*, 1989), and it is necessary to extract from them the information needed for the assimilation process, namely, the values recorded by the instruments in each pixel they screened, and the corresponding coordinates of the pixels. A `DataArray` is nothing more than a kind of meta-structure, a wrapper, around a set of n-dimensional NumPy arrays (NumPy v1.26 Manual, n.d.), which makes it possible to associate arrays containing different information by distinguishing them into categories: there are arrays that contain data related to the variables of interest, coordinate arrays, which contain values representing a coordinate type to which the data of interest refer, and attributes, i.e., a whole range of information useful for being able to interpret the data. An example of an attribute might be the units of measure associated with the values of the variables, or a brief description of them.

The `load_dataset` function takes charge, once the file, i.e., the `DataArray` to be accessed, and the name of the variable of interest are indicated, of returning the array representing the variable and the arrays containing the latitude and longitude values that relate to the variable values.

- **Saving the result of the assimilation process**

  The `output_saving` function takes care of loading the `DataArray` result of the assimilation process into an indicated directory, and then providing the path to it: it will be useful during the operations that will be conducted in the main script. Since it was envisioned that most of the assimilation processes would involve OLCI and TROPOMI data related to an extended period, it was thought to equip `output_saving` not only for saving the results in a directory, but also for their orderly arrangement. To this end, it was planned to use the date marked on the OLCI files (an arbitrary choice, driven simply by the immediate availability of the OLCI files in order to test the mechanism) as a reference for the following operation: in the event that the folder designated as the container for the results does not exist, `output_saving` will create it, inserting within it a sequence of folders reflecting the date indicated above. This will result in a sequence of folders reflecting the following pattern:

  ```
  destination_directory/yyyy/mm/dd
  ```

  Here, the `DataArray` obtained through the assimilation process will be saved in netCDF4 format, and named as:

  ```
  synergistic_product_yyyy/mm/dd.nc
  ```

- **Graphically represent the result of the assimilation process**

  The last function of this script is called `plotter`, and it allows users to visualize a graphical representation of the results of assimilation operations using the Cartopy (Cartopy 0.22.0 documentation, n.d.) and Matplotlib (Matplotlib, n.d.) libraries. `plotter` requires only the input of two parameters to work: the path to which to find the dataset to be represented (which is why it is useful for `output_saving` to return it), and the name by which the NumPy array to be represented was named.

3.3.2 Processor script

This script contains the two functions that, pragmatically, deal with the interpolation process: `optimal_interpolation` and `processor`. As reported in Chapter 2.3, the interpolation process is not done "in one go" but, to avoid operations with overly voluminous data masses, it proceeds pixel by pixel of the TROPOMI grid. It is therefore necessary to have a function that is responsible for scrolling through each pixel on the grid in which the TROPOMI data are arranged, selecting the corresponding pixels on the grid in which the OLCI data are arranged, and passing the values contained in both types of pixels to another function that is responsible for operating the assimilation of them. The first function, the one that flows between the grids, is `processor`, while the one that operates the assimilation process is `optimal_interpolation`. Let's go into more detail.

- `processor`

  The conception of `processor` was quite complex. First, it was necessary to try to figure out how to define pixels in such a way as to iterate on them quickly: once the gridding procedure is completed, pixels are basically rectangles or squares, which means that they can be defined by four points, the corners, that delimit their area. In the first instance, it was chosen to proceed by identifying each pixel with only one of these corners, specifically, if we imagined the vertices of a regular quadrilateral, that of the upper left point. For the sake of brevity, I will refer to this point as UL (Upper Left), while I will denote the other points that define a pixel as UR (Upper Right), LL (Lower Left), and LR (Lower Right).

  The first idea that came up to try to run `processor` was a quadruple for loop that would consider all the UL points in the TROPOMI grid, and for each of them, consider all the UL points in the OLCI grid to check whether a series of conditions were met: if the UL-UR segment of an OLCI pixel is included in the UL-UR segment of the TROPOMI pixel, and if the UL-LL segment of the same OLCI pixel is included in the UL-LL segment of the same TROPOMI pixel, then the value contained in the OLCI pixel will be included in a list to be passed, along with the value contained in the TROPOMI pixel, to the `optimal_interpolation` function.

  Theoretically, such a mechanism should have worked, and indeed it did, but it was practically unusable on large grids because of its slowness. To solve this problem, it

was thought to use Numba decorators (Numba, n.d.), i.e., to rely on a compiler that would translate the Python language in which the quadruple for loop was written into a language more similar to machine language, so as to make it faster. Indeed, applying the Numba decorators to the loops showed some small improvement, but unfortunately, in order to make them compatible with the libraries used in the rest of the code it was necessary to make so many changes to the it that the benefits obtained from the decorators became negligible. After a long period of exploring this strategy, it was decided to abandon it in favor of another solution: boolean masks.

Rather than sifting through whether each individual pixel in the OLCI grid meets the above conditions, the Boolean masks allow all pixels that do not meet them to be "deleted" from that grid. The name of this technique derives from the fact that Boolean masks are nothing more than matrices filled with 1s and 0s, distributed according to the conditions desired by the user, which are multiplied to the data grid of interest: all data that do not meet the conditions according to which the masks were constructed become zeros, while those that do meet them remain unchanged.

Imagining OLCI grids as N × M matrices, what the use of masks allowed to obtain was the set of indices (n, m) of the matrix elements that satisfied the desired conditions. To give a graphic illustration of how one could cycle over the OLCI matrix by exploiting Boolean masks, imagine a TROPOMI matrix defined by 9 elements and an OLCI matrix defined by 81 elements: for each of the TROPOMI elements, one must check which elements of the OLCI matrix are included in its UL-UR and UL-LL segments, and this can be done by maintaining a fixed condition structure, in which the coordinates of the UL-UR and UL-LL segments simply vary. In this way, with only a double for loop and nine masks, one can cycle over the TROPOMI grid and obtain the corresponding pixels of the OLCI grid. We saved ourselves 729 (9 × 81) condition-checking operations.

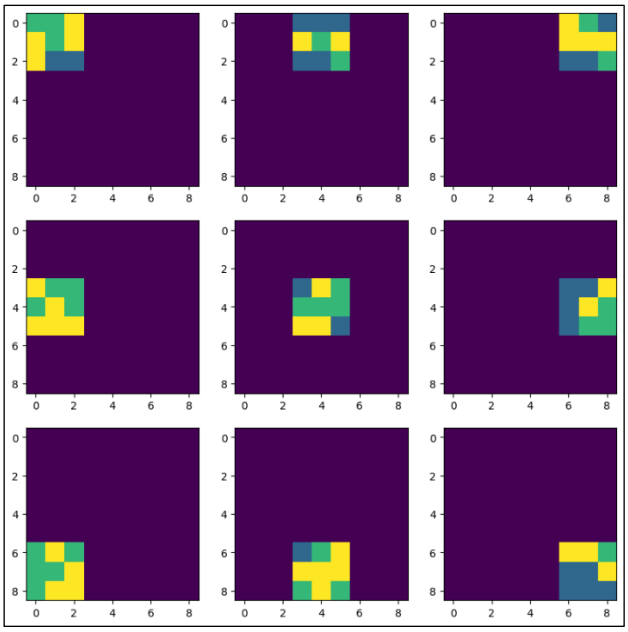

Figure 29 | **Graphical representation of cycling over a grid with the help of Boolean masks.** Applying a Boolean mask to a data grid means multiplying each element of that matrix with the corresponding value within a matrix, of the same size, which, however, contains only numbers 1 and 0. The distribution of 1's and 0's follows a certain condition, based on which the shape of the mask is defined. By varying the mask shape 9 times, the number of pixels in an imaginary grid of TROPOMI data, one can easily select the pixels in an imaginary grid of 81 OLCI pixels.

Implementing such a solution significantly improved **`processor`** performance, but it was felt that it could be improved even more. In fact, although the mask strategy had eliminated the two for cycles related to the OLCI grid, there were still two left that were used to cycle on the TROPOMI grid. How to reduce the number of cycles? The idea that came to mind revolved around the fact that the information contained in the grids could have been summarized in arrays with this structure:

$$[[m_i,\ n_i,\ val_i],\ [m_j,\ n_j,\ val_j],\ \ldots]$$ [2]

where:

- i and j represent two different pixels.
- m_i, n_i, val_i represent, respectively, the row index of the value contained in the pixel i, the column index of the same value, and the value associated with the pixel.
- m_j, n_j, val_j are equivalent to what is given in the previous point, but refer to the pixel j.

[2] For the sake of convenience, in this text the array is shown as a horizontal array, but it is a vertical array.

The use of such structures, and knowledge of the latitude and longitude resolutions of OLCI and TROPOMI, would have made it possible to operate only a single for loop on the array of arrays representing the TROPOMI matrix: for each of its points, let us imagine the generic point $[m_{TROP}, n_{TROP}, val_{TROP}]$, masks would be applied to the other array of arrays, the one representing the OLCI array, to see which elements would satisfy the following conditions:

$$m_{OLCI} \geq m_{TROP} * lat\_ratio$$
$$m_{OLCI} < (m_{TROP} * lat\_ratio) + lat\_ratio$$
$$n_{OLCI} \geq n_{TROP} * lon\_ratio$$
$$n_{OLCI} < (n_{TROP} * lon\_ratio) + lon\_ratio$$

where:

- m_{OLCI} and n_{OLCI} represent the row and column indices of the values contained in a generic OLCI pixel.
- `lat_ratio` and `lon_ratio` are, respectively, the ratios between the resolutions in latitude of OLCI and the resolutions in latitude of TROPOMI, and the ratio between the resolutions in latitude of OLCI and the resolutions in latitude of TROPOMI.

Effectively, using this strategy seemed to further improve `processor` execution times. An attempt was made to quantify this improvement by comparing the performance of the three `processor` versions as the size of the input data grids increased: from 1 degree latitude by 1 degree longitude to two and then to three.

Imagining that the resolutions of OLCI and TROPOMI are equal in both latitude and longitude, and are, respectively, 0.01 degrees and 0.05 degrees, six square data grids were created, three with a number of pixels congruent with the resolution of TROPOMI, three with a number of pixels congruent with the resolution of OLCI, and all filled with random real numbers. To give an idea of the size of these grids, figures 30, 31, and 32, shown below, were created, in which the matrices are placed with their center in the point (20 °E, 35 °N). For convenience, only the three matrices with a number of pixels congruent to the resolution of OLCI are shown: the others would have had identical shape, and they would be placed at the same position: the only difference would have been in the number of pixels.

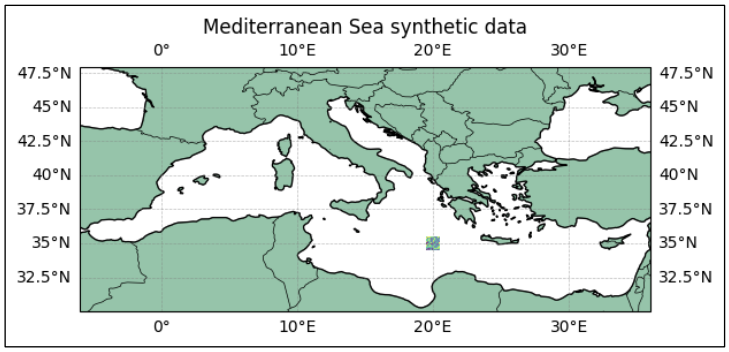

Figure 30 | **1° × 1° matrix size, compared to the Mediterranean Sea.**

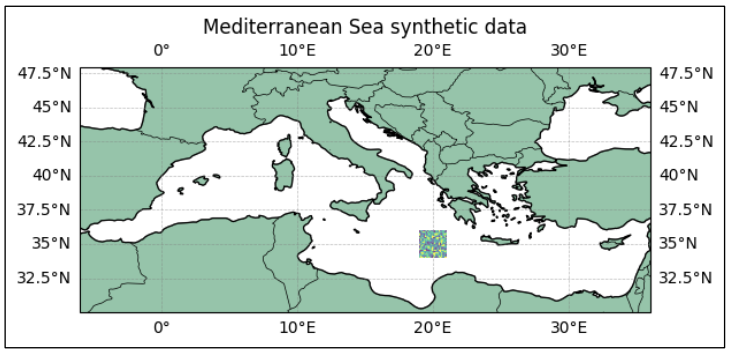

Figure 31 | **2° × 2° matrix size, compared to the Mediterranean Sea.**

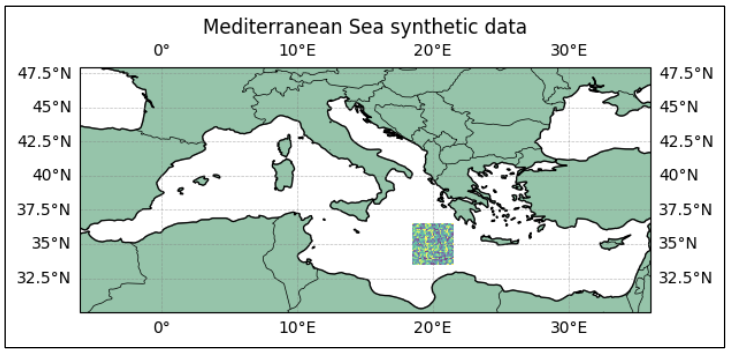

Figure 32 | **3° × 3° matrix size, compared to the Mediterranean Sea.**

At this point, the six grids were fed to the three `processor` versions, distinguished by the labels P0 (indicating the quadruple for loop), P1 (representing the double for loop in conjunction with the masks), and P2 (indicating the version with a single for loop used in combination with the masks). The resulting running times are shown in Figure 33 below. As can be seen, at the slightest increase in the size of the arrays on which to iterate, the performance of P0 deteriorates dramatically, while that of P1 and P2 does not seem to change too much, remaining around 0 seconds. It is thus understood that P0 is not a viable option for processing large masses of data, but there seems to be no difference between P1 and P2. Which one to choose?

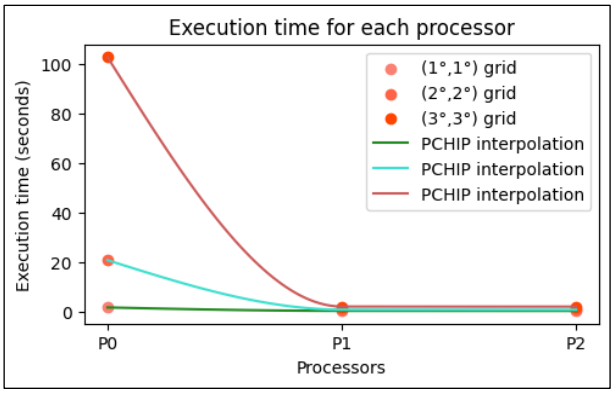

Figure 33 | **Running times of the three versions of `processor` at the increasing of the input data grids' size.** The running time values were interpolated using a Piecewise Cubic Hermite Interpolating Polynomial offered by the `interpolate` function of the SciPy library. (SciPy v1.12.0 Manual, n.d.)

To figure out which **`processor`** to choose between P1 and P2, another square matrix was created, this time of 5 degrees × 5 degrees, and the differences between the execution times presented by the two versions of **`processor`**s when employed in operations on matrices of 1 degree × 1 degree, 2 degrees × 2 degrees, 3 degrees × 3 degrees, and 5 degrees × 5 degrees were calculated. A graphical representation of these differences is shown in Figure 34 below. As can be seen, it seemed evident that as the amount of data to be processed increases, P2 is faster than P1. Hence, P2 was thought to be the version of **`processor`** that would have been maintained in the code.

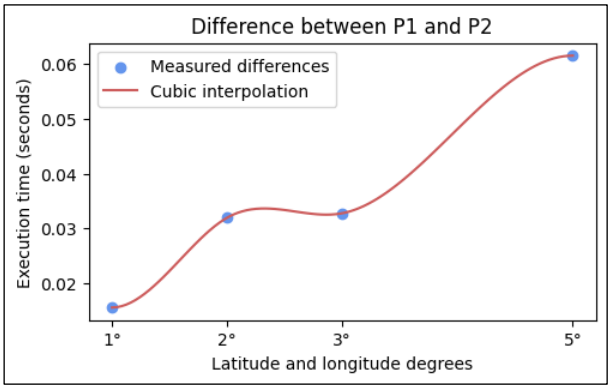

Figure 34 | **Difference between P1 and P2 performances at the increase of the volume of input data.** The results of these comparisons were interpolated using a cubic polynomial offered by the `interpolate` function of the SciPy library. (SciPy v1.12.0 Manual, n.d.)

- **`optimal_interpolation`**

  Once **`processor`** offers a value of TROPOMI, and the corresponding values of OLCI, **`optimal_interpolation`** takes care of conducting the assimilation operations, according to the equations described in Chapter 2.3. To do this, **`optimal_interpolation`** takes advantage of several functions in the NumPy library

that allow to vectorize operations between arrays, making them much faster than they would be if done otherwise (for example, if done by looping over the arrays).

Moreover, among the input data it needs, `optimal_interpolation` requests a matrix full of zeros, of number of elements equal to the general OLCI matrix. In fact, once the process of interpolation between a subset of the OLCI matrix and a subset of the TROPOMI matrix is completed, `optimal_interpolation` takes care of inserting the result into the above matrix, substituting the zeros with the appropriate values. The insertion is done by respecting the coordinates of the subsets that were used during the assimilation process; in this way, the matrix full of zeros updates from interpolation to interpolation, becoming the synergistic matrix that was being sought.

`optimal_interpolation` does not work sequentially with `processor` but is called within `processor`. Precisely, each time `processor` detects a TROPOMI pixel-OLCI pixels pair, it calls `optimal_interpolation` on this element: at the end of the cycle that `processor` operates, the matrix that was full of zeros at the beginning of that cycle will be loaded with the values that are the result of assimilation. `processor` takes care of loading that matrix inside a `DataArray`, ready to be passed to the rest of the functions contained in in the main script.

3.3.3 Main script

The last script that characterizes the assimilation algorithm is the main script, defined by a single function: `main`. This function is responsible for orchestrating the work of all those previously described in the following way:

1. Having defined a period of interest for the assimilation process, the paths to the two folders in which the netCDF4 OLCI files and the netCDF4 TROPOMI files are located must be given. In case it is desired to proceed in chronological order, `main` takes care of sorting the files according to the date they refer to and placing them in two lists using the `list_files` function: one list will contain all files containing data from TROPOMI, and the other list will contain all files with data collected from OLCI.
2. For each pair of files from the lists, main extracts the datasets of interest through the use of `load_dataset`, and passes them to `processor`, which returns a result `DataArray`.

3. The result is then passed to `output_saving` and loaded into an appropriate directory, which `plotter` simultaneously accesses to produce a graphical representation of it.

With the main script, the structure of the assimilation algorithm was complete. It was now necessary to test its functioning.

4. Results and outlook

To make an initial verification of the algorithm's functioning, synthetic data were used: two-dimensional arrays containing concentration values related to different PFTs were thought to be simulated using two-dimensional arrays filled with randomly generated real numbers. To give greater realism to such data sets, the values within them were thought to oscillate between 0.0 mg/m^3 and 3.0 mg/m^3. These numbers were then associated with errors of different types, chosen arbitrarily to be quite similar to those found in satellite datasets. In the case of the arrays that simulated arrays of data from TROPOMI, each value was associated with an error equal to half of the value itself, while the synthetic data representing the values obtainable from OLCI were randomly associated with errors between 0% and 50% of the values themselves. It was also planned that the ratio of the resolution of the simulated TROPOMI data to that of the simulated OLCI data would be equal to 5 in both latitude and longitude: the pixels created would thus be square, and in the area covered by one TROPOMI pixel there would be 25 OLCI pixels.

The first version of the algorithm that was subjected to square matrices filled with this type of data was the one containing the first version of the processor function, named previously as P0. An example of the results obtained is shown in the figure below:

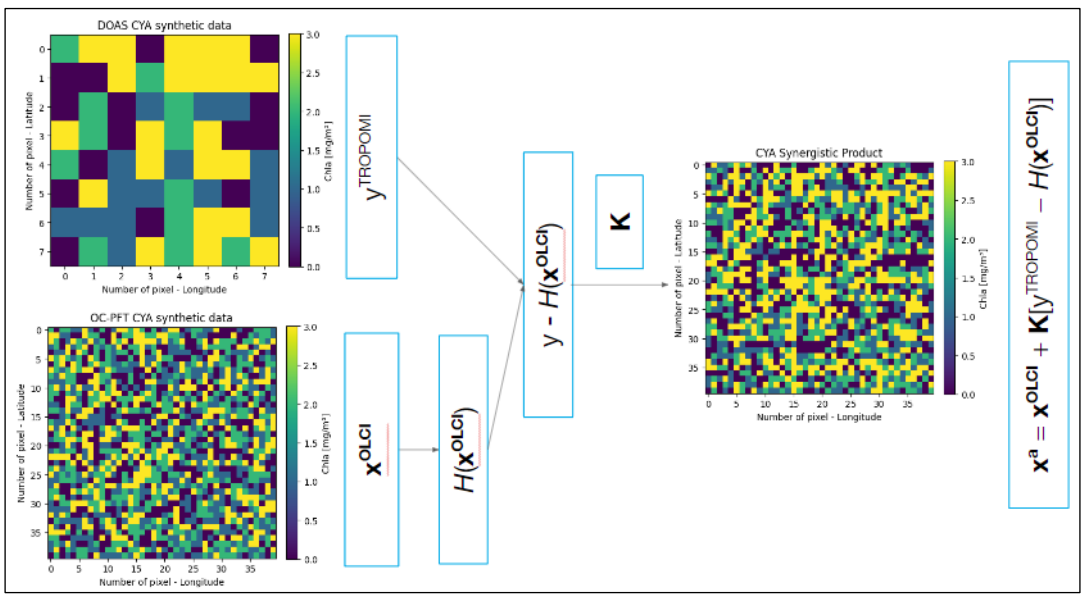

Figure 35 | **Results obtained by input of synthetic data in the first version of the algorithm.** Each of the matrices shown in the figure represents fictitious cyanobacteria concentration values at a specific location whose latitude and longitude are represented by the row and column indices corresponding to the value. The matrices are named "DOAS CYA synthetic data" and "OC-PFT CYA synthetic data" to indicate the fact that the data within them simulate what would be obtained at the end of applying the DOAS technique and the OC-PFT algorithm. Also shown in the figure is a small diagram that reassumes the steps in the optimal interpolation process.

As can be seen, the first tests were not satisfactory, given the appearance of the matrix resulting from the assimilation process: the values reported within it are different from those within the matrix simulating an OLCI data set, but not in the right direction. In fact, one of the aspects that should become apparent after the optimal interpolation operations is the influence of the synthetic TROPOMI data on the synergistic grid. For example, if a pixel simulating a TROPOMI pixel contains a value of 0 mg/m^3 chlorophyll, the success of the assimilation process will be qualitatively evident if the resulting matrix reports, at the same coordinates, values that are significantly lower than those in the OLCI grid. In other words, the contribution of the TROPOMI synthetic data during the assimilation process must be evident, and the results of the first tests did not go in this direction: when represented graphically, the resulting matrices looked like jumbles of randomly generated numbers similar to those used as input data.

After careful investigation of the first version of the algorithm, the catch turned out to lie in the mechanisms that should have ensured the formation of the $\mathbf{P^b}$ covariance matrices described in Chapter 2.3. Instead of creating covariance matrices representing an entire set of OLCI pixels, the `optimal_interpolation` function employed only 1/5 of that set in the formation of the $\mathbf{P^b}$ matrices. Solving this problem involved several changes in the `optimal_interpolation` function, which were made in parallel with the upgrade of the `processor` function from version P0 to version P2.

By subjecting the new version of the algorithm to data identical to those described above – except for the values contained in the matrices, which, as before, were randomly picked from the range of real numbers between 0.0 mg/m^3 and 3.0 mg/m^3 – the results proved to be much more promising. As can be seen from figures 36 and 37, below, for example, the influence of TROPOMI pixels in the calculation of the synergistic product is much improved over what was shown in the previous figure.

The idea of evaluating the quality of the algorithm's performance based on the influence of TROPOMI pixels on the results of the assimilation process came from observing the operation diagram of the SynSenPFT algorithm (Losa *et al.*, 2017), shown in Figure 38. To be precise, Figure 38 – and the equations described in Chapter 2.3 – shows that a good assimilation process takes into account both model and observation inputs, however, given the characteristics of the

synthetic data on which the algorithm was tested, checking the influence of TROPOMI data on the final result seemed a good strategy to make a quick qualitative assessment of how the system works. In fact, as can be seen in Figures 35, 36, 37 and 39, the matrices simulating grids of TROPOMI data are full of zero values (the purple squares, to be clear). Figure 38 shows that, if a pixel contains a value equal to zero, that value will be maintained throughout the assimilation process: in fact, the value of some pixels in the upper left corner of the OC-CCI data matrix, close to or equal to zero, was also maintained as such in the results matrix. Since in the synthetic data matrices used to test the algorithm OLCI pixels are many and difficult to distinguish, and since, on the other hand, there is ample provision of larger TROPOMI pixels containing zero values, it seemed a good strategy for evaluating the operation of the algorithm to see whether, in the areas of the results matrix corresponding to such TROPOMI pixels, values of pixels close to zero, or at least smaller than those detectable in the starting OLCI matrix (in case these were very positive), were concentrated.

Another aspect that Figure 38 suggests is that a process of data assimilation by optimal interpolation may result in slightly higher values being present in the results matrix than the maximum values contained in the source matrices: if one looks closely, the last row of the SynSenPFT results matrix has two more positive pixels than the corresponding ones in the model matrix. As can be seen, while this aspect is missing in Figure 35, it is present in Figures 36 and 37.

So, the greater similarity between the results that the algorithm was producing and those produced by SynSenPFT made it possible to decree that the system was working well.

The implementation of the P2 version of the processor function involved also another testing phase in which matrices were used that, from the point of view of the characteristics of the data within them, were identical to those just mentioned, but it was decided to vary their shape in order to test the algorithm's ability to handle rectangular-shaped matrices as well (recall that the area chosen for the interpolation operations spans the Atlantic Ocean as a large rectangular polygon). The results of this testing phase were positive: as shown in Figure 39, the algorithm confirmed the performance of which it had been capable previously, and also demonstrated its ability to handle rectangular-shaped matrices.

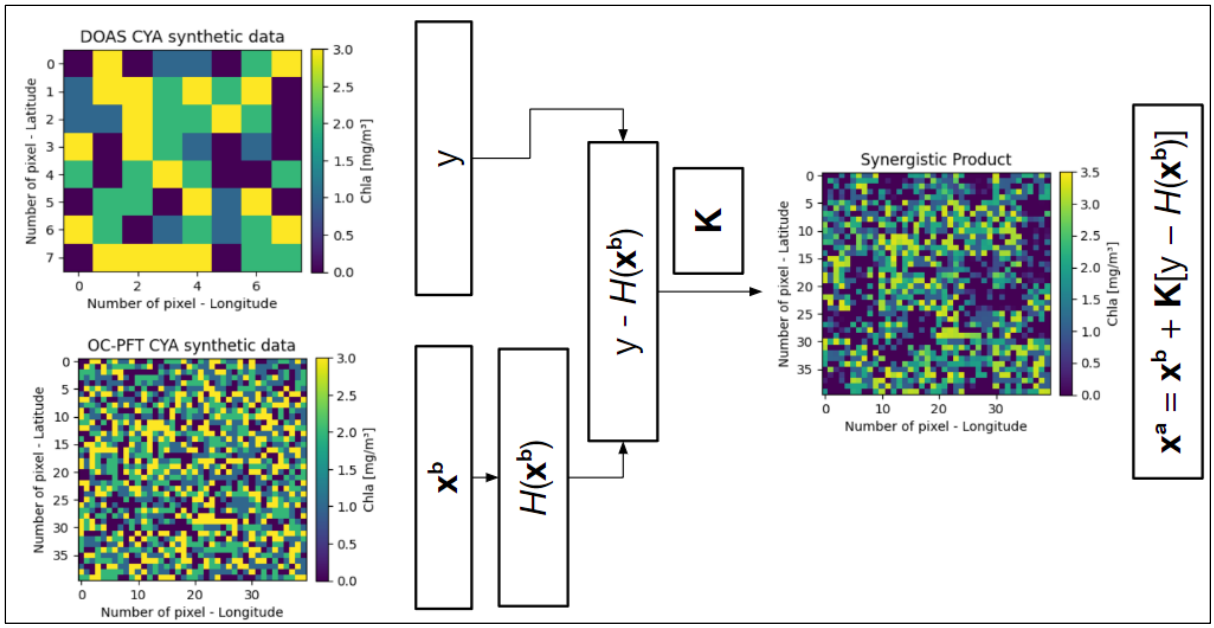

Figure 36 | **Testing of the new version of the algorithm.** `processor` was updated from P0 to P2, `optimal_interpolation` was fixed, and the testing procedure was repeated. As can be seen, the influence of TROPOMI pixels in the interpolation process is much more evident than in Figure 35. Again, a small diagram recalling the passages of the interpolation process is shown.

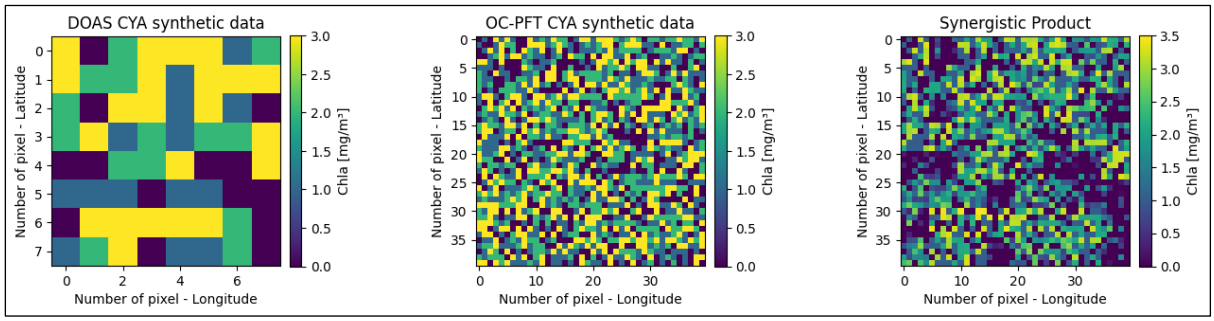

Figure 37 | **Further results obtained through the algorithm containing the P2 version of `processor`.** In this case, only the two input matrices and the output matrix of the interpolation process are shown.

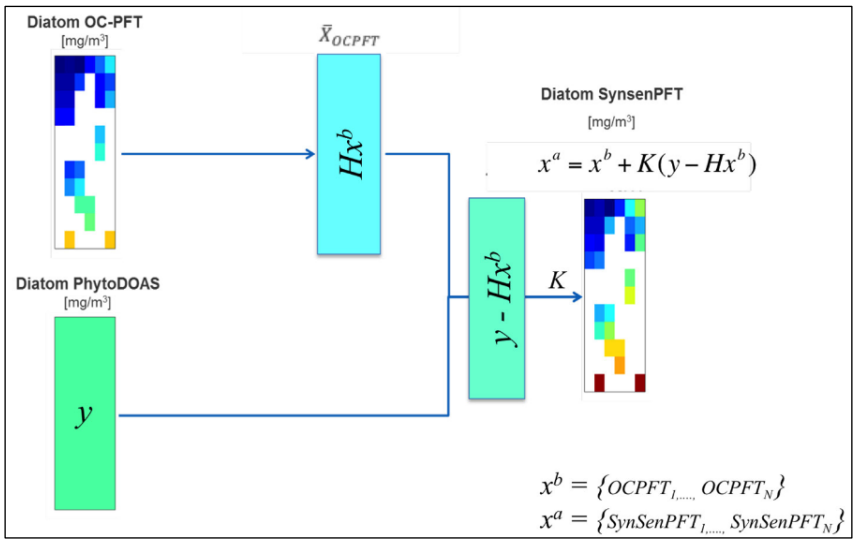

Figure 38 | **Diagram illustrating the operation of the SynSenPFT algorithm.** The algorithm was developed to fuse data from the SCIAMACHY hyperspectral instrument and OC-CCI, multispectral instrument. Data from SCIAMACHY are brought to level 2A using the DOAS technique (reported here as PhytoDOAS), while data from the OC-CCI detections are brought to the same level using the Polymer and OC-PFT algorithms. Thus, the rectangles on the left represent two matrices containing the diatom PFT concentration values to be used as observations (y) and models ($\boldsymbol{x}^b$). (Losa *et al.*, 2017)

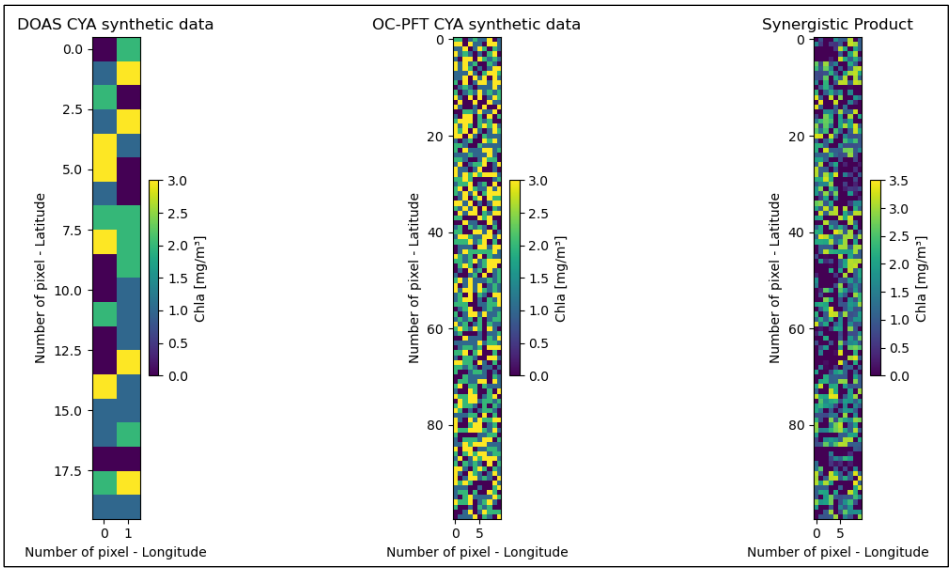

Figure 39 | **Results obtained by using the latest version of the algorithm (P2 version processor) on rectangular input matrices.** As in Figure 37, only the two input matrices and the synergistic output are reported in the image.

A new version of the processor function is currently being tested, exploiting the same strategies employed in the development of P2, but freeing itself from the indices of the data matrix it is to analyze. In fact, as can be read in Section 3.3.2, the P2 version of the processor function selects OLCI values to be assimilated to a TROPOMI value based on the indices at which they are placed within their respective matrices. In absolute terms, this is not a problem, since the two grids of data to be assimilated span the same area; simply, they have a different number of values within them. However, one can imagine that referring to the latitude and longitude values associated with the individual concentration values could make the job easier in case the assimilation process fits within a larger data manipulation sequence. To implement this idea, it was considered that the `DataArray`s subject to the assimilation process of our interest contain not only the array with the concentration values of the various PFTs, but also arrays of equal size representing the latitude and longitude coordinates of the pixel center point to which the concentration values refer.

Imagine we consider two `DataArray`s OLCI and TROPOMI that we wish to assimilate, and represent the information within them with the help of arrays of arrays with the following structure:

$$[[lat_{TROP}, lon_{TROP}, val_{TROP}]]$$
$$[[lat_{OLCI}, lon_{OLCI}, val_{OLCI}]]$$

where:

- lat_{TROP} and lon_{TROP} represent two generic values taken from the latitude and longitude arrays of the TROPOMI `DataArray` referenced to the coordinates of the center point of the generic pixel containing the generic concentration value val_{TROP}.
- lat_{OLCI}, lon_{OLCI} and val_{OLCI} are the equivalents of the values above but referred to the OLCI `DataArray`.

In this case, one could operate a loop over the array of arrays representing the TROPOMI `DataArray` and exploit a set of conditions like those shown in Section 3.3.2 to verify that the latitude and longitude extent of a single OLCI pixel are within the latitude and longitude extents of the TROPOMI pixel selected by the loop. Unfortunately, for the time being, the practical implementation of this idea in the processor function (called P3 in this version) is presenting some difficulties: P3 fails to select the right number of OLCI pixels corresponding to a given TROPOMI pixel, and this prevents the success of the entire interpolation process. Future work could be continued in this direction so that a working version of P3 could be developed.

This is not the only prospect of improvement for the algorithm, which could be enriched and worked on in several ways, some of which are suggested in the following paragraphs.

Input error handling and output error calculation

In the current version of the algorithm, input values are associated with automatically generated uncertainties within the `optimal_interpolation` function, however, arrays containing real data would arrive in the input along with arrays containing the errors associated with them. In order to be able to apply the algorithm on real data, it is necessary, in the future, to have the processor function take care of not only selecting the value of a certain TROPOMI pixel and those of certain OLCI pixels, but also of selecting the errors assigned to them, and then passing them to the `optimal_interpolation` function.

In addition, one aspect that has not yet been implemented in the code concerns the calculation of errors associated with the results: further efforts must be channeled in this direction in the near future.

Continuation of the testing phase and application on real data

At the current state of the work, the algorithm has been tested on both square and rectangular matrices, in both cases divided into square pixels, all "filled" with real numbers. The testing phase could continue through the creation of synthetic data that more closely reflects the features of real data: in fact, after gridding operations one often finds oneself with grids filled with rectangular pixels, some of which do not contain any real numbers, but NaN, i.e., non-numeric values.

Before continuing with the application of the algorithm to the actual data for the May-June 2018 period, it might be useful to test its performance on data that have a similar appearance to the latter, and then continue with gridding operations on the data for that bimonthly period. Once this phase is completed, we would proceed to choose a random day from the 61 available to test the algorithm's performance on the actual data for that date.

Should the outcome of this process be positive, the assimilation process would be applied to the data for each individual day of the time frame of interest. The results would be validated by comparison with values measured *in situ* during the same time frame.

Development of strategies for the validation of *in situ* measurements

Just as the correctness of the results of the algorithm must be determined by comparison with measurements conducted in the field, it would be interesting to develop strategies to validate measurements conducted *in situ* using the algorithm. To give a very simple example, imagine we want to measure the concentration of cyanobacteria in a lake from an initial estimate and two sets of *in situ* measurements: on the first day we use the algorithm on the estimate and the first set of observations, thus producing an estimate called A1, which is validated using the second set of observations. Assuming that the validation went well, new measurements are made the following day and A1 is merged with part of these, becoming A2, and being validated on the remaining observations. Again, the validation confirms the good performance of the algorithm. The following day, new observations are collected, with part of these and with the algorithm a new concentration estimate (A3) is created and validated on the remaining part of the observations. In this case, the operation does not go as hoped, because the set of measurements used for the validation operations deviates greatly from the A3 estimates. Should one necessarily assume that the algorithm does not work, or, since it worked the previous two times, can one assume that something went wrong during the collection of the observations used for A3 validation?

It would be interesting to develop strategies to propose such a hypothesis and test it, so that we get an instrument whose functioning is indeed verified through *in situ* measurements, but which, through its operation, also allows us to screen the quality of those measurements.

New mapping functions in optimal_interpolation

Currently, `optimal_interpolation` operates the data assimilation process through the use of a mapping function, $H(\boldsymbol{x}^b)$, that arithmetically averages over the values of $\boldsymbol{x}^b$; however, averaging is only one of many functions that can be employed for the purpose of mapping the values of $\boldsymbol{x}^b$ to the space of observations (y).

Over the course of this work, attempts were made to explore the possibility of using other functions in place of the mean; for example, a parameter was added to the `optimal_interpolation` function so that users could specify the mapping function of their interest, being able to choose between the arithmetic mean and the median. The idea of offering this choice arose after realizing that the median function might have better enabled a handling of outliers within $\boldsymbol{x}^b$. Unfortunately, the derivability characteristics of this function prevented keeping it as a viable choice for users of the algorithm: since the first derivative of the mapping function is essential to the process of optimal interpolation, and since the median function has an undefined derivative, we were forced to the sole use of the mean function.

The exclusion of the median function from the range of possible options for mapping functions does not prevent, but rather, spurs, the conduct of further research in this area: having implemented the possibility of choosing among different mapping functions, it will be possible to make comparisons between the results obtained by the optimal interpolation algorithm through the use of the different functions, and to estimate which ones are best suited for different situations.

Other methods of assimilation

Just as with the mapping functions, it would be interesting to implement in the algorithm the possibility for users to choose different methods of assimilating data. This would allow both to have a more versatile algorithm and to conduct comparisons between different methods of assimilating data from OLCI and TROPOMI.

Last March 11, a study was published (Reyes-Muñoz *et al.*, 2024) in which information from OLCI and TROPOMI is combined to build a model to estimate gross primary productivity (GPP) and net primary productivity (NPP) on the land surface. It would be interesting to further

investigate the methods of this study to see if the data assimilation strategies that were used could be implemented in the synergistic algorithm developed over the course of this thesis period. In this way, they could be applied to the dataset of our interest and their performance compared with the one of the optimal interpolation procedure by comparing the validations of the results performed on the *in situ* datasets held by the staff of the PHYTOOPTICS group.

In addition, optimal interpolation is a technique that has long been used in meteorology, now often replaced by data assimilation techniques such as 4DVar methods. In the future, these meteorological methods could also be investigated to evaluate whether they could be implemented in the algorithm and their performance compared with that of optimal interpolation.

Command line interface

With a view to making the algorithm more easily usable by people who did not contribute to its development, it might make sense to develop an interface that facilitates both the setting of the parameters the algorithm requires to function and its launch on a couple of datasets. An approach similar to what was used during the creation of the CLIs applied to Polymer and OC-PFT might be a good option. In a configuration file would be stowed the information necessary for the algorithm to work such as, for example:

- Type of assimilation desired.
- Type of mapping function desired (imagining the case in which more assimilation strategies are available, and it is chosen to proceed by optimal interpolation).
- Folders in which to retrieve datasets for assimilation.
- Folder in which to save the results of the assimilation process.
- Period of interest.

Once a configuration file with these characteristics has been created, the entire assimilation process would be initiated through a simple command line consisting of a command that, for convenience, could be called `synergy`, referring to a path leading to the configuration file mentioned above.

```
synergy   path-to-config_file
```

Kalman gain

The initial idea that was conceived regarding the data for the May-June 2018 period was to operate 61 interpolation processes: one per day. This would have provided a time series of 61 results on which to conduct studies to better understand some of the ecological characteristics of the PFTs of our interest. If, during the process of validating the results, it was to be seen that the concentration estimates for some days deviate greatly from the *in situ* measurements, it would be interesting to exploit the so-called Kalman gain, employing other data from the days of interest, to measure the improvement in the estimate that the algorithm proposes of PFT concentrations. In other words, the results of the interpolation process of the OLCI and TROPOMI data for May-June 2018 could in turn be interpolated with data from other satellites for the same period: the results would then be validated on the same *in situ* data on which the results of the first assimilation process were validated, so that the improvement in the estimates could be measured.

Further insight in the application of optimal interpolation for the interest of biologists

The system of equations describing the process of optimal interpolation has been applied to satellite data, but this does not exclude that it can also be employed to data of different types. After all, as reported in Chapter 2.3, what optimal interpolation allows you to do is to assimilate information from models with information from observations. It follows that any type of observation and any type of model can be employed in an optimal interpolation process.

One example might involve data from various microscopy techniques, such as atomic force microscopy (AFM), which, through the sliding and subsequent deformation of a probe over several bodies, measures their conformation at the nanoscale. Some studies have already demonstrated the advantage of assimilating data from molecular dynamics simulations with data from AFM microscopy: in the case of (Kato *et al.*, 2023) a Bayesian data assimilation method, known as sequential Monte Carlo method, was used, which is different than the optimal interpolation used to shape the algorithm that is the subject of this thesis work, however, of extreme interest both because it offers an additional perspective on how to assimilate data (to be considered for future developments of the algorithm), and because this perspective involves biological data of a different type than satellite data (to be considered for future applications of the algorithm).

The potential applications of a data assimilation algorithm to biological data do not end at the example given above. In fact, in order to use the algorithm developed during this thesis work, it is sufficient for the input data to be organized into arrays, which means that, for

example, some digital images from microscopy experiments could be used as input data. In fact, digital color images defined as “rasters” are nothing more than the union of three overlapping matrices containing the amounts of red, green, and blue corresponding to each pixel. Such matrices could be used as inputs to the data assimilation algorithm. Imagining having raster images portraying the same sample from two different microscopy instruments (which would therefore play a similar role as OLCI and TROPOMI), the synergistic product of them could be created. Or, to cite another very simple example, if one had an initial estimate of the distribution of certain molecules in a sample, one could create a more accurate estimate of that phenomenon through the assimilation of different types of measurements made on the same sample. In short, the possibilities of using a data assimilation algorithm in the biological field are innumerable.

In conclusion, the question raised during this thesis work was on how to assimilate data from the OLCI and TROPOMI instruments aboard the Sentinel-3 and Sentinel-5P satellites of the European Space Agency's Copernicus system, with a view to using the results of this assimilation process for improved estimation of the spatiotemporal distributions of two PFTs in the Atlantic Ocean. To meet this need, it was decided to use a data assimilation method called optimal intepolation, and to create an algorithm that would be able to apply it to two different input datasets. The structure of such an algorithm was defined and tested on synthetic data: it provided very encouraging first results.

Given that during the course of the work efforts were made to bring the OLCI data for the May-June 2018 period from level 1B to level 2A, and given that the TROPOMI level 2A data for the same period are already in the possession of the PHYTOOPTICS group staff, it is necessary in the immediate future to finalize the gridding procedures on both types of data, and, once the testing phase of the algorithm is completed, to put it to the test on any pair of gridded datasets from that period.

In the event that the outcome of this is positive, all gridded dataset pairs would be assimilated and the results validated using the set of *in situ* measurements conducted in the Atlantic Ocean during May-June 2018.

At the end of these operations, one could consider the idea of implementing in the algorithm both the possibility of conducting the optimal interpolation processes through the use of different mapping functions and, more generally, the possibility of conducting the assimilation

processes according to different methods, and, in case one decides to go down these paths, compare the performances of the different methods.

Finally, it is emphasized that the development of such an algorithm offers many perspectives both in terms of the analysis of satellite data and of the many interesting applications of data assimilation methods to data derived from the most widely used techniques in the biological field.

Appendix

Downloaded OLCI data

The data chosen for download were defined by the following quartet of variables:

- `contain = "%27OL_1_ERR%27"`, representing a string contained in all OLCI data at reduced resolution (in fact, RR is an acronym for "Reduced Resolution").
- `start_time = "2018-05-01T00:00:00.000Z"`, i.e., the initial time of the interest period, expressed according to the scheme `yyyy-mm-ddThh:mm.sssZ.`
- `end_time = "2018-06-30T23:59:59.000Z"`, that is, the final moment of the period of interest.
- `loc = f"POLYGON(({-75} {89}, {11} {89}, {11} {-90}, {-75} {-90}, {-75} {89}))"`, that is, the area the data should refer to. In this case it is a polygon (as the function `POLYGON` suggests), a rectangle, extended between the four points marked in parentheses, defined by their latitude and longitude coordinates: `{lat} {lon}`.

OLCI compressed files sorting function

`OLCI_zip_sorting`, the code in charge of ordering the downloaded OLCI files, is based on a function that requires four input data (the path to the folder in which the files are stowed in random order, the path to the folder in which it's wanted to stow them neatly, the start date of the period to be considered, and the end date of the same period), and the following logic:

1. Using the `datetime` library, the function transforms the start and end dates of the period under consideration into `datetime` objects with the form `"%Y-%m-%d"` and assigns them to two variables called `sdate` and `edate`.
2. With the help of the `os` library (Python Documentation, n.d.), which provides an interface to the operating system of the machine in use, the function produces a list of the files contained in the starting folder, the one in which the files are in no particular order.
3. It then creates a variable called `cdate`, that is used to operate a `while` loop in that list, and assign it the value of `sdate`, the starting date of the period under consideration: until `cdate` is equal to `edate`, the reference date of the last files we want, the loop continues.

4. During the cycle, the function checks which files are marked with a date equal to `cdate`, collects them in a list, and, again using `os`, checks if the destination folder contains a sequence of folders corresponding to the criteria imposed by `cdate`. In other words, if `cdate` refers to the day 2018/05/02, it is necessary that the destination folder contains the folder `2018`, which should contain the folder `05`, that in turn must contain the folder `02`. If this sequence of folders does not exist, the function provides for their creation via `os.makedirs` command.

Once it is determined that the saving path exists, the files stowed in the list just above are copied to it and a day is added to `cdate`, which it is checked to see if the `while` condition is still met. If yes, the loop iterates again, otherwise it aborts.

`new_polymer_submission` request creation and submission

The part of `new_polymer_submission` devoted to the creation and submission of the request files can be roughly summarized along the following points:

1. In order to create a `while` loop very similar to the one in `OLCI_zip_sorting`, three variables are initialized, `sdate`, `cdate`, and `edate`, which represent, respectively, the start date of the period of interest, the date that is used during the loop, and the end date of the period of interest. The cycle begins by assigning `cdate` the same value as `sdate` and continues until the value of `cdate` is equal to that of `edate`.
2. During the cycle, using the `os` library introduced above, `new_polymer_submission` moves to the directory in which it is desired to store the configuration files for `polymer_run`, and checks for the existence of a configuration file related to the date indicated by `cdate`. If the file does not exist, `new_polymer_submission` proceeds to create it using the `subprocess` library (Python Documentation, n.d.) and the functionality of the command line `polymer_run`.
3. Once the configuration file has been created, the information within it must be modified to ensure that Polymer is applied to the correct data set. To do this, `new_polymer_submission` makes use of the `configparser` library (Python Documentation, n.d.), which provides access to different parts of a configuration file via the simple structure:

```
configparser_object["section"]["key"] = value
```

In fact, as stated in Python's documentation, a typical configuration file is organized into sections and keys: in the configuration file example below, Figure 28, an example of a section is represented by `[DEFAULT]`, while an example of a key is `Compression`.

```
[DEFAULT]
ServerAliveInterval = 45
Compression = yes
CompressionLevel = 9
ForwardX11 = yes

[forge.example]
User = hg

[topsecret.server.example]
Port = 50022
ForwardX11 = no
```

Figure 28 | **Example of a simple configuration file.** (Python Documentation, n.d.)

The `configparser` library allows nimbly editing of the values characterizing a configuration file by turning the file itself into a `configparser` object, and using the structure shown above. For example, imagine that the configuration file shown above was transformed in a `configparser` object named `A`, and that it was desired to modify the value of the key `Compression` from `yes` to `no`: it would be sufficient to use a structure like

```
A["DEFAULT"]["Compression"] = no
```

4. After customizing the configuration file, `new_polymer_submission` moves to the folder designated to contain the Bash scripts that will be used as an interface to Albedo's request handling system. Here, it creates a Bash script with all the necessary context information (account of the user submitting the request, time to work, request name, etc...) to which it adds two lines of code:

```
source activate venv

polymer_run path-to-config_file
```

where

a. `source activate venv` is a command that is used to activate the virtual environment in which Polymer and the `polymer_run` command line are installed.
b. `polymer_run path-to-config_file` is the command used to apply Polymer to the data set specified in the configuration file created in step 2 and customized in step 3.

5. Through the `subprocess` library, Albedo is ordered to execute the requests marked in the `.sh`[3] script just created.

This cycle is repeated for all dates between `sdate` and `edate`, resulting in an equal number of requests being submitted to the HPC, which it will conduct in parallel.

[3] Bash files are characterized by `.sh` extension.

References

Bibliography

Aiken, J., Pradhan, Y., Barlow, R., Lavender, S., Poulton, A., Holligan, P., & Hardman-Mountford, N. (2009). Phytoplankton pigments and functional types in the Atlantic Ocean: A decadal assessment, 1995–2005. *Deep Sea Research Part II: Topical Studies in Oceanography*, *56*(15), 899–917. https://doi.org/10.1016/j.dsr2.2008.09.017

Barlow, R., Stuart, V., Lutz, V., Sessions, H., Sathyendranath, S., Platt, T., Kyewalyanga, M., Clementson, L., Fukasawa, M., Watanabe, S., & Devred, E. (2007). Seasonal pigment patterns of surface phytoplankton in the subtropical southern hemisphere. *Deep Sea Research Part I: Oceanographic Research Papers*, *54*(10), 1687–1703. https://doi.org/10.1016/j.dsr.2007.06.010

Basu, S., & Mackey, K. R. M. (2018). Phytoplankton as Key Mediators of the Biological Carbon Pump: Their Responses to a Changing Climate. *Sustainability*, *10*(3), Article 3. https://doi.org/10.3390/su10030869

Bonomo, M., Zucol, A. F., Gutiérrez Téllez, B., Coradeghini, A., & Vigna, M. S. (2009). Late Holocene palaeoenvironments of the Nutria Mansa 1 archaeological site, Argentina. *Journal of Paleolimnology*, *41*(2), 273–296. https://doi.org/10.1007/s10933-008-9225-3

Bracher, A., Brewin, R. J. W., Ciotti, A. M., Clementson, L. A., Hirata, T., Kostadinov, T. S., Mouw, C. B., & Organelli, E. (2022). Chapter 7—Applications of satellite remote sensing technology to the analysis of phytoplankton community structure on large scales. In L. A. Clementson, R. S. Eriksen, & A. Willis (Eds.), *Advances in Phytoplankton Ecology* (pp. 217–244). Elsevier. https://doi.org/10.1016/B978-0-12-822861-6.00015-7

Bracher, A., Vountas, M., Dinter, T., Burrows, J. P., Rottgers, R., & Peeken, I. (2009). *Quantitative observation of cyanobacteria and diatoms from space using PhytoDOAS on SCIAMACHY data*.

Budge, S. M., Devred, E., Forget, M.-H., Stuart, V., Trzcinski, M. K., Sathyendranath, S., & Platt, T. (2014). Estimating concentrations of essential omega-3 fatty acids in the ocean:

Supply and demand. *ICES Journal of Marine Science*, *71*(7), 1885–1893. https://doi.org/10.1093/icesjms/fsu003

Claustre, H., Legendre, L., Boyd, P. W., & Levy, M. (2021). The Oceans' Biological Carbon Pumps: Framework for a Research Observational Community Approach. *Frontiers in Marine Science*, *8*. https://www.frontiersin.org/articles/10.3389/fmars.2021.780052

Damatac, A. M., & Cao, E. P. (2022). Identification and diversity assessment of cyanobacterial communities from some mine tailing sites in Benguet Province, Philippines using isolation-dependent and isolation-independent methods. *Environment, Development and Sustainability*, *24*(1), 1166–1187. https://doi.org/10.1007/s10668-021-01489-8

Dutkiewicz, S., Cermeno, P., Jahn, O., Follows, M. J., Hickman, A. E., Taniguchi, D. A. A., & Ward, B. A. (2020). Dimensions of marine phytoplankton diversity. *Biogeosciences*, *17*(3), 609–634. https://doi.org/10.5194/bg-17-609-2020

Fussell, J., & Rundquist, D. (1986). On Defining Remote Sensing. *PHOTOGRAMMETRIC ENGINEERING*.

Glibert, P. M., & Mitra, A. (2022). From webs, loops, shunts, and pumps to microbial multitasking: Evolving concepts of marine microbial ecology, the mixoplankton paradigm, and implications for a future ocean. *Limnology and Oceanography*, *67*(3), 585–597. https://doi.org/10.1002/lno.12018

Gold, V. (Ed.). (2019). *The IUPAC Compendium of Chemical Terminology: The Gold Book* (4th ed.). International Union of Pure and Applied Chemistry (IUPAC). https://doi.org/10.1351/goldbook

Golub, G. H., & Van Loan, C. F. (2013). *Matrix computations* (Fourth edition). The Johns Hopkins University Press.

Grattan, L. M., Holobaugh, S., & Morris, J. G. (2016). Harmful Algal Blooms and Public Health. *Harmful Algae*, *57*(B), 2–8. https://doi.org/10.1016/j.hal.2016.05.003

Guanter, L., Aben, I., Tol, P., Krijger, J. M., Hollstein, A., Köhler, P., Damm, A., Joiner, J., Frankenberg, C., & Landgraf, J. (2015). Potential of the TROPOspheric Monitoring Instrument (TROPOMI) onboard the Sentinel-5 Precursor for the monitoring of terrestrial

chlorophyll fluorescence. *Atmospheric Measurement Techniques*, *8*(3), 1337–1352. https://doi.org/10.5194/amt-8-1337-2015

Hallegraeff, G. M., Anderson, D. M., Belin, C., Bottein, M.-Y. D., Bresnan, E., Chinain, M., Enevoldsen, H., Iwataki, M., Karlson, B., McKenzie, C. H., Sunesen, I., Pitcher, G. C., Provoost, P., Richardson, A., Schweibold, L., Tester, P. A., Trainer, V. L., Yñiguez, A. T., & Zingone, A. (2021). Perceived global increase in algal blooms is attributable to intensified monitoring and emerging bloom impacts. *Communications Earth & Environment*, *2*(1), 117. https://doi.org/10.1038/s43247-021-00178-8

Hirata, T., Hardman-Mountford, N. J., Brewin, R. J. W., Aiken, J., Barlow, R., Suzuki, K., Isada, T., Howell, E., Hashioka, T., Noguchi-Aita, M., & Yamanaka, Y. (2011). Synoptic relationships between surface Chlorophyll-*a* and diagnostic pigments specific to phytoplankton functional types. *Biogeosciences*, *8*(2), 311–327. https://doi.org/10.5194/bg-8-311-2011

IOCCG (2014). Phytoplankton Functional Types from Space. Sathyendranath, S. (ed.), Reports of the International Ocean-Colour Coordinating Group, No. 15, IOCCG, Dartmouth, Canada

Irwin, A. J., & Finkel, Z. V. (2017). *Phytoplankton functional types: A trait perspective* (p. 148312). bioRxiv. https://doi.org/10.1101/148312

Isada, T., Kuwata, A., Saito, H., Ono, T., Ishii, M., Yoshikawa-Inoue, H., & Suzuki, K. (2009). Photosynthetic features and primary productivity of phytoplankton in the Oyashio and Kuroshio-Oyashio transition regions of the northwest Pacific. *Journal of Plankton Research*, *31*(9), 1009–1025. https://doi.org/10.1093/plankt/fbp050

Jin, Z., Ainsworth, E. A., Leakey, A. D. B., & Lobell, D. B. (2018). Increasing drought and diminishing benefits of elevated carbon dioxide for soybean yields across the US Midwest. *Global Change Biology*, *24*(2), e522–e533. https://doi.org/10.1111/gcb.13946

Kato, S., Takada, S., & Fuchigami, S. (2023). Particle Smoother to Assimilate Asynchronous Movie Data of High-Speed AFM with MD Simulations. *Journal of Chemical Theory and Computation*, *19*(14), 4678–4688. https://doi.org/10.1021/acs.jctc.2c01268

Kharbush, J. J., Close, H. G., Van Mooy, B. A. S., Arnosti, C., Smittenberg, R. H., Le Moigne, F. A. C., Mollenhauer, G., Scholz-Böttcher, B., Obreht, I., Koch, B. P., Becker, K. W., Iversen, M. H., & Mohr, W. (2020). Particulate Organic Carbon Deconstructed: Molecular and Chemical Composition of Particulate Organic Carbon in the Ocean. *Frontiers in Marine Science*, *7*. https://doi.org/10.3389/fmars.2020.00518

Lacey, T. (n.d.). *Tutorial: The Kalman Filter*.

Lahoz, W. A., & Schneider, P. (2014). Data assimilation: Making sense of Earth Observation. *Frontiers in Environmental Science*, *2*. https://www.frontiersin.org/articles/10.3389/fenvs.2014.00016

Le Moigne, F. A. C. (2019). Pathways of Organic Carbon Downward Transport by the Oceanic Biological Carbon Pump. *Frontiers in Marine Science*, *6*, 634. https://doi.org/10.3389/fmars.2019.00634

Losa, S. N., Soppa, M. A., Dinter, T., Wolanin, A., Brewin, R. J. W., Bricaud, A., Oelker, J., Peeken, I., Gentili, B., Rozanov, V., & Bracher, A. (2017). Synergistic Exploitation of Hyper- and Multi-Spectral Precursor Sentinel Measurements to Determine Phytoplankton Functional Types (SynSenPFT). *Frontiers in Marine Science*, *4*. https://www.frontiersin.org/articles/10.3389/fmars.2017.00203

Mansur, L., Plaza, G., Landaeta, M. F., & Ojeda, F. P. (2014). Planktonic duration in fourteen species of intertidal rocky fishes from the south-eastern Pacific Ocean. *Marine and Freshwater Research*, *65*(10), 901–909. https://doi.org/10.1071/MF13064

Nature, 350, 669–674. https://doi.org/10.1038/350669a0

Neftel, A., Moor, E., Oeschger, H., & Stauffer, B. (1985). Evidence from polar ice cores for the increase in atmospheric CO2 in the past two centuries. *Nature*, *315*(6014), 45–47. https://doi.org/10.1038/315045a0

Nowicki, M., DeVries, T., & Siegel, D. A. (2022). Quantifying the Carbon Export and Sequestration Pathways of the Ocean's Biological Carbon Pump. *Global Biogeochemical Cycles*, *36*(3), e2021GB007083. https://doi.org/10.1029/2021GB007083

O'Reilly, J. E., Maritorena, S., Mitchell, B. G., Siegel, D. A., Carder, K. L., Garver, S. A., Kahru, M., & McClain, C. (1998). Ocean color chlorophyll algorithms for SeaWiFS. *Journal of Geophysical Research: Oceans*, *103*(C11), 24937–24953. https://doi.org/10.1029/98JC02160

Peter, S., Chopra, S., & Jacob, J. (2013). A fish a day, keeps the cardiologist away! - A review of the effect of omega-3 fatty acids in the cardiovascular system. *Indian Journal of Endocrinology and Metabolism*, *17*(3), 422. https://doi.org/10.4103/2230-8210.111630

Pimm, S., Lawton, J. & Cohen, J. (1991). Food web patterns and their consequences.

Rew, R., Davis, G., Emmerson, S., Cormack, C., Caron, J., Pincus, R., Hartnett, E., Heimbigner, D., Appel, L., & Fisher, W. (1989). *Unidata NetCDF* [Application/java-archive,application/gzip,application/tar]. [object Object]. https://doi.org/10.5065/D6H70CW6

Reyes-Muñoz, P., D.Kovács, D., Berger, K., Pipia, L., Belda, S., Rivera-Caicedo, J. P., & Verrelst, J. (2024). Inferring global terrestrial carbon fluxes from the synergy of Sentinel 3 & 5P with Gaussian process hybrid models. *Remote Sensing of Environment*, *305*, 114072. https://doi.org/10.1016/j.rse.2024.114072

Salmaso, N., Naselli-Flores, L., & Padisák, J. (2015). Functional classifications and their application in phytoplankton ecology. *Freshwater Biology*, *60*(4), 603–619. https://doi.org/10.1111/fwb.12520

Sanders, R., Henson, S. A., Koski, M., De La Rocha, C. L., Painter, S. C., Poulton, A. J., Riley, J., Salihoglu, B., Visser, A., Yool, A., Bellerby, R., & Martin, A. P. (2014). The Biological Carbon Pump in the North Atlantic. *Progress in Oceanography*, *129*, 200–218. https://doi.org/10.1016/j.pocean.2014.05.005

Sigman, D. M., & Hain, M. P. (2012). *The Biological Productivity of the Ocean*. *3*(6).

Siriwardhana, N., Kalupahana, N. S., & Moustaid-Moussa, N. (2012). Chapter 13 - Health Benefits of n-3 Polyunsaturated Fatty Acids: Eicosapentaenoic Acid and Docosahexaenoic Acid. In S.-K. Kim (Ed.), *Advances in Food and Nutrition Research*

(Vol. 65, pp. 211–222). Academic Press. https://doi.org/10.1016/B978-0-12-416003-3.00013-5

Soppa, M. A., Hirata, T., Silva, B., Dinter, T., Peeken, I., Wiegmann, S., & Bracher, A. (2014). Global Retrieval of Diatom Abundance Based on Phytoplankton Pigments and Satellite Data. *Remote Sensing*, *6*(10), Article 10. https://doi.org/10.3390/rs61010089

Soppa, M. A., Silva, B., Steinmetz, F., Keith, D., Scheffler, D., Bohn, N., & Bracher, A. (2021). Assessment of Polymer Atmospheric Correction Algorithm for Hyperspectral Remote Sensing Imagery over Coastal Waters. *Sensors*, *21*(12), 4125. https://doi.org/10.3390/s21124125

Styan, G. P. H. (1973). Hadamard products and multivariate statistical analysis. *Linear Algebra and Its Applications*, *6*, 217–240. https://doi.org/10.1016/0024-3795(73)90023-2

Sultan, B., Parkes, B., & Gaetani, M. (2019). Direct and indirect effects of CO2 increase on crop yield in West Africa. *International Journal of Climatology*, *39*(4), 2400–2411. https://doi.org/10.1002/joc.5960

Supit, I., Van Diepen, C. A., De Wit, A. J. W., Wolf, J., Kabat, P., Baruth, B., & Ludwig, F. (2012). Assessing climate change effects on European crop yields using the Crop Growth Monitoring System and a weather generator. *Agricultural and Forest Meteorology*, *164*, 96–111. https://doi.org/10.1016/j.agrformet.2012.05.005

Suzuki, K., Hinuma, A., Saito, H., Kiyosawa, H., Liu, H., Saino, T., & Tsuda, A. (2005). Responses of phytoplankton and heterotrophic bacteria in the northwest subarctic Pacific to *in situ* iron fertilization as estimated by HPLC pigment analysis and flow cytometry. *Progress in Oceanography*, *64*(2–4), 167–187. https://doi.org/10.1016/j.pocean.2005.02.007

Uitz, J., Claustre, H., Morel, A., & Hooker, S. B. (2006). Vertical distribution of phytoplankton communities in open ocean: An assessment based on surface chlorophyll. *Journal of Geophysical Research: Oceans*, *111*(C8), 2005JC003207. https://doi.org/10.1029/2005JC003207

Vidussi, F., Claustre, H., Manca, B. B., Luchetta, A., & Marty, J. (2001). Phytoplankton pigment distribution in relation to upper thermocline circulation in the eastern Mediterranean Sea during winter. *Journal of Geophysical Research: Oceans*, *106*(C9), 19939–19956. https://doi.org/10.1029/1999JC000308

Volk, T., & Hoffert, M. I. (1985). Ocean Carbon Pumps: Analysis of Relative Strengths and Efficiencies in Ocean-Driven Atmospheric CO2 Changes. In *The Carbon Cycle and Atmospheric CO2: Natural Variations Archean to Present* (pp. 99–110). American Geophysical Union (AGU). https://doi.org/10.1029/GM032p0099

Vountas, M., Dinter, T., Bracher, A., Burrows, J. P., & Sierk, B. (2007). Spectral studies of ocean water with space-borne sensor SCIAMACHY using Differential Optical Absorption Spectroscopy (DOAS). *Ocean Science*, *3*(3), 429–440. https://doi.org/10.5194/os-3-429-2007

Werdell, P. J., & Bailey, S. W. (2005). An improved in-situ bio-optical data set for ocean color algorithm development and satellite data product validation. *Remote Sensing of Environment*, *98*(1), 122–140. https://doi.org/10.1016/j.rse.2005.07.001

Zhong, W., & Haigh, J. D. (2013). The greenhouse effect and carbon dioxide. *Weather*, *68*(4), 100–105. https://doi.org/10.1002/wea.2072

Sitography

configparser—Configuration file parser. (n.d.). Python Documentation. Retrieved 23 March 2024, from https://docs.python.org/3/library/configparser.html

Food Web: Concept and Applications | Learn Science at Scitable. (n.d.). Retrieved 12 March 2024, from https://www.nature.com/scitable/knowledge/library/food-web-concept-and-applications-84077181/

Harmful Algal Bloom Programme | Intergovernmental Oceanographic Commission. (n.d.). Retrieved 19 March 2024, from https://www.ioc.unesco.org/en/harmful-algal-bloom-programme

HYGEOS - polymer. (n.d.). Retrieved 22 March 2024, from https://www.hygeos.com/polymer

ICYMARE 2023 OLDENBURG RETROSPECTIVE - ICYMARE. (n.d.). Retrieved 24 March 2024, from https://www.icymare.com/about-the-conference/past-conferences/icymare-2023-oldenburg/icymare-2023-oldenburg-retrospective/

Interpolation (scipy.interpolate)—SciPy v1.12.0 Manual. (n.d.). Retrieved 27 March 2024, from https://docs.scipy.org/doc/scipy/reference/interpolate.html

Introduction—Cartopy 0.22.0 documentation. (n.d.). Retrieved 25 March 2024, from https://scitools.org.uk/cartopy/docs/latest/

Matplotlib—Visualization with Python. (n.d.). Retrieved 25 March 2024, from https://matplotlib.org/

Numba: A High Performance Python Compiler. (n.d.). Retrieved 25 March 2024, from https://numba.pydata.org/

numpy.ndarray—NumPy v1.26 Manual. (n.d.). Retrieved 25 March 2024, from https://numpy.org/doc/stable/reference/generated/numpy.ndarray.html

Ocean Optics—AWI. (n.d.). Retrieved 21 March 2024, from https://www.awi.de/en/science/climate-sciences/physical-oceanography/main-research-focus/ocean-optics.html

OLCI Instrument – Sentinel-3 OLCI Technical Guide – Sentinel Online. (n.d.). Sentinel Online. Retrieved 20 March 2024, from https://copernicus.eu/technical-guides/sentinel-3-olci/olci-instrument

os—Miscellaneous operating system interfaces. (n.d.). Python Documentation. Retrieved 22 March 2024, from https://docs.python.org/3/library/os.html

Sentinel-3 | EUMETSAT. (2020, May 26). https://www.eumetsat.int/sentinel-3

Sentinel-5P TROPOMI User Guide—Sentinel Online. (n.d.). Sentinel Online. Retrieved 26 March 2024, from https://copernicus.eu/user-guides/sentinel-5p-tropomi

Sentinel-5P. (n.d.). Retrieved 20 March 2024, from https://www.esa.int/ESA_Multimedia/Videos/2017/07/Sentinel-5P/%28lang%29

Stations—AWI. (n.d.). Retrieved 21 March 2024, from https://www.awi.de/en/expedition/stations.html

subprocess—Subprocess management. (n.d.). Python Documentation. Retrieved 23 March 2024, from https://docs.python.org/3/library/subprocess.html

US Department of Commerce, N. (n.d.). *Global Monitoring Laboratory—Carbon Cycle Greenhouse Gases*. Retrieved 16 March 2024, from https://gml.noaa.gov/ccgg/trends/

User Guides—Sentinel-3 OLCI - Naming Convention—Sentinel Online. (n.d.). Sentinel Online. Retrieved 22 March 2024, from https://copernicus.eu/user-guides/sentinel-3-olci/naming-convention

What is an HPC cluster | High Performance Computing. (n.d.). Retrieved 23 March 2024, from https://www.hpc.iastate.edu/guides/introduction-to-hpc-clusters/what-is-an-hpc-cluster

Xarray.`DataArray`. (n.d.). Xarray. Retrieved 25 March 2024, from https://docs.xarray.dev/en/latest/generated/xarray.`DataArray`.html