Comparison of Operational and Scientific Atmospheric Carbon Monoxide Data Products from the Sentinel-5 Precursor Satellite

A thesis submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE
in EARTH AND ENVIRONMENTAL SCIENCES

Ann-Kathrin A. Krutsch

Jacobs University
Department of Physics and Earth Sciences
Spring 2020

Supervisors:
Dr. Michael Buchwitz (Universität Bremen, Institut für Umweltphysik)
Prof. Dr. Joachim Vogt (Jacobs University, Department of Physics and Earth Sciences)
Dr. Oliver Schneising (Universität Bremen, Institut für Umweltphysik)

Abstract

Carbon monoxide is a trace gas that has significant impacts on human health, air pollution and the global climate. Although it is not considered a greenhouse gas, it is able to alter the concentrations of other greenhouse gases, wherefore it is referred to as an indirect greenhouse gas.

There are currently two algorithms used to generate global carbon monoxide data products from the TROPOMI instrument onboard the Sentinel-5 Precursor satellite, namely the operational Copernicus program SICOR algorithm and the scientific WFM-DOAS algorithm developed at the Institute of Environmental Physics at the University of Bremen.

The objective of this thesis is to compare these two data products. Global comparisons have been carried out for daily data, in order to quantify the systematic differences. To achieve this, different aspects of the retrievals are considered, in particular their quality flags and their spatial coverage. Global maps of the retrievals are generated and analyzed, as well as maps of absolute and relative differences, and the latitudinal distribution. The level of agreement or disagreement has been quantified by computing mean differences, standard deviations of differences, and their linear correlation.

For the three investigated days the mean differences are very small (around 1%), the standard deviation of the differences is below 10% and the linear correlation coefficient is about 0.97, indicating that the two data products agree very well. However, as shown by spatial maps and latitudinal difference plots, differences can be larger during certain times and at certain latitudes. These comparison results have been obtained after collocating the observations in order to be able to compute the difference for individual ground pixels, as the two data products show differences in their spatial coverage. This is due to the operational product aiming to also provide retrievals for partially cloudy scenes, whereas the scientific product limits its retrievals to cloud-free scenes. Therefore, the number of operational retrievals is roughly eight times larger than the amount of retrievals of the scientific data product.

Acknowledgements

Being one of many graduates of the Class of 2020, writing my thesis under the extraordinary circumstances faced due to COVID-19, this bachelor thesis would not have been possible without the support of my supervisors, professors, family and friends. These were the people who helped me through all the difficulties and tough moments I faced throughout this significant period of time. It was the help of these valuable individuals that made it possible for me to write this thesis.

First, I would like to thank my main supervisor, Dr. Michael Buchwitz, for his support from the first to the last step and for his precious time to answer my questions and concerns.

Second, I want to thank my second supervisor and academic advisor, Prof. Dr. Joachim Vogt, for his incredible support, for always being available to help me overcome the difficulties I was facing, for his kind and supporting words when I needed them and for all the time and effort he contributed to this thesis.

Third, I want to thank Dr. Oliver Schneising for guiding me through the programming needed for this thesis, for helping me where I needed his support and for proofreading my work.

I wish to thank my parents, Rainer and Ursula Krutsch, for giving me the opportunity to study at Jacobs University and for their daily patience and efforts to support me the best they could.

I wish to thank Nele Stachlys, Tim Jährig and Carlo Fulde for the special support they gave me throughout the entire process, for always listening to my concerns, for their supportive words and the confidence they gave me. In particular I want to thank Nele Stachlys and Carlo Fulde for proofreading this thesis, and Tim Jährig and Nele Stachlys for guiding me through any difficulties I faced using Latex.

The TROPOMI/SICOR data files were obtained from the Institute of Environmental Physics at the University of Bremen, and given to me by Dr. Oliver Schneising.

The TROPOMI/WFMD data files are available at *https://www.iup.uni-bremen.de/carbon_ghg/products/tropomi_wfmd/*. Dr. Oliver Schneising has prepared a special data set for me to use, that includes the carbon monoxide column values, in order to facilitate the comparison done in this thesis.

Contents

1 Introduction

Carbon monoxide is an atmospheric trace gas that influences many fields and aspects, ranging from everyday human life to global environmental impacts.

This thesis focuses on two products of satellite data that aim to monitor carbon monoxide levels at a global, regional and local scale, to improve our understanding of this gas, its spread, and its effects. To better understand the significance of carbon monoxide, this introduction will outline some of its impacts, such as that on human health and the global climate, and will proceed to look at its different sources. Afterwards, the introduction will elaborate on some of the chemical properties of carbon monoxide, its environmental impact and its role in the atmosphere. As this thesis focuses on satellite measurements, an explanation of why this is believed to be a very good method for global carbon monoxide observations to be made is included. This introductory section will then be concluded with the aim for this thesis, the results of which are described in this document.

1.1 Overview of Atmospheric Carbon Monoxide

Carbon monoxide is a tasteless, odorless, colorless, non-corrosive, and quite stable diatomic molecule, which is found in the Earth's atmosphere in a gaseous state. Carbon monoxide absorbs radiation in the infrared region of the electromagnetic spectrum, while it doesn't absorb visible light and near ultra-violet radiation. In addition, it has some weak absorption bands between 125 and 155 nm. Furthermore, carbon monoxide has a low electric dipole moment of 0.10 debye, a short interatomic distance of 0.123 nm and a high heat of formation (2 072 kJ/mol) [Raub et al., 1999]. It has an average atmospheric lifetime of about 1 to 2 months, which makes it a good tracer to monitor the long-rage transport of pollution [Schneising et al., 2019].

Among the many properties of carbon monoxide, its chemical reactions can produce very significant amounts of ozone (O_3) in the Earth's troposphere. This means that an increase in carbon monoxide leads to an increase in ozone. In order to stay in balance, the hydroxyl radical (OH) is then depleted, affecting the abundances of many other trace gases, both natural and anthropogenic, which are removed from the atmosphere by reacting with the hydroxyl radical. Trace gas concentrations vary under two conditions: either with a variation in loss rate or with a change or cyclical variability in emission rate. Carbon monoxide is affected by both of these processes [Khalil and Rasmussen, 1990].

The direct forcing of carbon monoxide is often considered negligible [Sinha and Tuomi, 1996]. The direct effects can only be observed at high concentrations and large scales, to the extent that even an increase in direct forcing would only be of little significance [Holloway et al., 2000]. Nevertheless, carbon monoxide is a very important trace gas. It is a major constituent affecting tropospheric hydroxyl abundances and therefore also to the oxidizing capacity of the lower atmosphere. Overall, the trends and changes in hydroxyl emissions are highly complicated, non-linear, and therefore hard to foresee [Daniel and Solomon, 1998].

Carbon monoxide also has indirect forcing components, which are of higher significance than its direct climate forcing. One indirect forcing component is the production of tropospheric ozone. However, the dimension of ozone loss or production through carbon monoxide strongly depends on the abundance of nitrogen oxides (NO_X), which is highly uncertain and subject to variations [Daniel and Solomon, 1998].

Sources of Carbon Monoxide

Global and local carbon monoxide levels vary in seasonal cycles, influenced by both loss and emission rates. In the atmosphere carbon monoxide is produced by reactions of hydroxyl radicals with methane (CH_4) and other hydrocarbons of both natural and anthropogenic origin, as well as alkenes reacting with ozone. Carbon monoxide is removed from the atmosphere, by reacting with tropospheric hydroxyl radicals [Apituley et al., 2018]. The largest source of carbon monoxide in the atmosphere are combustion processes and the oxidation of hydrocarbons. However, average levels of hydroxyl radicals are decreasing, wherefore the production of carbon monoxide from hydrocarbons is receding. Nevertheless, the process of hydrocarbon oxidation is the most important natural chemical source and the second largest global source of carbon monoxide [Khlystova, 2010]. In addition, highest significance is given to the carbon monoxide production by combustion of carbonaceous fuels, such as oil, gasoline, coal and wood, as these generate two main outcomes: carbon monoxide and carbon dioxide. A remarkably large amount of carbon monoxide is produced, when there is a high amount of fuel and a limited amount of air and oxygen available for the combustion [Raub et al., 1999; Levy, 2015; Khalil and Rasmussen, 1990].

According to Raub et al. [1999], human activities constitute approximately 60% of tropospheric carbon monoxide in non-urban areas and environments, and the remaining 40% of emissions come from natural sources. 45% of the annual carbon monoxide emissions are directly produced by combustion processes, and the remaining 55% are mainly from the oxidation of hydrocarbons and from other sources such as plants and the ocean. The majority of carbon monoxide produced directly through combustion comes from the burning of fossil fuels (19% of carbon monoxide emissions, corresponding to 500 million tonnes of carbon monoxide) and forest clearing (15% or 400 million tonnes yearly).

The oxidation of methane and other hydrocarbons is often referred to as a natural source of carbon monoxide, while direct carbon monoxide emissions from fossil fuel combustion processes are considered to be anthropogenic. However, some hydrocarbons that oxidize to become carbon monoxide are produced by these same combustion processes, and are still considered an indirect natural source. Approximately half of the methane in the Earth's atmosphere is anthropogenic, originating from agriculture and urban activities. Therefore, about 50% of the carbon monoxide that is thought to come from the oxidation of methane could be considered to be anthropogenic as well. Furthermore, the total emissions of carbon monoxide are highly dependent on the abundances of other trace gases, wherefore it is very difficult and complex to estimate all carbon monoxide sources [Khalil and Rasmussen, 1990].

As described by Raub et al. [1999], anthropogenic carbon monoxide emissions sources can be categorized into five groups: transportation sources, stationary combustion equipment, industrial processes, solid waste carbon and miscellaneous carbon monoxide emissions. Transportation sources include all kinds of motor-vehicles running on combustion engines, such as cars, trucks, buses, motorcycles, airplanes, farm equipment engines, lawnmowers and snowmobiles. Stationary combustion equipment refers to heating- and power-plants fueled with coal, gas and oil, as they produce carbon monoxide through improper and inefficient operating practices or combustion techniques. Additionally to fuel combustion, there is a variety of other industrial processes generating and emitting different quantities of carbon monoxide. Solid waste carbon refers to the emissions from domestic and municipal refuse, while miscellaneous carbon monoxide emissions refers to the emissions resulting from the burning of agricultural and forest materials and some other minor sources. According to Holloway et al. [2000], carbon monoxide from biomass burning and from biogenic carbon oxidation each have a contribution of more than 15% to the total carbon monoxide concentrations globally.

Seasonal changes in atmospheric carbon monoxide levels are mostly controlled by the emissions, transport and combustion of single sources, while methane oxidation is not always considered when analyzing the global distribution and seasonal changes of carbon monoxide emissions. Due to the long lifetime methane has in the atmosphere, its oxidation generates a relatively uniform background concentration on the global carbon monoxide distributions. In the northern hemisphere, seasonal dominance in emission comes from fossil fuels, while in the tropics biomass burning and the oxidation of certain hydrocarbons such as isoprene dominate [Holloway et al., 2000; Apituley et al., 2018].

According to Raub et al. [1999], the latitudinal distribution of carbon monoxide sources can be summarized in a one-dimensional model. Especially in the middle and northern latitudes, emissions are significantly higher in spring and summer months for three main reasons. First, the oxidation of hydrocarbons (including methane) is notably faster during summer due to the seasonal variations in the abundance of hydroxyl radicals. Second, many of the direct emission sources of carbon monoxide increase in the spring and summer time. At last, at mid and high latitudes, hydrocarbons build up during the winter time and start to oxidize more with the large increase in hydroxyl radicals in spring. Overall, carbon monoxide sources can be estimated if sinks, transport and concentrations are known.

Figure 1 shows carbon monoxide mixing ratios, denoted as CO, averaged from November 13^{th} to 19^{th} 2017 in [ppb] by Borsdorff et al. [2018a]. The increased carbon monoxide concentrations at mid and mid-northern latitudes as mentioned by Raub et al. [1999] can also be observed. Areas of significantly high concentrations can be seen along the Chinese coast representing its industrial area, in India especially in the north, in central western Africa, and in central South America in Brazil. Furthermore, some smaller hot spots can be seen on the eastern coast of southern Africa, on the western coast of Madagascar, and in northwestern Australia. Extremely low carbon monoxide levels can be observed in the Himalaya and Tibetan Plateau area, and on the western coast of the United States.

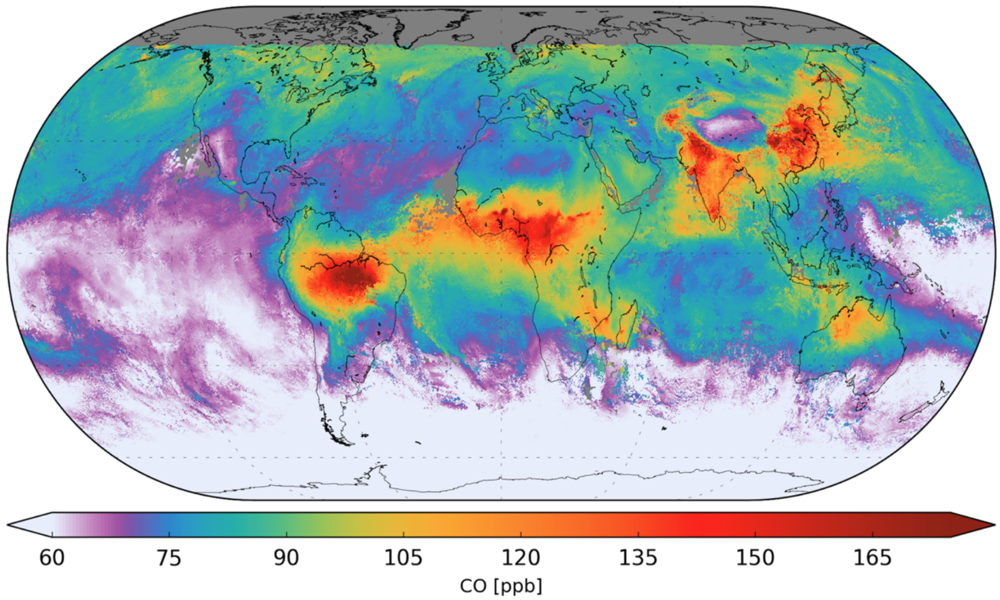


Figure 1: Carbon monoxide mixing ratios (denoted as CO) from Copernicus Atmosphere Monitoring Service, averaged from November 13th to November 19th, 2017, in [ppb] (parts per billion). (Figure 1, Borsdorff et al. [2018a]).

Health Impact of Carbon Monoxide

Carbon monoxide is a determinant of air quality, that can be of great danger to humans. The exposure to higher concentrations has a direct effect on human health, because it can hinder the transport of oxygen by hemoglobin in red blood cells [Levy, 2015; Holloway et al., 2000].

According to Goldstein [2009], carbon monoxide is a very infamous poison that silently takes human lives. In the United States, unintentional and non-fire-related exposures to enhanced carbon monoxide concentrations every year lead to more than 20 000 emergency room admissions, over 2 000 hospitalizations and about 6 000 deaths. Carbon monoxide is the most common cause of mortality by poisoning in the US. Furthermore, carbon monoxide pollution is associated to neurocognitive abnormalities and behavioral disorders [Levy, 2015].

Raub et al. [1999] indicates that carbon monoxide can be found in various places. The most prominent exposures to higher carbon monoxide concentrations are due to the engines of motor vehicles, which are a part of the daily life of many individuals. In addition, work places, the commute, and a number of occupations lead to an increase in carbon monoxide exposure, affecting many families. The highest indoor exposures include restaurants, service stations and enclosed parking garages, whereas the lowest indoor exposures are found to be in homes, churches and health care facilities.

The World Health Organization (WHO) provides guidelines for human exposure to carbon monoxide. They specify the exposure times to certain carbon monoxide concentrations in the air, determined for carboxyhaemoglobin levels of 2.5% not to be exceeded (normal carboxyhaemoglobin levels of non-smokers average to be 1% while those of smokers average to 4%). The suggested exposure time for a carbon monoxide concentration in the air of 100 mg/m^3 is 15 minutes, for 60 mg/m^3 it is 30 minutes, and for 10 mg/m^3 it is 8 hours [Raub et al., 1999].

Climate Impact of Carbon Monoxide

According to Schneising et al. [2019], carbon monoxide can be understood as an "indirect agent of climate change", because it has an influence on the concentrations of multiple direct greenhouse gases. Also, Daniel and Solomon [1998] write that "changes in carbon monoxide emissions have been identified to be relevant to climate change", due to the relationship it has with methane and hydroxide concentrations, both of which have the ability to change the global average of surface temperatures. It directly contributes to global radiative forcing by absorbing and emitting infrared radiation, as well as indirectly due to its capacity to chemically change the abundances of other radiative gases such as methane, ozone and carbon dioxide.

The direct radiative effect and forcing of carbon monoxide is small enough to not be very significant, therefore the focus lies on its indirect effects as it influences both the concentrations and lifetimes of methane and hydroxide, which are both greenhouse gases. This highly significant indirect forcing of carbon monoxide and its effect on climate change is very hard to quantify but is of major importance [Sinha and Tuomi, 1996].

Environmental Impact of Carbon Monoxide

Carbon monoxide is an atmospheric pollutant that influences and endangers air quality, as the main gaseous constituents of air pollution are ozone, carbon monoxide, nitrogen dioxide (NO_2) and sulfur dioxide (SO_2) [Levy, 2015; Schneising et al., 2019]. Carbon monoxide is a very important atmospheric trace gas to better understand the tropospheric chemistry, and in a variety of urban areas it is considered a major atmospheric pollutant [Apituley et al., 2018]. Over continental and more populated areas the concentration of carbon monoxide is usually higher comparing to ocean air. In areas with tropical forests, carbon monoxide can be formed in the air above the forests. In some oceanic regions, carbon monoxide concentrations are observed to be higher during daytime and lower during nighttime, due to the carbon monoxide emissions of the ocean [Khalil and Rasmussen, 1990]. In addition, soils take up a fraction of carbon monoxide emissions from the atmosphere, estimated to be about 250 million tonnes per year [Raub et al., 1999].

The global distribution of carbon monoxide is considered both a primary and a secondary determinant of air quality. Due to being the dominating sink of hydroxyl radicals and an atmospheric tracer with a relatively long lifetime, it is used to study the global redistribution of pollutants. In fact, as an atmospheric tracer with such a long lifetime and such relatively simple chemistry, carbon monoxide illuminates the role of transport in the redistribution of chemical pollutants. Carbon monoxide has the greatest impact on air quality when it is exposed to sufficient nitrogen oxide (NO_X), being a precursor to tropospheric ozone, which again is a secondary pollutant that can also cause severe respiratory problems. When in contact with high nitric oxide (NO) concentrations, the hydroperoxyl (HO_2) product of carbon monoxide destruction loses an oxygen atom and forms nitrogen dioxide, which rapidly creates ozone. If there is no nitrogen oxide present, the hydroperoxyl molecule reacts immediately to destroy ozone (O_3) [Holloway et al., 2000].

As already mentioned, the reaction with hydroxyl radicals is the main sink removing carbon monoxide from the atmosphere. When considering the global scale, carbon monoxide removes more hydroxyl radicals from the atmosphere than methane does, on a regional scale this may vary (for example, in the southern hemisphere where there is less carbon monoxide and a similar level of methane compared to the northern hemisphere, there is much more removal of hydroxyl radicals by methane) [Raub et al., 1999].

Atmospheric Impact of Carbon Monoxide Carbon monoxide concentrations decreasing with altitude there are very high concentrations of carbon monoxide at the boundary layer [Sinha and Tuomi, 1996]. The troposphere is the lowest part of the Earth's atmosphere and is essential to examine and understand climate change. To do so, one must know what gases are present in the troposphere and in what quantities [Airbus Defence Space Dutch Technology, 2016].

The composition of the atmosphere, by volume, is as follows: 78% is nitrogen (N_2), 21% oxygen (O_2), and the remaining 1% is made up of noble gases and a variation of other minor gases. These atmospheric molecules that are only present in very small amounts (within this 1% of the Earth's atmosphere), are highly influential to the conditions at the Earth's surface. The troposphere, the lowest layer of the atmosphere, is the area that extends from the surface of the Earth up until the tropopause, at a height of approximately 10 to 15 km [Khlystova, 2010]. The troposphere is the most dense fraction of the atmosphere, containing 80% of the Earth's atmospheric mass [Murgatroyd, 2019]. The troposphere also has the highest variability in gases, the most important trace gases in the troposphere are methane, nitrous oxide, chlorofluorocarbons, ozone, water vapor, carbon dioxide, and carbon monoxide. The global burden from carbon monoxide is more uncertain when compared to methane or carbon dioxide due to its comparatively short atmospheric lifetime and the great variations in its emission patterns [Khlystova, 2010].

Not only the emissions of carbon monoxide vary across the globe, also the atmospheric lifetime of carbon monoxide varies a lot with latitude and seasonal changes compared to its global average. The average atmospheric lifetime is calculated to be about 2 months, but regularly ranges from 1 to 3 months. On a regional scale, large variations of this lifetime can be seen. For example, at high middle latitudes, during winter time carbon monoxide molecules have a lifetime of more than a year. At middle latitudes during summer time, the average lifetime of carbon monoxide is much closer to its average. Furthermore, in the tropical areas, carbon dioxide has an average atmospheric lifetime of only about 1 month [Raub et al., 1999; Khalil and Rasmussen, 1990].

1.2 Satellite Observations of Carbon Monoxide

Space based measurements were first made for time spans of several weeks in the years 1984 and 1994 with a correlation radiometer instrument called Measurement of Air Pollution from Satellite (MAPS) onboard the Shuttle spacecraft. Since this start of using satellites to take measurements of the Earth, a variety of space based instruments, including different spectral ranges and viewing geometries, trying to capture the global carbon monoxide variability were launched [Khlystova, 2010].

During the past decades carbon monoxide has been measured in a variety of ways, both with ground and aircraft measurements, as well as from space. In order to further investigate global carbon monoxide concentrations, it is necessary to record continuous global remote sensing observations, including a good sensitivity for sources and transport layers. In order to improve our current knowledge and understanding of the climate system, tropospheric chemistry and atmospheric transport processes, global coverage of detailed and continuous observations of carbon monoxide and methane are needed. This can only be achieved through the use of satellite measurements. There are two principal requirements for satellite instruments aiming to monitor atmospheric trace gases, which are a spectral resolution able to resolve the spectral signature of the considered molecules, and sensitivity to the lowest atmospheric layers, where the impact of surface sources and sinks is largest [Khlystova, 2010; Schneising et al., 2019].

1.3 Thesis Objectives and How They Are Addressed

In October 2017 the Sentinel-5 Precursor (also known as Sentinel-5P and S5P) satellite by the European Space Agency (ESA) and the European Commission (EC) Copernicus program, has been launched. This satellite enables the retrieval of atmospheric data products, which includes carbon monoxide. There are two algorithms that are currently used to generate global carbon monoxide products from Sentinel-5P, namely, the operational Copernicus SICOR algorithm, and the scientific WFM-DOAS algorithm, which is developed at the University of Bremen.

The objective of this thesis is to compare these two global carbon monoxide data products, in order to quantify systematic differences. Currently, the only existing comparison is that of Schneising et al. [2019], being the initial comparisons of these two data products.

The results of this comparison will be of relevance to the assessment of the information content of Sentinel-5 Precursor and to the discussions of strengths and weaknesses of the two considered data products. This information will be useful for further improvements of satellite products and has the potential to be taken into consideration during assessments related to ESA projects concerning the data quality of Sentinel-5P data products, which are carried out at the University of Bremen.

In order for this comparison to be done, different aspects of the retrievals for each data product are considered. Global maps of the retrievals of three considered days (June 6th, September 18th and November 4th, 2018) are generated and analyzed. The differences and relative differences of the collocated data points from the two data sets are both mapped and plotted against latitude. Furthermore, the latitudinal distribution of both the complete and the collocated data sets has been plotted, as well as the correlation of the collocated data sets.

This thesis is structured as follows. Section 2 explains the program and the mission delivering the data, which the comparison focuses on, the instrument that is used, and it includes a description of the data itself. Continuing with Section 3, there is the explanation of the methods that are used to compare the two data sets. It starts by describing previously used comparison methods by Schneising et al., and proceeds with the methods used for this thesis. This is followed by Section 4, the Results section, where the created maps and plots, as well as useful statistic values about the data are shown and described. This is then complemented with Section 5, the Discussion, where the results are put into context and further analyzed. Lastly, the project will be concluded in Section 6.

2 Satellite Data

The objective of this thesis is to compare two data products, the scientific product by the University of Bremen's Institute of Environmental Physics, the WFM-DOAS algorithm, and the operational product by the Netherlands Institute for Space Research (SRON), the SICOR algorithm. Both data products are level 2 carbon monoxide data from the TROPOMI instrument onboard the Sentinel-5 Precursor satellite, which is part of the Copernicus program. This section will give a detailed overview of the program, mission and instrument, where the considered data is obtained from, as well as the information required to understand compare the data sets.

2.1 Copernicus Programme

The Directorate of Earth Observation Programmes (EOP) of the European Space Agency aims to pursue scientific knowledge with the objective of transforming it to benefit society. On this aim, EOP has three main branches of projects: the Sentinel missions for Earth monitoring, meteorological missions for weather monitoring and forecasting, and the Earth Explorers for scientific research missions [ESA, 2014].

The Sentinel satellite series is part of the Copernicus Programme, which aims to provide accurate and timely information to improve environmental management, to understand and mitigate the effects of climate change and to ensure civil security, with easily accessible data and information. "Copernicus will help shape the future of our planet for the benefit of us all" [ESA, 2019a]. Initially under the name of Global Monitoring for Environment and Security (GMES), the program is directed by the European Commission (EC) in partnership with ESA and other agencies. In total, the European Space Agency coordinates about 30 satellites under the guidance and requirements of the EC and the European Union (EU). For the operational needs of the Copernicus Programme, ESA developed a family of Earth observing satellites called Sentinels. This series of satellites is a space component which constitutes the European contribution to the Global Earth Observation System of Systems (GEOSS). Copernicus provides a unified system, through which immense amounts of data are distributed to a variety of organizations and services that are designed to have beneficial impacts on the environment we live in, our everyday lives, humanitarian needs and that supports effective policy making for a more sustainable future. This program has its services split up in 6 major groups: land management, marine environment, atmosphere, emergency response, security and climate change. With this program, the European Commission and the European Space Agency aim to support the European goals with respect to sustainable development and global governance of the environment [Veefkind et al., 2012; ESA, 2019a].

The Copernicus Programme comprises 7 Sentinel satellites currently in orbit, namely Sentinel-1 (A and B), Sentinel-2 (A and B), Sentinel-3 (A and B) and Sentinel-5P, and five additional satellites are planned to be launched in the near future, namely Sentinel-6 (A and B) to be launched November 2020 and 2025, Sentinel-4 (A and B) to be launched in 2024 and 2030, and Sentinel-5 (A) to be launched in 2023 (with

Sentinel-5B and Sentinel-5C to then be launched in 7 year intervals) [Levrini, 2020].

The different groups of Sentinel satellites observe and monitor different aspects of the Earth. Sentinel-1 are polar orbiting satellites, with a day and night radar imaging mission for land and ocean services. Sentinel-1A was launched in April 2014 and Sentinel-1B was put into orbit in April 2016. Sentinel-2 are polar orbiting, multi-spectral satellites, on high-resolution imaging mission for land monitoring. They aim to provide information on vegetation, soil and water coverage, inland waterways and coastal areas. They are also a tool for emergencies management. Sentinel-2A was launched in June 2015 and Sentinel-2B was put into orbit in March 2017. Sentinel-3 is a multi-instrument mission measuring sea surface topography, sea- and land-surface temperature, and ocean and land color with high-end accuracy and reliability. This is used for ocean forecasting systems and environmental and climate monitoring. Sentinel-3A was launched in February 2016 and Sentinel-3B was put into orbit in April 2018. Sentinel-5P is a mission developed to reduce the data gap between the Environmental Satellite (ENVISAT) mission with the SCanning Imaging Absorption spectroMeter for Atmospheric CHartographY (SCIAMACHY) instrument onboard, and the Sentinel-5A instrument to be launched on the Metop Second Generation satellite. Sentinel-5P was the first Copernicus satellite to go in orbit to monitor the Earth's atmosphere, and was launched in October 2017. Sentinel-6 will have a radar altimeter to measure sea-surface height, planned to be used for operational oceanography and for climate studies. Sentinel-6A will be launched in November 2020 and Sentinel-6B in 2025. Sentinel-4 and Sentinel-5 are planned to provide data for atmospheric composition monitoring from geostationary and polar orbits respectively. Sentinel-4A is to be launched in 2024, Sentinel-4B in 2030 and Sentinel-5A in 2023, and Sentinel-5B and Sentinel-5C in 7 year intervals. ESA is planning to expand Copernicus by including C and D satellites of the different Sentinels and further missions for more detailed and specific monitoring of the Earth [ESA, 2019a,b; Levrini, 2020].

2.2 Sentinel-5 Precursor Satellite

Sentinel-5P is a single payload satellite mission, in a low Earth orbit, providing daily global information on the concentrations of trace gases and aerosols in the air. The objective of the mission is to globally monitor air quality, climate forcing, the ozone layer and surface UV radiation. S5P targets to provide information and services on climate and air quality from 2017 to 2023, when Sentinel-5A is planned to be in orbit and take over. The operational phase of the Sentinel-5P mission started in April 2018. The mission is made up of a satellite bus, the payload and a ground system. The single payload of the mission is the TROPOspheric Monitoring Instrument (TROPOMI), an instrument that has been developed by the Netherlands in cooperation with the European Space Agency [Veefkind et al., 2012; Apituley et al., 2018].

As indicated, Sentinel-5P is meant to be a "gap-filling" mission between the end of the OMI (Ozone Monitoring Instrument) and SCIAMACHY missions and the launch of Sentinel-5. In addition, it is also a preparatory program for the product and application definitions of Sentinel-5. Due to being such a "gap-filler" and preparatory mission, the mission had two principal constraints: a limited budget and a short development time. The technical definitions of S5P are based on national studies done in the Netherlands [Veefkind et al., 2012].

Sentinel-5P was launched in October 2017 and the predicted in-orbit lifetime of the satellite is seven years. The reference orbit of the spacecraft is a near-polar frozen sun-synchronous orbit, which has been adapted for the optimization of the mission, with the mean Local Solar Time at Ascending Node (LTAN) at 13:30 and a repeat cycle of 17 days. The orbital height of the satellite is 824 km and has been chosen to synergy with the U.S. Suomi National Polar-orbiting Partnership (NPP) mission [Hille, 2015]. This early afternoon orbit has been selected because at this time the boundary layer is well developed and gives more information about the pollution emitted during the day. It is also the latest that can be used to forecast air quality and for warnings concerning the next day [Veefkind et al., 2012].

There have been some spaceborne instruments measuring global carbon monoxide levels already, such as the Atmospheric Infrared Sounder (AIRS), the Tropospheric Emission Spectrometer (TES) and the Infrared Atmospheric Sounding Interferometer (IASI) observing emissions in the thermal infrared (TIR). Additionally also the Measurement of Pollution in the Troposphere (MOPITT), which combines observations in TIR and shortwave infrared (SWIR), enabling an increase in surface-level sensitivity for some scenes. This sensitivity can be achieved at all altitudes by using radiance measurements of reflected solar radiation in the SWIR wavelengths, and was previously demonstrated with SCIAMACHY [Buchwitz et al., 2007; Burrows et al., 1995].

2.3 TROPOMI Instrument

TROPOMI is the only instrument on board of the Copernicus Sentinel-5 Precursor satellite [KNMI R&D Satellite Observations, 2011; SRON, 2019]. It is an instrument mapping the Earth's atmosphere. The instrument works by measuring the sunlight reflected by the Earth's surface through the radiance port, while the direct sunlight is measured with the irradiance port [SRON, 2019]. It measures the levels of atmospheric trace gases with a passive remote sensing technique from the top of the atmosphere. It does so by comparing the light reflected by the Earth's surface and atmosphere with measurements of the light coming directly from the sun. This allows the instrument to map the levels of gases such as ozone, carbon monoxide and methane present in the Earth's atmosphere. In addition it is able to monitor the amount of volcanic ash in the atmosphere, using measurements in the ultraviolet (UV) spectral range [Airbus Defence Space Dutch Technology, 2016]. The requirements for the TROPOMI instrument were compiled in the Dutch national studies, and are derived from the requirements of the Level 2 data products [Apituley et al., 2018; Veefkind et al., 2012].

TROPOMI maps the globe every 24 hours and has a spatial resolution, which is high enough to detect air pollution for individual cities and areas. The spatial resolution for TROPOMI is 7×3.5 km^2 for bands 2 to 6 (ultraviolet, visible light and near-infrared), 7×7 km^2 for bands 7 and 8 (shortwave infrared) and 21×28 km^2 for band 1 (deep ultraviolet). Due to this high resolution, which is a significant advancement compared to previous instruments, it enables scientists to study atmospheric trace gases with unprecedented level of detail [Airbus Defence Space Dutch Technology, 2016]. The instrument works with a wide swath push-broom configuration, and images the Earth in strips, on a two-dimensional detector for periods of 1s each, in which the satellite moves approximately 7 km. These strips have the dimensions of about 2 600 km in the across track direction and about 7 km in the along track direction. When the 1 second measurement is taken, the next one is started right away, which means that the instrument scans the globe while the satellite is in motion [Veefkind et al., 2012]. The light of an entire swath is recorded at the same time, and is then dispersed on two-dimensional imaging detectors (the position along the swath on one detector and the spectral information of the positions on the other). The combination of this high resolution with the wide swath is what enables the daily global coverage of the instrument [Apituley et al., 2018].

The TROPOMI design is structured in four modules, the first containing the telescope, the ultraviolet, visible light and near infra-red spectrometers and the calibration unit. The second module holds the SWIR spectrometer and relay optics. The third encloses the Instrument Control Unit (ICU) and the fourth module is the cooler. TROPOMI also includes some components from the OMI design, such as the wide-field telescope, the polarization scrambler and the two-dimensional detectors. The instrument uses two spectrometer modules behind a common telescope; one of the two covers the ultraviolet to visible spectral range (270-495 nm) and near infra-red range (675-775 nm), and the other spectrometer covers the SWIR spectral range (2 305-2 385 nm), enabling it to observe atmospheric constituents such as ozone, nitrogen dioxide, carbon monoxide, sulfur dioxide, methane, formaldehyde, aerosols and clouds [SRON, 2019]. The instrument has four detectors, split among the different spectral bands. In the SWIR, the spectral resolution is about 0.25 nm with a spectral sampling interval of 0.1 nm [Sentinel-5P Mission Performance Centre, 2018]. The SWIR spectrometer is made up of a slit, collimator mirror optics, an immersed grating, camera optics of multiple lenses and a Mercury Cadmium Telluride (HgCdTe) detector. The latter has 1 000 columns in spectral direction and 256 rows in spatial dimension. 976 of the columns and 217 of the rows are illuminated. The TROPOMI-SWIR spectrometer is used to retrieve the atmospheric trace gases carbon monoxide, methane and water vapor [SRON, 2019]. The instrument is able to measure the full range of signals, from dark oceans to bright clouds. Nevertheless, especially for observing the lower troposphere, the ideal pixels to analyze are the cloud-free ones, and those that almost are. For clear sky observations, TROPOMI provides total carbon monoxide columns including sensitivity to the tropospheric boundary layer, while for rather cloudy observations the column sensitivity may change [Veefkind et al., 2012; Apituley et al., 2018].

The TROPOMI instrument has four main scientific objectives. The first is to better constrain the spatiotemporal variability, the evolution and the strength of the trace gas sources and aerosols that have an effect on both climate and air quality. The second objective is to improve the understanding of climate forcing by gaining a better understanding of what controls the atmospheric lifetime and distribution of methane, tropospheric ozone and aerosols. Third is to be able to better estimate the tropospheric long-term trends in regards to climate and air quality, ranging from regional to global scales. The fourth and last objective for this instrument is to both develop and improve air quality model processes and data assimilation, in order to support operational services, such as protocol monitoring and air quality forecasting [Veefkind et al., 2012].

2.4 Operational SICOR CO Retrieval Algorithm and Data Product

The Shortwave Infrared CO Retrieval (SICOR) algorithm has been developed for the operational processing of the TROPOMI data by the Netherlands Institute for Space Research. The algorithm simultaneously retrieves the carbon monoxide total column densities and effective cloud parameters, such as cloud optical thickness and cloud center height, to account for the atmospheric light path [Borsdorff et al., 2018a,b]. This algorithm retrieves data for land areas and cloudy ocean regions, as the surface albedo of the ocean is too low to retrieve carbon monoxide at clear sky conditions [Landgraf et al., 2018].

The algorithm provides the vertically integrated column of carbon monoxide and includes the corresponding averaging kernels for the individual measurements, which define the sensitivity of the retrieved carbon monoxide column to changes in the true vertical carbon monoxide profile. Carbon monoxide columns are derived from sunlight that has been reflected by the Earth's atmosphere, in the spectral range between 2305 nm and 2385 nm. This is a spectral range, which at clear sky conditions only experiences little scattering, meaning most radiation is reflected by the Earth's surface. Therefore, this is the most suitable spectral range for detecting carbon monoxide sources [Landgraf et al., 2018]. The retrieval is based on the profile scaling approach, which scales a reference profile of carbon monoxide, in order to fit the spectral measurements. This leads to the degree of freedom of the carbon monoxide signal to be 1 by definition, even for retrievals of cloud contaminated measurements [Borsdorff et al., 2018a,b].

For the retrieval using this algorithm, four types of inputs are required. The first is the measured Earth radiance and solar irradiance spectra, as well as noise estimate, solar and viewing geometry, and information of geo-location. The second kind of input is European Centre for Medium-Range Weather Forecasts (ECMWF) temperature, water vapor and pressure profiles, and geo-potential height. Thirdly, an estimate of the methane field using a chemistry transport model is required to obtain cloud information, by comparing with the retrieved methane. The fourth type of needed input is an estimate of the carbon monoxide profile from a chemistry transport model [Landgraf et al., 2018]. This reference profile for carbon monoxide that is scaled during the retrievals comes from simulations of the global chemical transport model TM5 and is averaged monthly over $3^\circ \times 2^\circ$ latitude $\times$ longitude boxes. The result of the retrieval is the carbon monoxide total column density in [molecules cm^{-2}] [Borsdorff et al., 2018a,b].

The retrieval of the carbon monoxide data product from the TROPOMI SWIR measurements is based on a Philips-Tikhonov regularization scheme, and is done in two main steps. In the first step, the vertically integrated amount of methane is retrieved from the SWIR band between 2315 nm and 2324 nm, by using a non-scattering radiative transfer model. This first step is necessary to obtain critical information about cloudiness, which is subsequently used in the second step to infer carbon monoxide columns from the adjacent spectral window (2324 nm - 2338 nm). Besides this information on cloud scattering, modelled information on carbon monoxide and water vapor is used as a first guess in this step. The final retrieval product is a carbon monoxide column estimate including a column averaging kernel and an estimate of the random error [Landgraf et al., 2018].

The TROPOMI/SICOR data used for this thesis is based on TROPOMI Level 1b files. June 6th (20180606) and September 18th (20180918) retrievals are based on the operational Level 2 V01.02.02 files, and the November 4th (20181104) retrievals on the operational Level 2 V01.03.01 files.

The data files were obtained from the Institute of Environmental Physics at the University of Bremen.

2.5 Scientific WFM-DOAS CO Retrieval Algorithm and Data Product

The Weighting Function Modified Differential Optical Absorption Spectroscopy algorithm (WFM-DOAS), is a scientific algorithm which is used to simultaneously retrieve carbon monoxide and methane, aiming to complement the operational algorithms, looking to provide new geophysical insights. The algorithm seeks to perform in accordance to the mission requirements regarding random and systematic errors. Nevertheless, it has a number of differences when compared to the operational algorithm and can be used in combination with the operational ones to assess the robustness of retrieval results [Schneising et al., 2019].

WFM-DOAS is a linear least-squared method, which is based on scaling pre-selected atmospheric vertical profiles, using the U.S. standard atmosphere profiles. To retrieve the carbon monoxide data, sun-normalized radiances in the SWIR spectral region with noise estimates are used. GMTED2010 data on topography (Global Multi-resolution Terrain Elevation Data 2010) and the solar geometry are used among other parameters to select the most suitable reference spectrum from a look-up table. ECMWF data are required in the post-processing to convert the retrieved vertical columns to column-averaged mole fractions. With the use of this algorithm the vertical columns of the desired gases, such as carbon monoxide, are determined by fitting a linearized radiative transfer model to the measured sun-normalized radiance. Other than the operational SICOR data product, WFM-DOAS provides data for cloud free scenes only and mainly over land as column-averaged dry air mole fractions in parts per billion. Data over the ocean is mostly limited to sun-glint conditions, due to the otherwise weak signal over water. Further differences of the algorithm compared to TROPOMI/SICOR are the quality filters, the spectroscopy that is used, and the state vector elements, such as the treatment of clouds and aerosols, although similar spectral bands are used [Schneising et al., 2019, 2020; Schneising, 2019].

The WFM-DOAS data used in this thesis is based on TROPOMI Level 1b V01.00.00 files using the algorithm version TROPOMI/WFMD v1.2, which is available at *https://www.iup.uni-bremen.de/carbon_ghg/products/tropomi_wfmd/*.

For the comparison performed in this thesis, Dr. Oliver Schneising from the Institute of Environmental Physics at the University of Bremen has prepared a special data set, which also includes the carbon monoxide columns in [mol m^{-2}], in order to facilitate the direct comparison with the operational product.

3 Methodology

Aiming to compare the two data products, the methods used to conduct this research will be inspired by the comparison methods from "*A scientific algorithm to simultaneously retrieve carbon monoxide and methane from TROPOMI onboard Sentinel-5 Precursor*" [Schneising et al., 2019] where the SICOR data product was compared to Total Carbon Column Observing Network (TCCON) ground based measurements and the operational data product of the European Space Agency.

3.1 Comparison methods used by Schneising et al.

Schneising et al., in "*A scientific algorithm to simultaneously retrieve carbon monoxide and methane from TROPOMI onboard Sentinel-5 Precursor*", gives comparing figures for the relation between the S5P TROPOMI measurements and the ground-based measurements by TCCON.

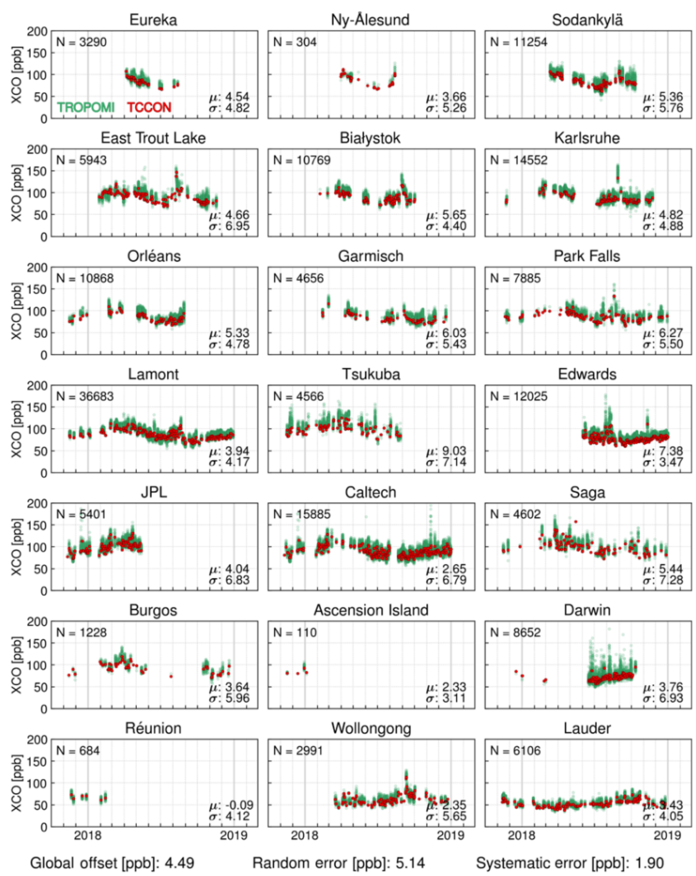


Figure 2: O. Schneising's "Comparison of the TROPOMI/WFMD v1.2 XCO time series (green) with ground-based measurements from the TCCON (red)" (Figure 10, Schneising et al. [2019]).

For instance, in Schneising et al. [2019] there is a figure, where for each TCCON ground station, a one-year time series of the TROPOMI/WFMD XCO and the TCCON ground-based measurements is made. For each station the two data sets are plotted in two different colors on one set of axes. Then, the collocated data is compared, by plotting the correlation of the data. Figure 2 shows this comparison. The TROPOMI measurements are shown in green and the TCCON measurements in red. N is the number of collocations, μ refers to the mean bias and σ to the scatter of the satellite data compared to TCCON in [ppb].

In addition, global maps are shown with daily, monthly and yearly data, which is then compared to a similar plot of the operational product. A similar representation is given with a local map of a bimonthly distribution of carbon monoxide over China, India and Southeast Asia. Part (a) of Figure 3 shows a global map of all TROPOMI/WFMD carbon monoxide measurements passing the quality filter from December 2018 in [10^{18} molecules per cm^{-2}]. Underneath there is a similar representation of the TROPOMI/Operational data. In part (b) one can see a bivariate histogram of all collocated measurement points passing the quality filters of both algorithms. The linear regression, correlation of the data, the mean and the standard deviation of the difference are also shown [Schneising et al., 2019].

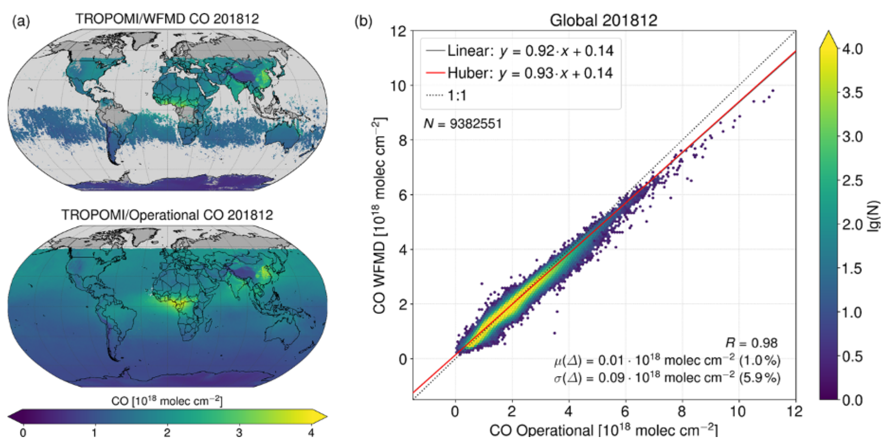


Figure 3: "Comparison of TROPOMI/WFMD CO with the operational TROPOMI/SICOR data for December 2018" (Figure 15, Schneising et al. [2019]).

3.2 Comparison methods used in this Thesis

In order to compare the Sentinel-5P TROPOMI WFM-DOAS and operational (SICOR) products, the data sets from these two algorithms will be compared for three days, June 6th, 2018 (20180606), September 18th, 2018 (20180918) and November 4th, 2018 (20181104). To facilitate working with this data, the original data files were loaded and read in Python and the information required for the comparisons was written out in separate text files to have smaller files to work with. The regular latitude and longitude, corner latitude and longitude, and the carbon monoxide measurements, filtered with the suggested quality flags were written in these new data files. Furthermore, for better comparisons later on, the two data sets have been collocated. To achieve this, a Python program was written to find the exact points at which there are data retrievals in both products, and to write them out in a separate text file. In order to do so all data points were compared by ground pixel and scan line, and when retrievals for a given ground pixel and scan line combination existed, they were written in the new file.

To obtain a first impression of the data, four types of global maps have been plotted using the *basemap* tool in Python. The first map shows the global collocated SICOR data, the second map shows the global collocated WFM-DOAS data, the third map the absolute difference between the two data sets and the fourth map shows the relative difference. This has been done for all three considered days.

In order to better understand and compare the data, a series of four plots has been made for each day, using *matplotlib* plotting tools. The first plot shows all measurements of both the scientific and the operational data products by latitude, with latitude in [degrees] on the x-axis and carbon monoxide in [mol m^{-2}] on the y-axis. In the upper right corner the number of considered retrievals for each of the two products is shown. The second plot was structured like the first plot, but showing the collocated data retrievals. For the first two plots, the SICOR data is shown in pink and the WFMD data in blue. The third plot shows the difference between the two collocated data sets, namely WFMD - SICOR, with errorbars representing the mean and the standard deviation. The x-axis shows the latitude in [degrees] and the y-axis shows the difference (WFMD-SICOR) in [mol m^{-2}]. The fourth and last plot is a bivariate histogram of the two collocated data sets, with the collocated SICOR carbon monoxide retrievals (CO) in [mol m^{-2}] on the x-axis and the collocated WFMD carbon monoxide retrievals (CO) in [mol m^{-2}] on the y-axis. In the lower left corner the number of plotted points (N), the mean of the difference (D), the standard deviation of the difference (S) and the Pearson correlation coefficient (R), calculated with *NumPy* and *SciPy* statistic tools, are shown. The colorbar on the right side of the plot indicates the amount of retrievals shown at a point of a given color, with red indicating many retrievals and purple very few.

Furthermore, as an overview of the statistic of the data, five tables have been compiled. Each table shows the number of data points of the considered data set (N), the minimum and maximum value of the data set (min and max), the mean, and the standard deviation (St.Dev.), for each of the considered days. Minimum, maximum, and mean are in [mol m^{-2}]. This type of table has been compiled for the complete SICOR data set (Table 1), the collocated SICOR data set (Table 2), the complete WFMD data set (Table 3), the collocated WFMD data set (Table 2) and for the absolute difference between the two collocated data sets (Table 5).

3.2.1 Working with the Data Files

The following three figures, namely Figure 4, Figure 5 and Figure 6, give an overview of what should be regarded and taken into account when working with this data and to build a comparison of the kind done in this thesis.

Figure 4 refers to reading and extracting the SICOR data product from its files, Figure 5 outlines how to do so with the WFM-DOAS data files, and Figure 6 explains the main steps in collocating the data. The full programming code is given in Appendix 1 (Reading the Data and Creating Text Files with the Data for Further Analysis).

Step 1: Reading and Converting the SICOR Data File

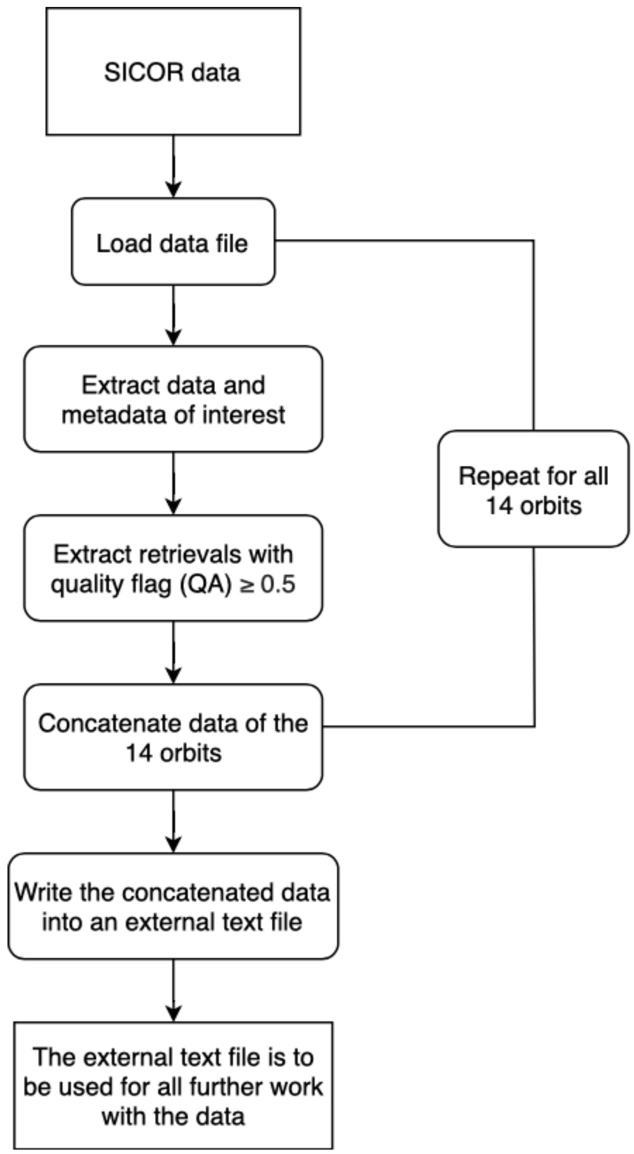


Figure 4: Outline of the most relevant steps to working with SICOR data files.

Step 2: Reading and Converting the WFM-DOAS Data File

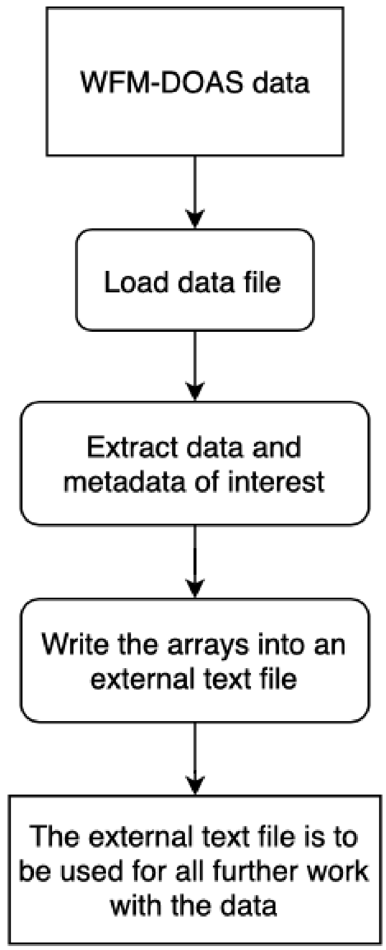


Figure 5: Outline of the most relevant steps to working with WFM-DOAS data files.

Step 3: Collocating the Data Sets

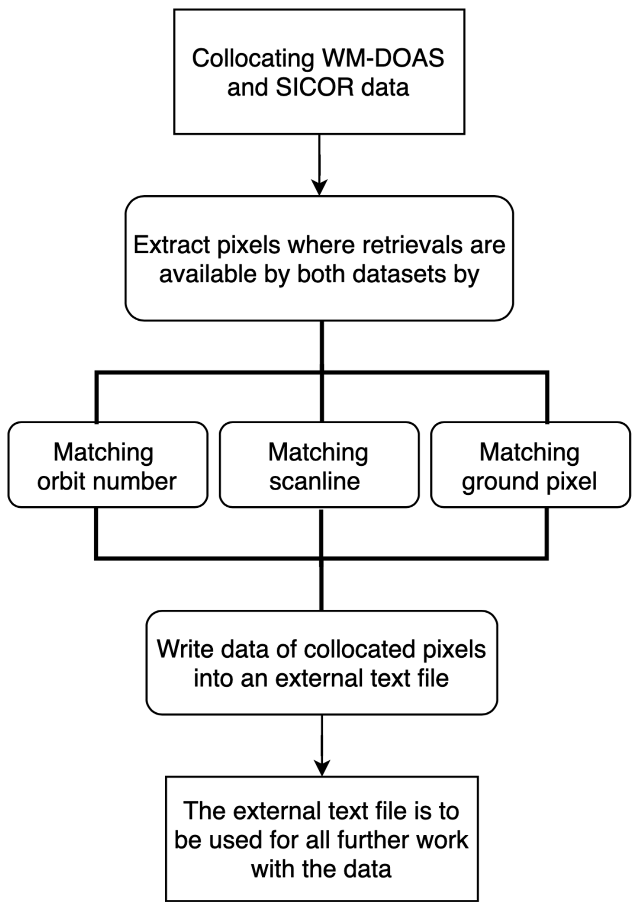


Figure 6: Outline of the most relevant steps to be considered when collocating the data sets.

4 Results

This section is divided into three subsection. The first showing the comparisons done with the use of global maps on which the data is displayed. The second showing the retrievals and their differences plotted against latitude, enabling the further observation of the spacial distribution of the data and the absolute differences between the products, as well as their correlation. The third and last subsection gives an overview of further statistic measures used in the comparison.

4.1 Comparison of global daily maps

For each day, four maps have been generated, two maps showing the measurements of the collocated retrievals for both data products and two maps to show the absolute and relative differences of the retrievals.

4.1.1 June 6th, 2018

In this subsection, the global maps for June 6th, 2018, are presented and discussed.

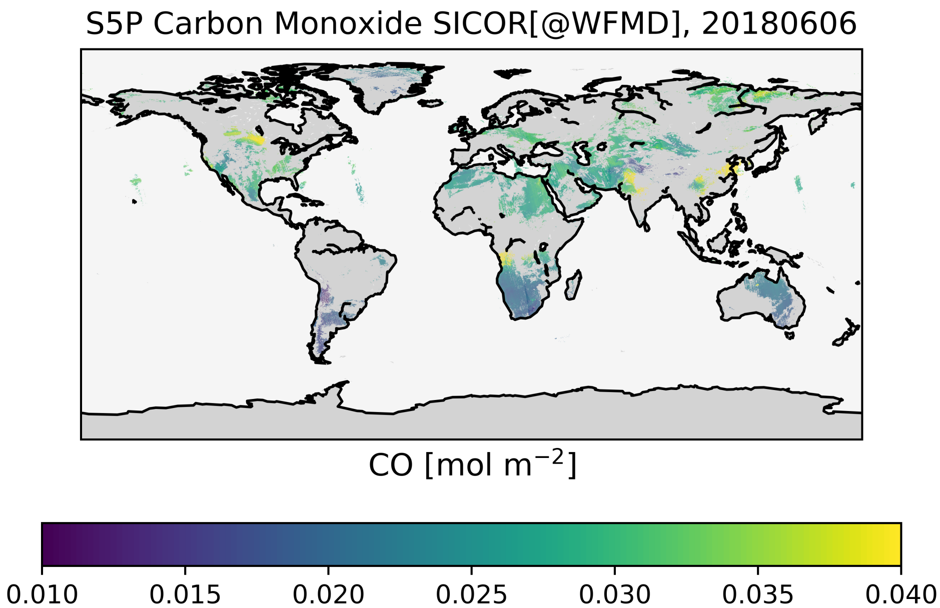
Figure 7 (a) and (b) show the carbon monoxide column retrieval values, denoted as CO, for the collocated data pixels of June 6th. Figure 7 (a) shows the SICOR retrievals and Figure 7 (b) shows the WFM-DOAS retrievals. Retrievals are mainly over land, but some can be seen over the oceans as well. Enhancements are recognizable in northern Canada, in Africa near Angola, in north-western India, and China. They are seen on both maps. In addition, these areas seem to have a very similar level of enhancement. Furthermore, also in areas with low carbon monoxide levels such as South America, southern Africa and central Australia seem to have similar levels across the products.

Figure 8 shows the absolute (a) and relative (b) differences between the WFM-DOAS and the SICOR retrievals for June 6th. A positive difference, marked in shades of blue, indicates WFM-DOAS measurements to be higher, and a negative difference, marked in shades of orange and red, indicates the SICOR measurements to be higher. The prevailing differences range to a maximum of about 0.003 [mol m^{-2}], which corresponds to about 10%. The differences also seem to be balanced in both directions, with slightly more differences shown in red, suggesting the readings of the SICOR operational data product to be slightly higher.

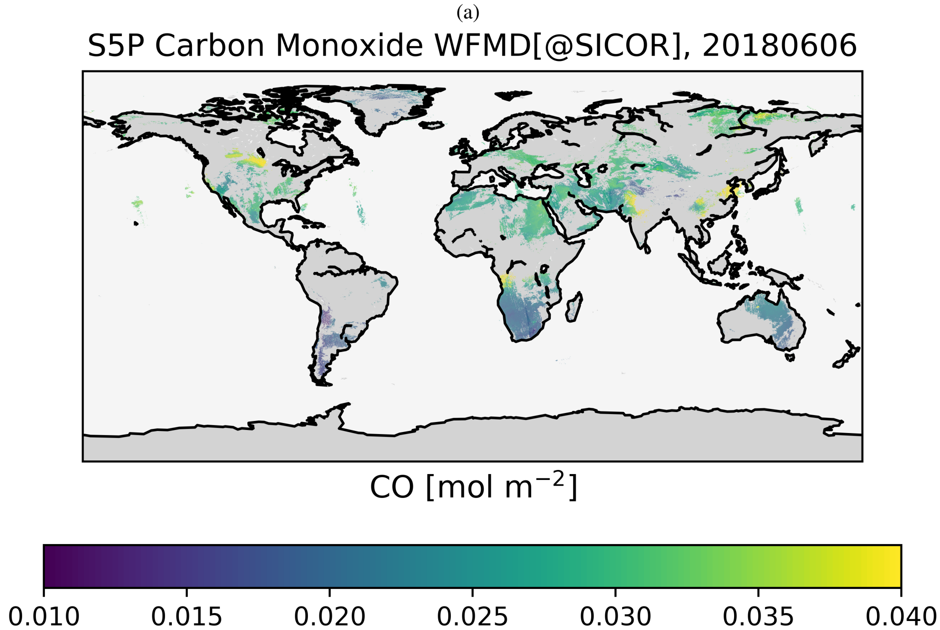
4.1.2 September 18th, 2018

In this subsection, the global maps for September 18th, 2018, are presented and discussed.

Figure 9 (a) and (b) show the carbon monoxide column retrieval values for the collocated data pixels of September 18th. Figure 9 (a) shows the SICOR retrievals and Figure 9 (b) shows the WFM-DOAS retrievals. More collocated retrievals can be observed when compared to June 6th, as well as more retrievals over the oceans. There is a large area of strong enhancement in southern Africa and Madagascar, and both the areas of strong enhancement and of low carbon monoxide levels seem to be relatively similar across the two data products.

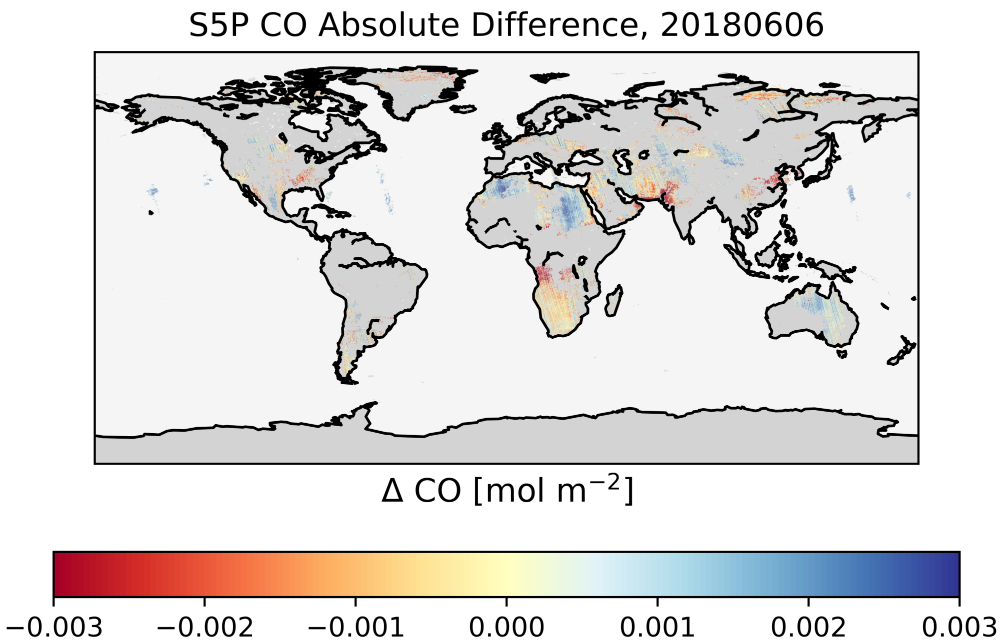


(a)

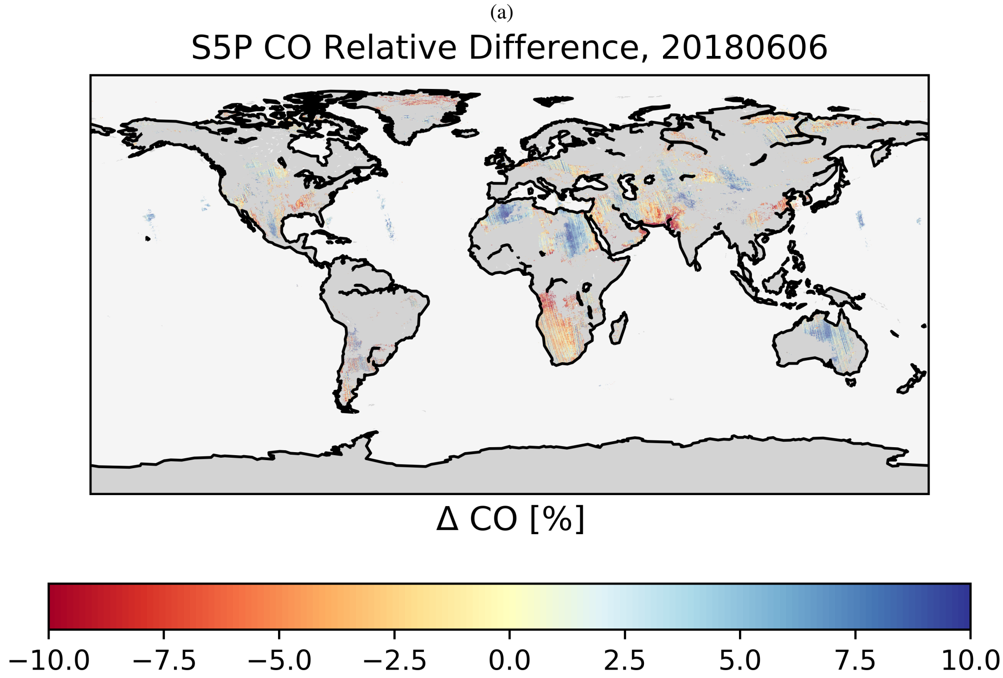


(b)

Figure 7: Collocated carbon monoxide retrievals in [mol m^{-2}] for June 6th, 2018, passing the suggested quality filter of the respective algorithms. (a) shows the TROPOMI/SICOR retrievals and (b) the TROPOMI/WFM-DOAS retrievals.

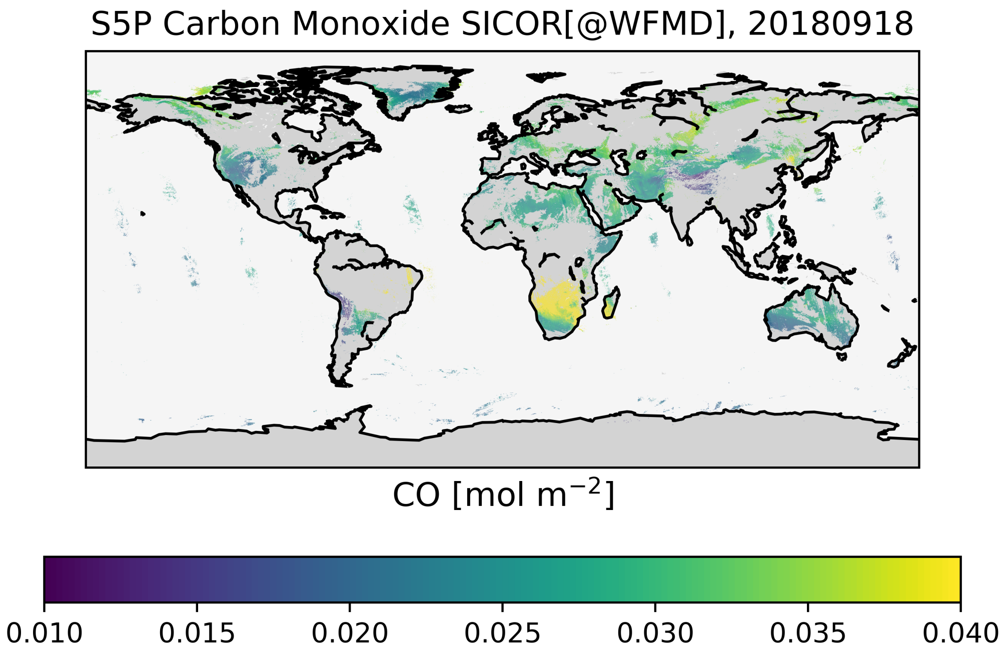


(a)

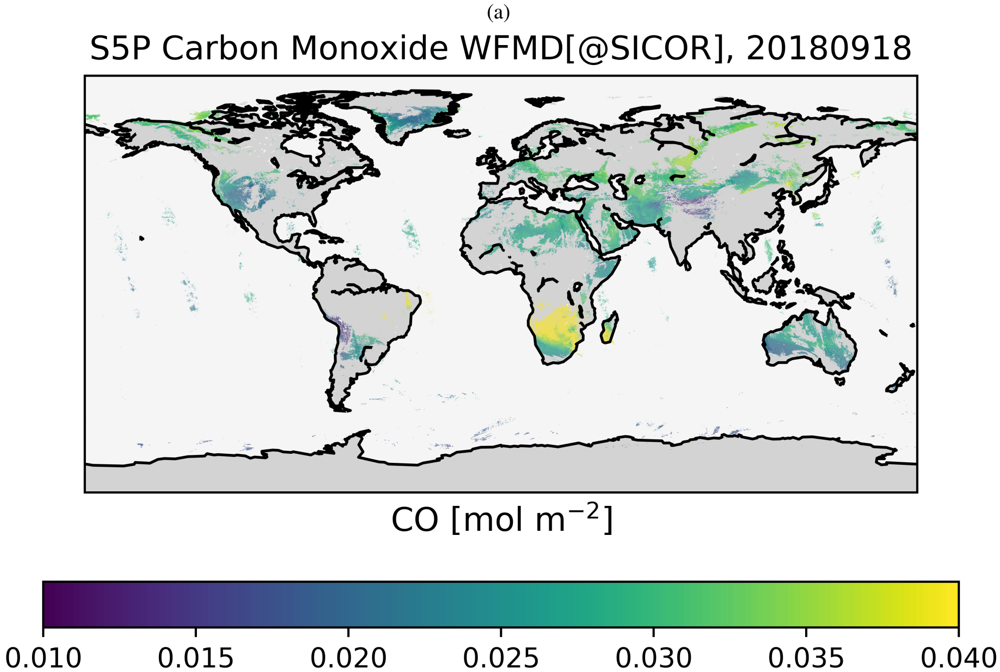


(b)

Figure 8: Difference collocated WFMD - collocated SICOR carbon monoxide retrievals for June 6th, 2018. (a) shows the absolute difference in [mol m^{-2}] and (b) the relative difference ((TROPOMI/WFMD - TROPOMI/SICOR) / TROPOMI/SICOR x 100 %).

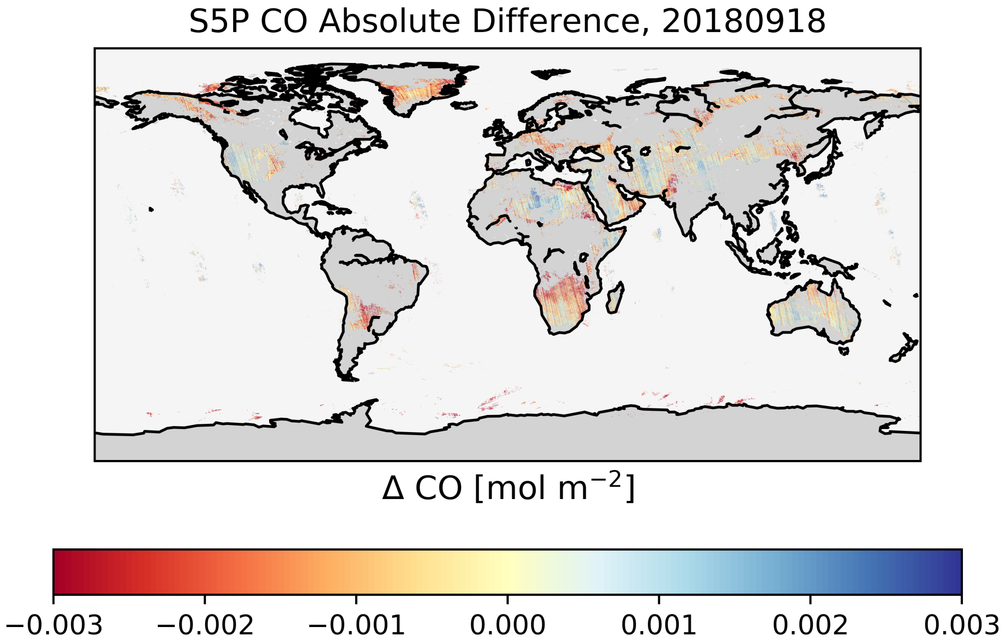


(a)

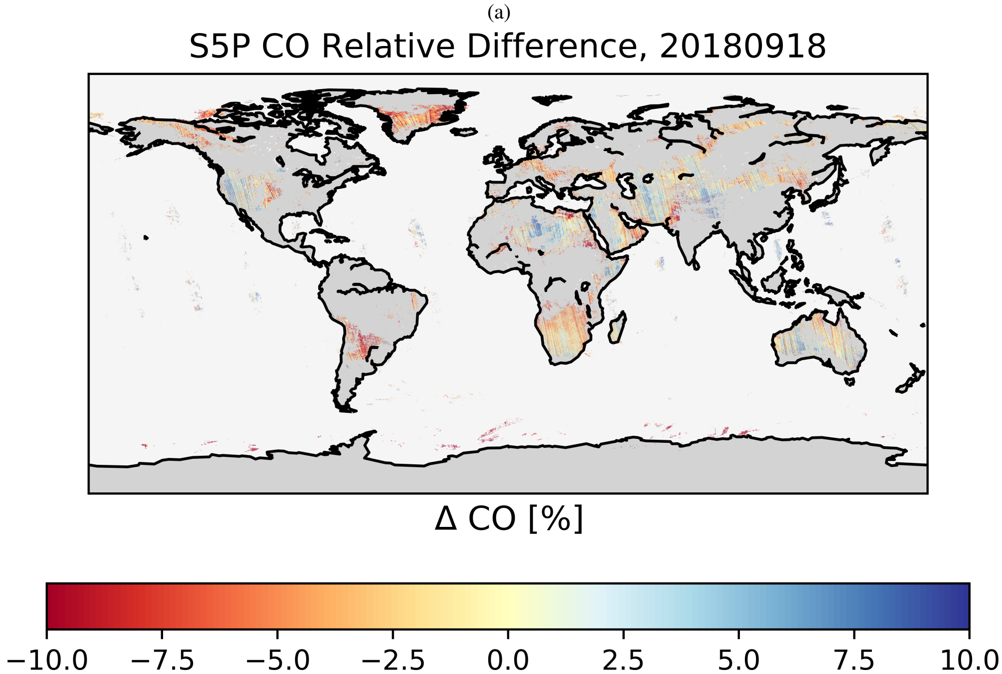


(b)

Figure 9: Collocated carbon monoxide retrievals in [mol m^{-2}] for September 18th, 2018, passing the suggested quality filter of the respective algorithms. (a) shows the TROPOMI/SICOR retrievals and (b) the TROPOMI/WFM-DOAS retrievals.



(a)



(b)

Figure 10: Difference collocated WFMD - collocated SICOR carbon monoxide retrievals for September 18th, 2018. (a) shows the absolute difference in [mol m^{-2}] and (b) the relative difference ((TROPOMI/WFMD - TROPOMI/SICOR) / TROPOMI/SICOR x 100 %).

Figure 10 shows the absolute (a) and relative (b) differences between the WFM-DOAS and the SICOR retrievals for September 18th. Significantly more red areas can be observed, indicating that the operational data product has a strong tendency to have higher measurements. In addition, a fair amount of regions with low or almost no difference between the products can be seen. There is a strong red absolute difference in the southern part of Africa and mid South America, and a strong red relative difference in Greenland.

4.1.3 November 4th, 2018

In this subsection, the global maps for November 4th, 2018, are presented and discussed.

Figure 11 (a) and (b) show the carbon monoxide column retrieval values for the collocated data pixels of November 4th, 2018. Figure 11 (a) shows the SICOR retrievals and Figure 11 (b) shows the WFM-DOAS retrievals. Compared to the other two days (June 6th and September 18th, 2018), there are even more collocated retrievals over sea, but not enough to really comment on them. November 4th, shows very good coverage of northern Africa, where an area of enhancement can also be recognized and is better visible for the scientific product. Another enhancement is seen in Australia, and it is visible relatively equally in both products.

Figure 12 shows the absolute (a) and relative (b) differences between the WFM-DOAS (or WFMD) and the SICOR retrievals for November 4th. The retrievals in the Antarctic show the SICOR product to have significantly higher measurements, visible in dark red, while overall November 4th shows more blue areas than the other two days. This large amount of blue differences, suggesting WFMD measurements to be higher on this day, is especially seen for the retrievals in Africa and Australia. In addition, there are also large areas of almost no difference, seen in light yellow, which is more prevalent than in the other two analyzed days.

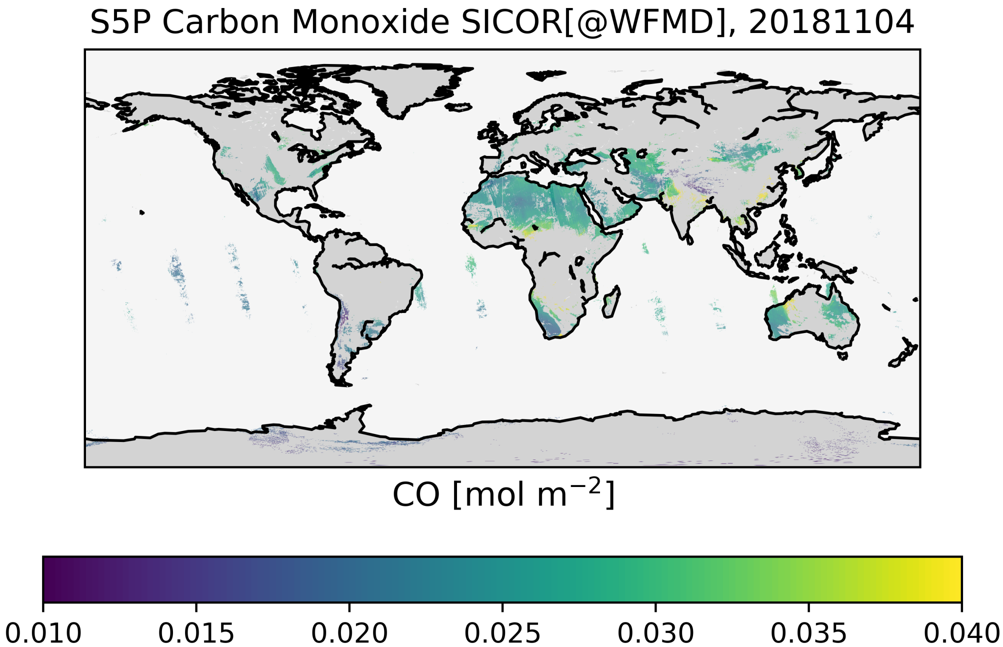
4.2 Comparison of data distribution

For each day, four plots have been generated. The first two show the carbon monoxide column retrievals plotted against latitude, correlated and non-correlated, with the two data products shown in different colors. The third plot shows the latitudinal distribution of the absolute differences between the two collocated data products, and the fourth plot is a bivariate histogram showing the correlation of the two collocated data products.

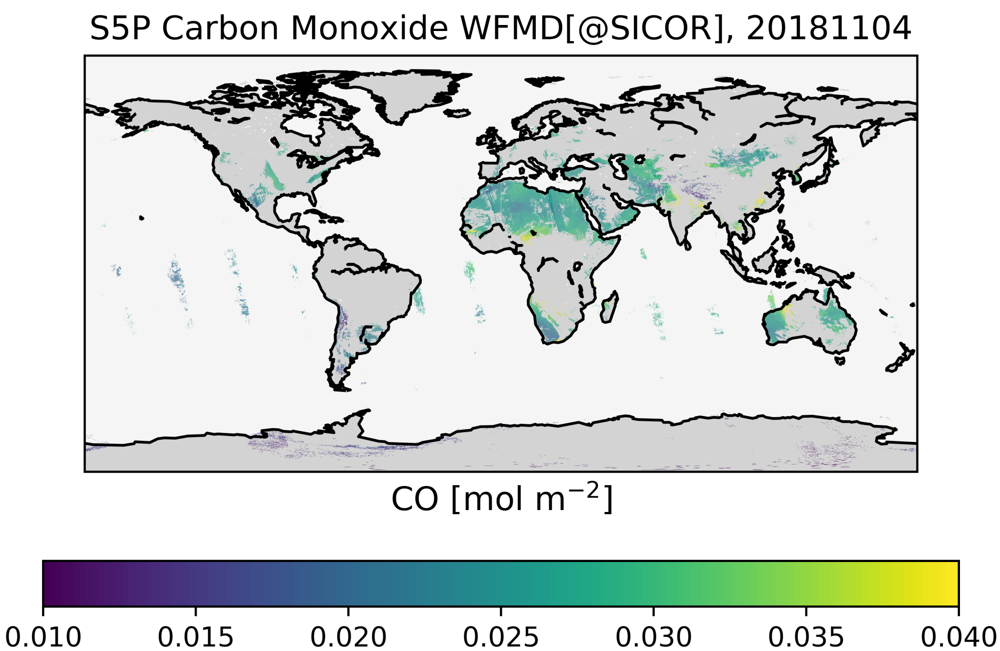
4.2.1 June 6th, 2018

In Figure 13 (a) the spreading of both the SICOR (in pink) and WFMD (in blue) measurements for June 6th can be observed. The TROPOMI/SICOR data shows more spreading and more points to be considered as outliers than TROPOMI/WFMD, but it also has almost 9 times as many retrievals (N = 3 585 415) passing the quality filters (N) than the WFMD data set has (N = 401 343).

From -90 to about -60 degrees of latitude both data sets show no retrievals at all. The WFMD data shows

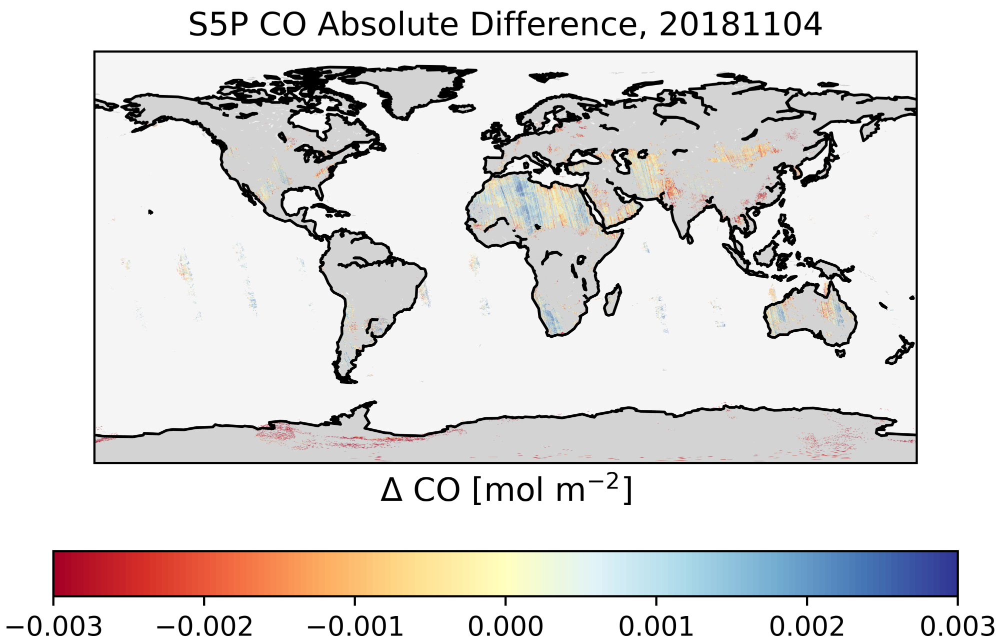


(a)

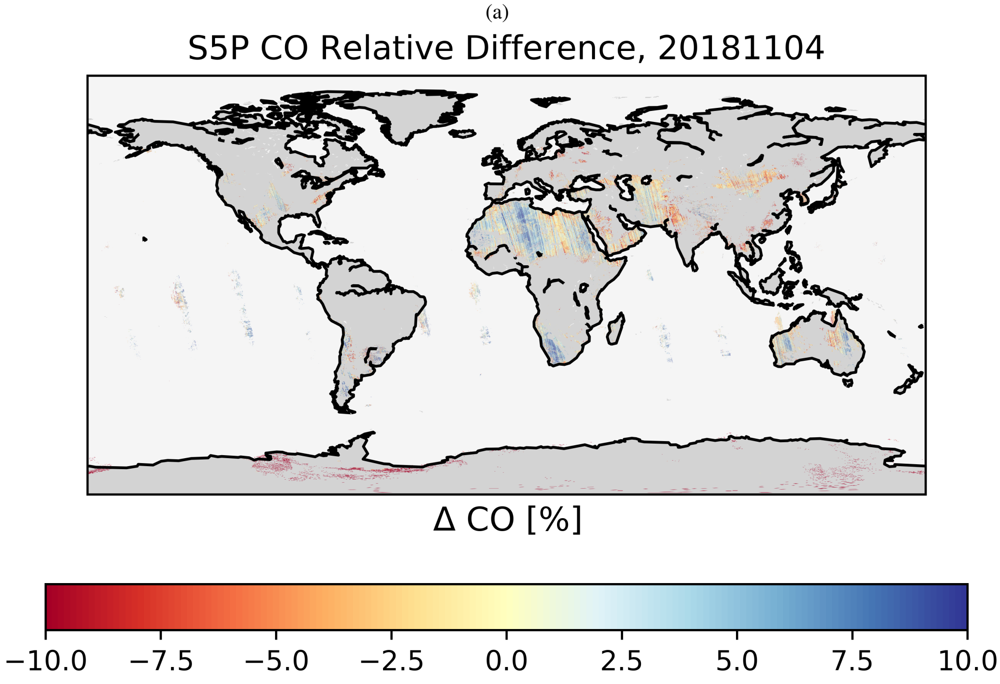


(b)

Figure 11: Collocated carbon monoxide retrievals in [mol m^{-2}] for November 4th, 2018, passing the suggested quality filter of the respective algorithms. (a) shows the TROPOMI/SICOR retrievals and (b) the TROPOMI/WFM-DOAS retrievals.

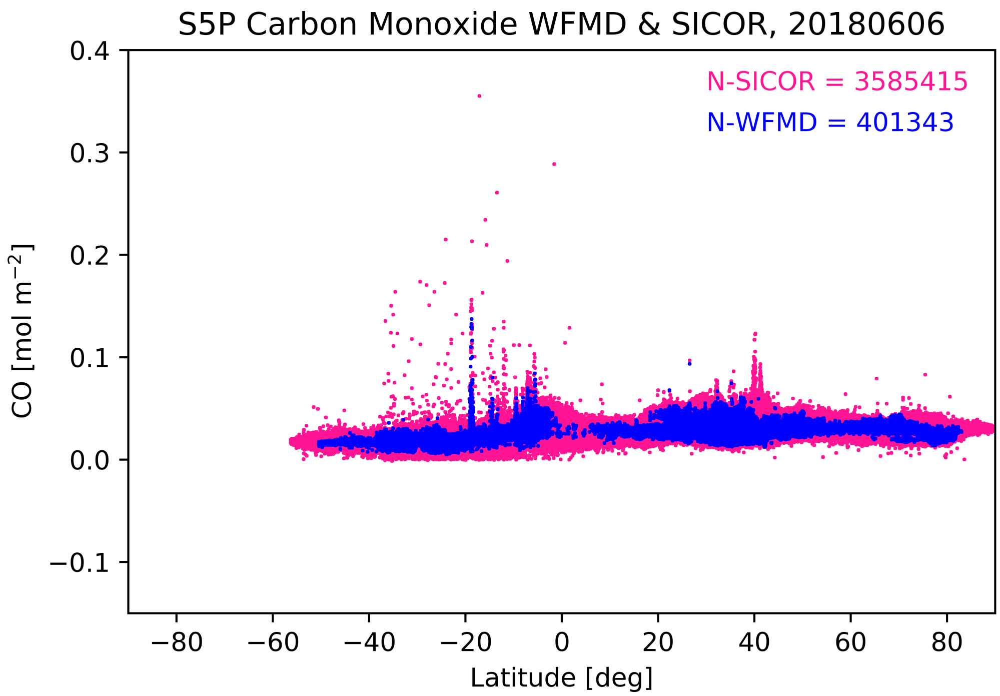


(a)

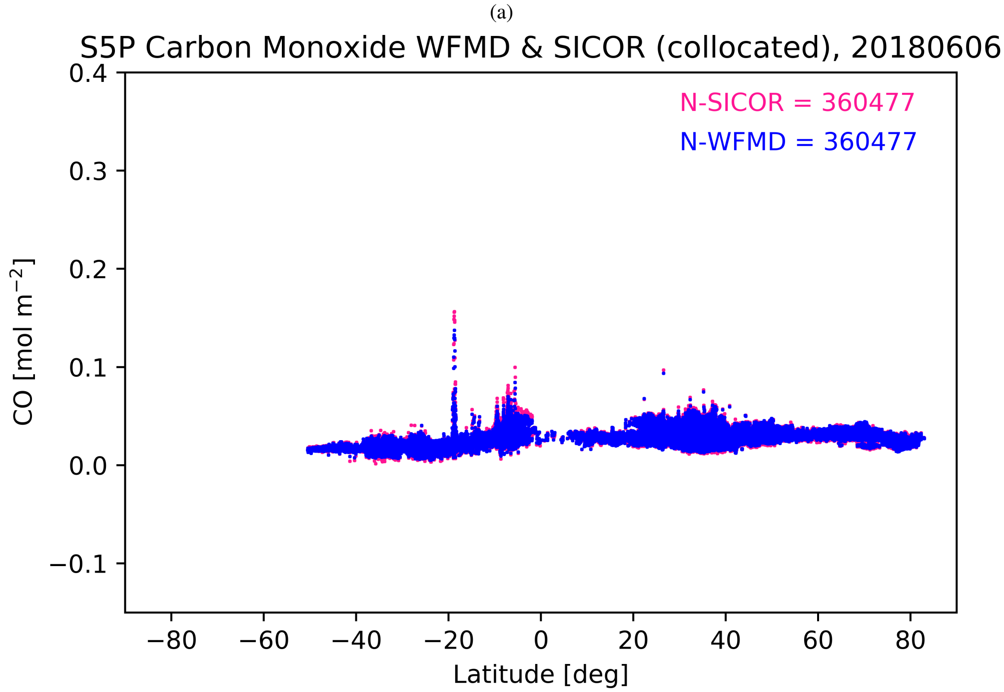


(b)

Figure 12: Difference collocated WFMD - collocated SICOR carbon monoxide retrievals for November 4th, 2018. (a) shows the absolute difference in [mol m^{-2}] and (b) the relative difference ((TROPOMI/WFMD - TROPOMI/SICOR) / TROPOMI/SICOR x 100 %).



(a)



(b)

Figure 13: S5-P carbon monoxide levels in [mol m^{-2}] with SICOR retrievals shown in pink and WFMD reading in blue, displayed by latitude for June 6^{th} 2018. Y-axis scale is chosen to visualize the data well, few retrievals may be beyond the axis limit. (a) shows all data of the respective products and (b) shows only the collocated retrievals.

a major peak at about -20 degrees of latitude, which is also seen in the SICOR data, and some minor peaks seen at about -18 and -10 degrees, again recognized in both data sets. Around 0 to 10 degrees of latitude, the WFMD data shows quite few retrievals and relatively low values compared to the rest of the plot. At about 40 to 42 degrees of latitude the SICOR data shows an elevation in data values.

Figure 13 (b) shows the collocated retrievals of Figure 13 (a). Most of the features and patterns seen in plot (a) are also visible in plot (b), but the single measurements with more extreme values, assumed to be outliers or wrong measurements, are no longer visible.

Figure 14 (a) shows the difference (WFMD-SICOR) for each measured point of the two collocated data sets. There is a fair amount of inconsistency among the data. Negative differences indicate higher retrievals in the SICOR data, while a positive difference indicates that the WFMD values would be higher. There are significant peaks to be observed in both directions, but the majority are negative, suggesting that the SICOR data retrievals are of slightly higher value.

Figure 14 (b) shows the correlation of the collocated data for June 6th, 2018. A high correlation (R = 0.96894) can be observed, but not the highest correlation among the three days.

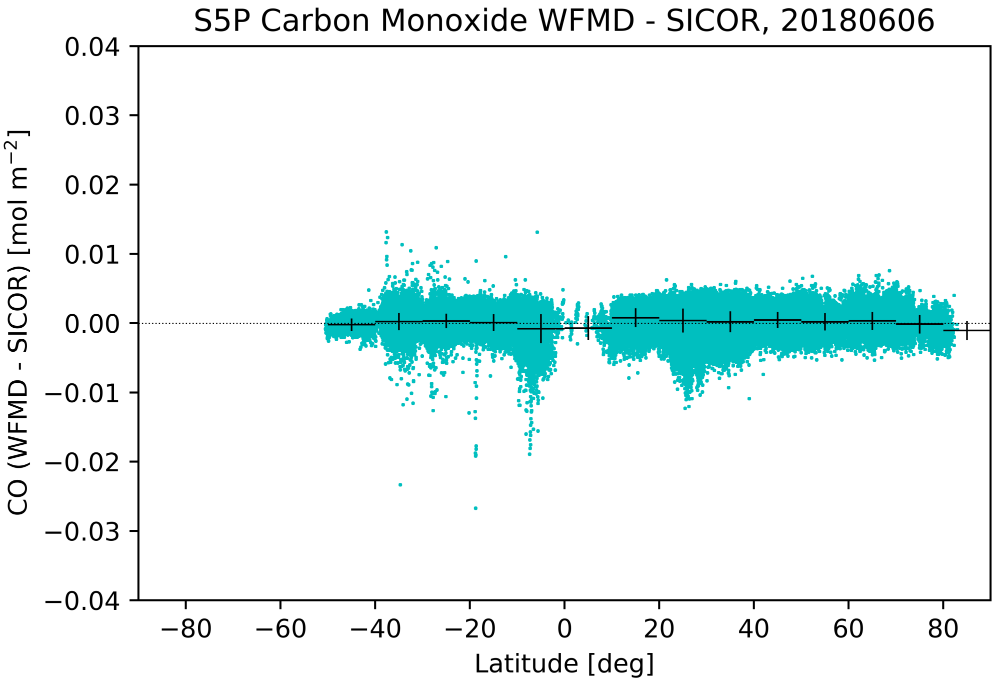
4.2.2 September 18th, 2018

In Figure 15 (a) the spreading of both the SICOR (in pink) and WFMD (in blue) measurements for September 18th, 2018 can be observed. Again, a greater spread can be seen in the SICOR data compared to the WFMD measurements. The number of SICOR measurements (N = 3 647 895) is more than 7 times the amount of WFMD measurements (N = 496 999), wherefore the greater spread is also reasonable. Retrievals for both data sets are available for almost all latitudes, showing some measurement-free areas close to the poles. For September 18th, as shown in these two plots, carbon monoxide levels are relatively low from -80 to -40 degrees of latitude, showing an enhancement between about -40 and 0 degrees, and again between 20 and 40 degrees. Smaller peaks can be seen at about -30, -20 and 35 degrees of latitude, which can be observed in both data sets. The SICOR also shows peaks at about -35, -10, 20 and 45 degrees.

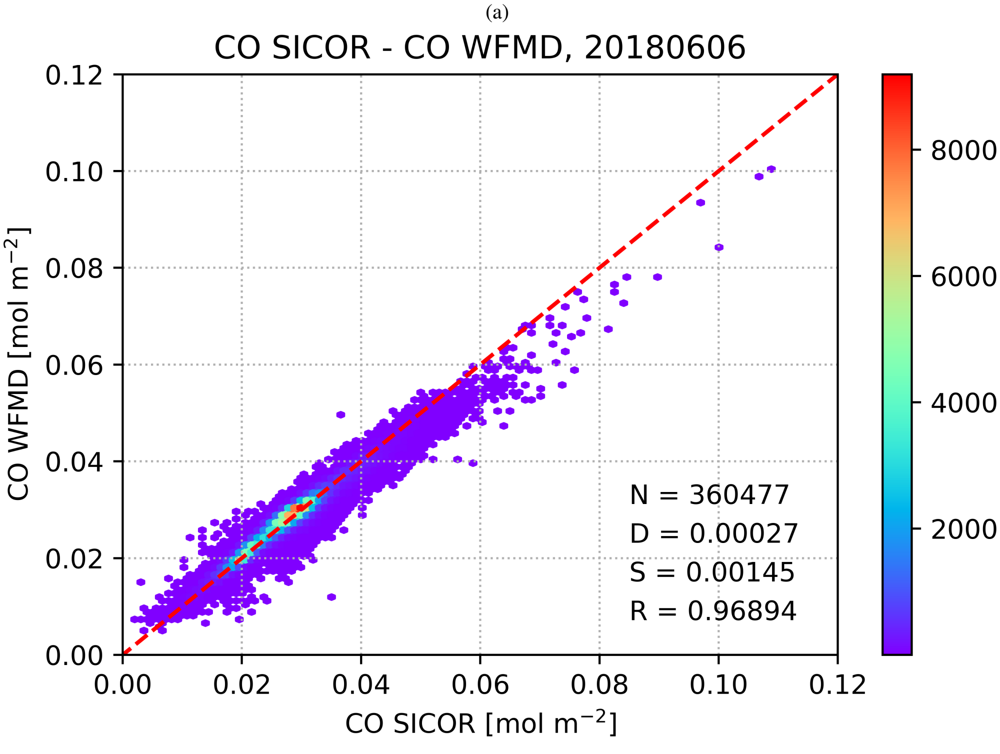
Figure 15 (b) shows the collocated retrievals of Figure 15 (a). The enhancements and patterns from Figure 15 (a) are visible, but can be recognized more clearly as single high and low measurements have been filtered out. However, the SICOR measurements seem to have a tendency to be a little higher in values that the WFMD retrievals are.

Figure 16 (a) shows the difference (WFMD-SICOR) for each measured point of the two collocated data sets. For -80 to -70 degrees of latitude the SICOR retrievals seem to be significantly higher than the WFMD retrievals. From -40 degrees onward the difference is more balanced in the positive and negative direction, however, it seems to be little more dominant towards the negative difference, supporting the observation from Figure 15 (b) where the SICOR data seems to have slightly higher measurements.

Figure 16 (b) shows the correlation of the collocated data for September 18th, 2018. The correlation is the best out of the three considered days (R = 0.97762) and the data spreads quite along the line of perfect

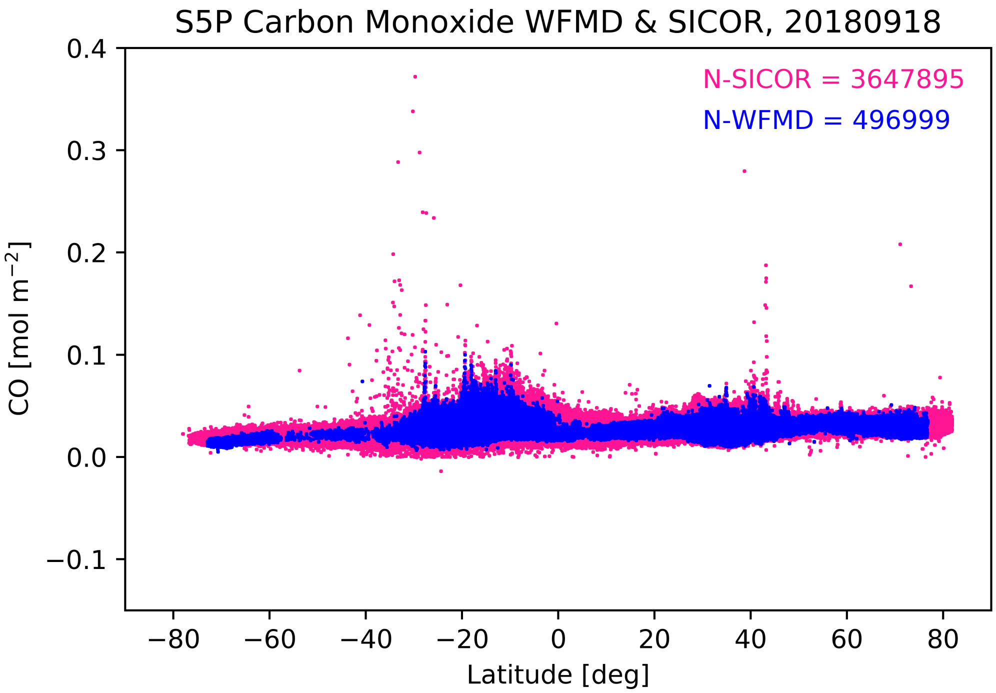


(a)

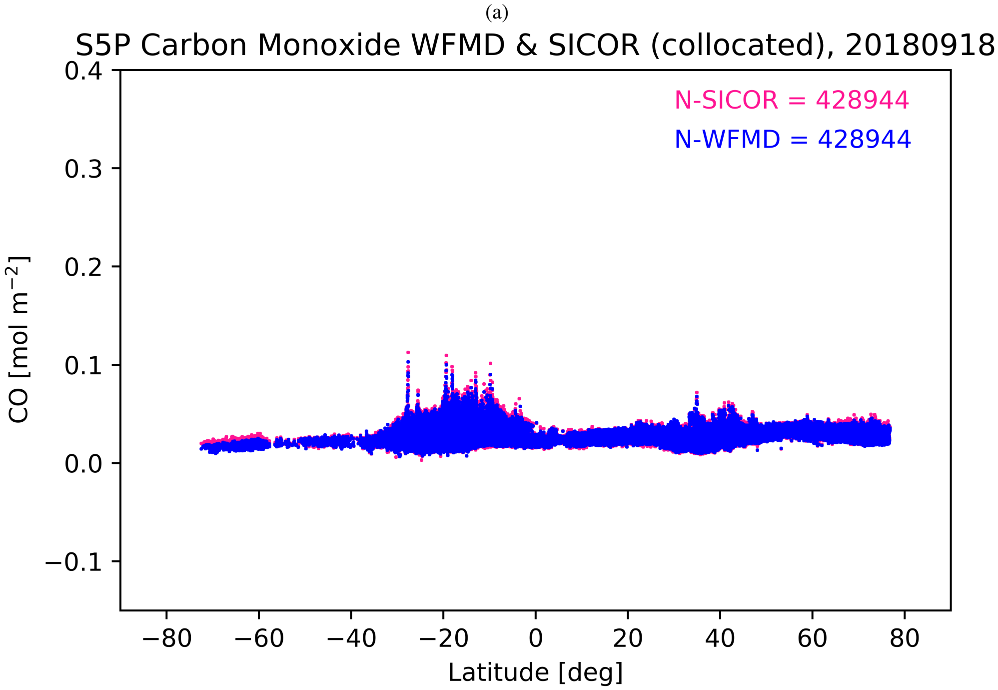


(b)

Figure 14: (a): Latitudinal distribution of the absolute difference in S5-P carbon monoxide levels (WFMD - SICOR) in [mol m^{-2}] for June 6th, 2018. Error-bars in x-direction show the mean and in y-direction the standard deviation. (b): Bivariate histogram of S5-P carbon monoxide levels in [mol m^{-2}] with SICOR retrievals on the x-axis and WFMD on the y-axis. N is the number of observations (collocated), D is the mean difference, S the standard deviation of the differences and R the correlation coefficient. Axes scales are chosen to visualize the data well, few retrievals may be beyond the shown areas.

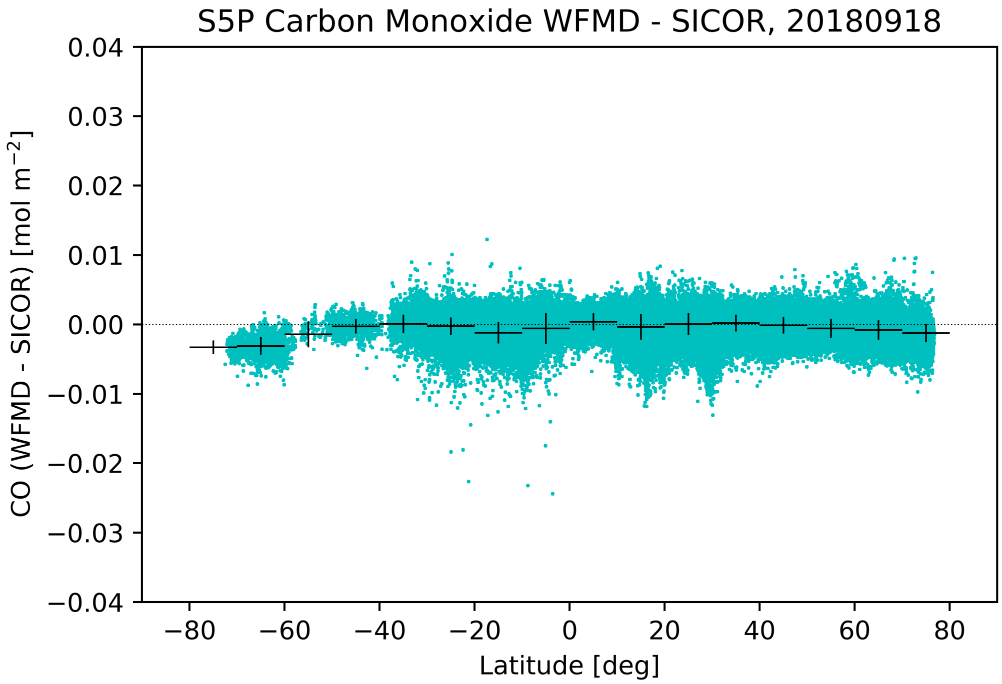


(a)

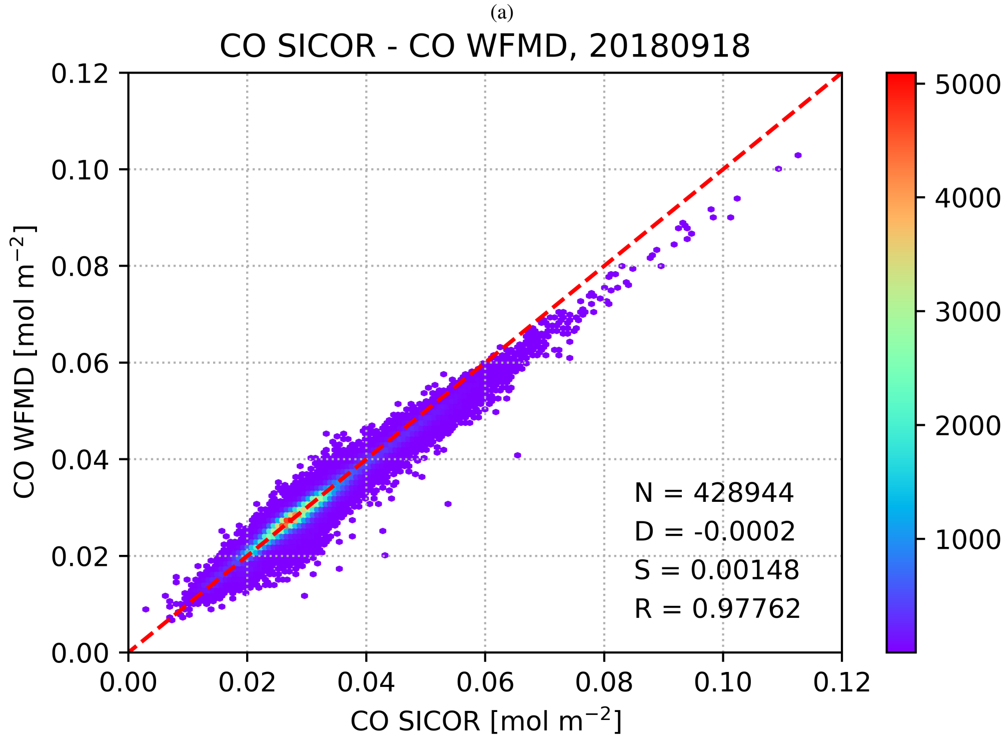


(b)

Figure 15: S5-P carbon monoxide levels in [mol m^{-2}] with SICOR retrievals shown in pink and WFMD reading in blue, displayed by latitude for September 18th 2018. Y-axis scale is chosen to visualize the data well, few retrievals may be beyond the axis limit. (a) shows all data of the respective products and (b) shows only the collocated retrievals.



(a)



(b)

Figure 16: (a): Latitudinal distribution of the absolute difference in S5-P carbon monoxide levels (WFMD - SICOR) in [mol m^{-2}] for September 18th, 2018. Error-bars in x-direction show the mean and in y-direction the standard deviation. (b): Bivariate histogram of S5-P carbon monoxide levels in [mol m^{-2}] with SICOR retrievals on the x-axis and WFMD on the y-axis. N is the number of observations (collocated), D is the mean difference, S the standard deviation of the differences and R the correlation coefficient. Axes scales are chosen to visualize the data well, few retrievals may be beyond the shown areas.

correlation, with the highest cluster of data being exactly on the red dashed line of perfect correlation, between carbon monoxide values of 0.02 and 0.035 [mol m^{-2}].

4.2.3 November 4th, 2018

In Figure 17 (a) the spreading of both the SICOR (in pink) and WFMD (in blue) measurements for November 4th, 2018 can be observed. Here it is most clearly visible that the SICOR data (N = 3 056 429) has more retrievals than the WFMD data (N = 397 286), in fact close to 8 times as many measurements as the WFMD data has. There is a range of about 20 degrees, from about -65 to -50 degrees of latitude, where there are no retrievals shown for the WFMD data, while SICOR measurements are available. There is a very clear peak in both data sets at about -35 degrees. In the WFMD data there are some smaller peaks to be seen at about 20 degrees and there are some further peaks in both data sets at about 15 degrees and multiple ones between 30 and 45 degrees.

Figure 17 (b) shows the collocated retrievals of Figure 17 (a). Again, the patterns visible for both data sets from from Figure 17 (a) are still visible here, and can be recognized more clearly. In addition, also here the SICOR data seems to have a tendency to be higher in values.

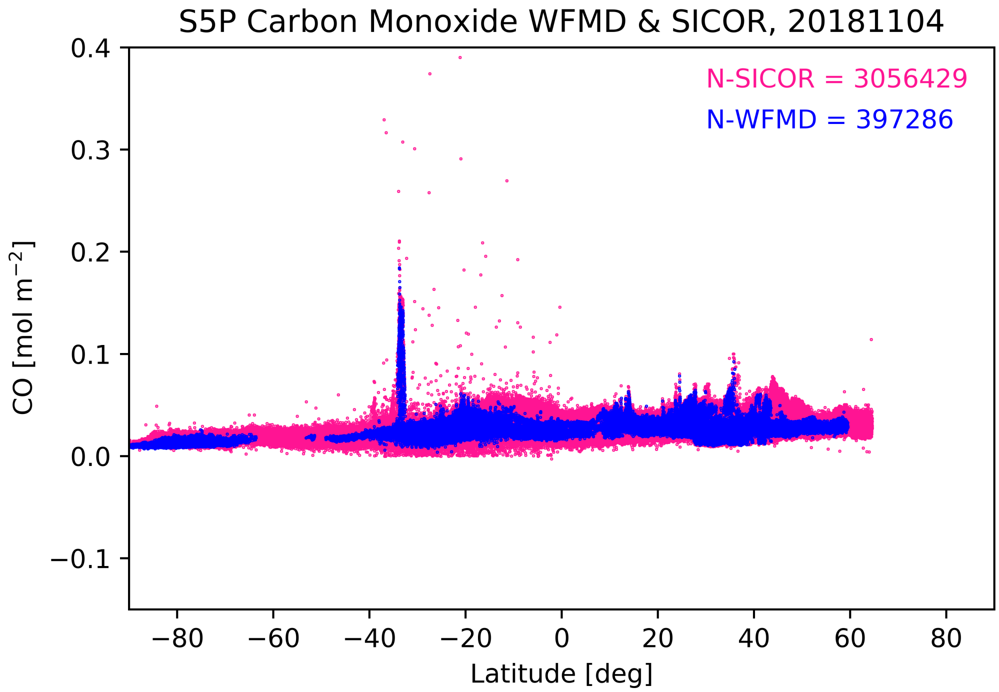
Figure 18 (a) shows the difference (WFMD-SICOR) for each measured point of the two collocated data sets, therefore the gap where WFMD did not have retrievals for this day is clearly visible. The difference confirms that SICOR has a tendency to be higher in values, with an extreme peak for this at about -35 degrees. For the rest of the data, although SICOR tends to be a little higher, the difference is quite balanced.

Figure 18 (b) shows the correlation of the collocated data for November 4th, 2018. The correlation is clearly visible and strong. Nevertheless, it is the lowest correlation that can be seen in the three observed days, with R = 0.96888. There is a cluster of data at CO levels of 0.02 - 0.03 [mol m^{-2}], and there is a larger spreading in the single measurements, which does roughly follow the line of perfect correlation.

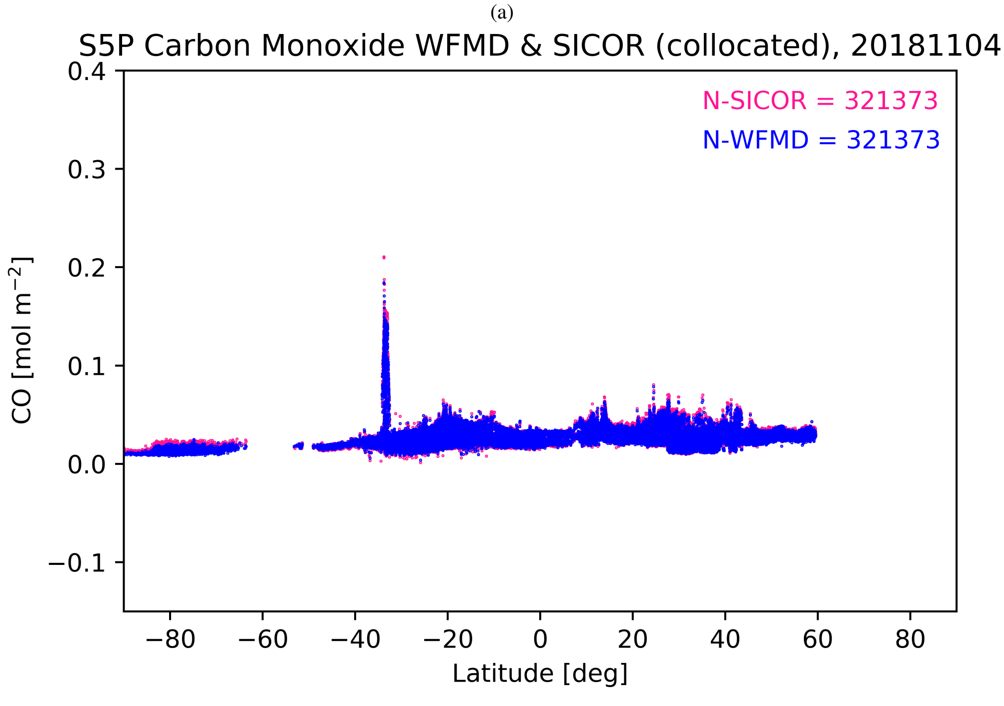
4.3 Comparison of statistic information

In order to have a better overview of the data, its basic statistic aspects are analyzed. To begin, the non-collocated and collocated data sets of the operational product are displayed, followed by the non-collocated and collocated data of the scientific data product, and ending with a statistic overview of the absolute difference of the collocated data.

Table 1 gives a general overview of the TROPOMI/SICOR data (the operational data product). The data set has about 3 million more retrievals per day, compared to the scientific TROPOMI/WFMD data set, with 3 585 415 retrievals on June 6th, 3 647 895 on September 18th and 3 056 429 on November 11th. The minimum values for all three days are negative values, which can happen when the values are very low and subject to noise. The maximum values are much higher than in the scientific data set for all three days, with June 6th reaching 0.80743 [mol m^{-2}], September 18th reaching 0.79583 [mol m^{-2}] and November 4th going as high as 1.68126 [mol m^{-2}]. The mean value are quite close to those of the WFMD data, with the

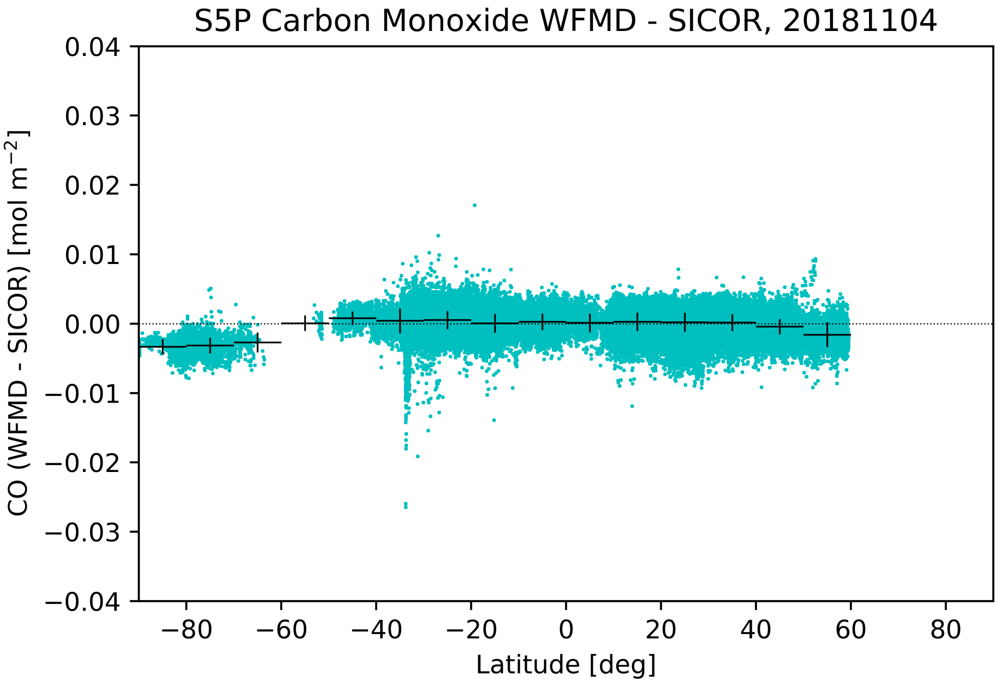


(a)

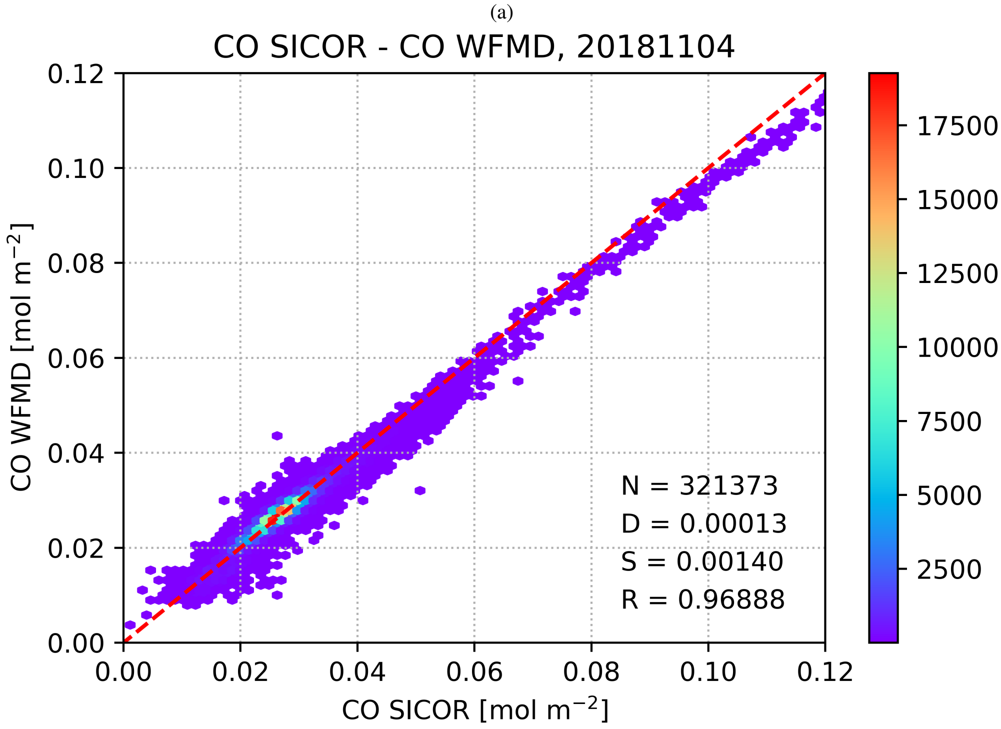


(b)

Figure 17: S5-P carbon monoxide levels in [mol m^{-2}] with SICOR retrievals shown in pink and WFMD reading in blue, displayed by latitude for November 4th 2018. Y-axis scale is chosen to visualize the data well, few retrievals may be beyond the axis limit. (a) shows all data of the respective products and (b) shows only the collocated retrievals.



(a)



(b)

Figure 18: (a): Latitudinal distribution of the absolute difference in S5-P carbon monoxide levels (WFMD - SICOR) in [mol m^{-2}] for November 4th, 2018. Error-bars in x-direction show the mean and in y-direction the standard deviation. (b): Bivariate histogram of S5-P carbon monoxide levels in [mol m^{-2}] with SICOR retrievals on the x-axis and WFMD on the y-axis. N is the number of observations (collocated), D is the mean difference, S the standard deviation of the differences and R the correlation coefficient. Axes scales are chosen to visualize the data well, few retrievals may be beyond the shown areas.

SICOR					
	N	min [mol m^{-2}]	max [mol m^{-2}]	mean [mol m^{-2}]	St.Dev. [mol m^{-2}]
20180606	3 585 415	-0.00118	0.80743	0.02698	0.00590
20180918	3 647 895	-0.01386	0.79583	0.02779	0.00697
20181104	3 056 429	-0.00276	1.68126	0.02501	0.00670

Table 1: Statistical overview of the SICOR data for 20180606, 20180918 and 20181104. N represents the number of measurements (passing the quality filter), min is the minimum value, max the maximum value, mean the mean of the data and St.Dev. the standard deviation.

difference between the mean values of the two data sets ranging from 0.00032 [mol m^{-2}] to 0.0012 [mol m^{-2}]. The difference in the standard deviations is also very small. With the operational data product having significantly more retrievals passing the quality filters, although the mean values and standard deviations are very close to those of the scientific data product, the range of values is also much larger, suggesting a much larger amount of outliers.

SICOR collocated					
	N	min [mol m^{-2}]	max [mol m^{-2}]	mean [mol m^{-2}]	St.Dev. [mol m^{-2}]
20180606	360 477	0.0015	0.15632	0.02711	0.00589
20180918	428 944	0.00293	0.11261	0.02914	0.00692
20181104	321 373	0.00115	0.21054	0.02681	0.00567

Table 2: Statistical overview of the collocated SICOR data for 20180606, 20180918 and 20181104. N represents the number of measurements (passing the quality filter), min is the minimum value, max the maximum value, mean the mean of the data and St.Dev. the standard deviation.

Table 2 is the statistic overview of the collocated TROPOMI/SICOR data. It therefore shows the same amount of retrievals as the collocated scientific data. The mimimum values are all lower than those of the collocated TROPOMI/WFMD data, but higher than those of the non-collocated SICOR data, suggesting that through the collocation process some outliers (especially the negative retrievals) get filtered out. The maximum values are only slightly higher than those of the collocated WFMD data, with a maximum difference of 0.02673 [mol m^{-2}] on June 6^{th}, suggesting here as well that the majority of outliers are filtered out through collocation. The mean values are slightly but not significantly higher than those of the non-collocated SICOR data and very close to the collocated WFMD values, with a maximum difference of 0.00028 [mol m^{-2}] on June 6 th. Also for the standard deviations there are no significant differences compared to the non-collocated data and the collocated WFMD data.

Table 3 gives a first overview of the TROPOMI/WFMD carbon monoxide measurements. A variation

WFMD					
	N	min [mol m^{-2}]	max [mol m^{-2}]	mean [mol m^{-2}]	St.Dev. [mol m^{-2}]
20180606	401 343	0.00502	0.1373	0.02730	0.00573
20180918	496 999	0.00504	0.1029	0.02870	0.00662
20181104	397 286	0.00372	0.18409	0.02626	0.00656

Table 3: Statistical overview of the WFMD data for 20180606, 20180918 and 20181104. N represents the number of measurements (passing the quality filter), min is the minimum value, max the maximum value, mean the mean of the data and St.Dev. the standard deviation.

of about 10^5 reading points can be observed among the three days. On June 6th and September 18th the minimum values are quite similar, with 0.00502 and 0.00504 [mol m^{-2}] respectively. The minimum value on November 4th is as low as 0.00372 [mol m^{-2}]. The maximum values of the considered June and September measurements are closer to each other compared to the November ones, with 0.1373 and 0.1029 [mol m^{-2}] respectively, while the November measurements going up to 0.18409 [mol m^{-2}] which could suggest the presence of outliers. The mean values of all three days are relatively similar, with a mean value of 0.02730 [mol m^{-2}] on June 6th, 0.02870 [mol m^{-2}] on September 18th and 0.02626 [mol m^{-2}] on November 4th.

Table 4 is an overview of the collocated TROPOMI/WFMD measurements, the data set composed of those TROPOMI/WFMD retrievals for which ground pixels there are also retrievals in the TROPOMI/SICOR data. As expected, the number of retrievals, N, is smaller for all three considered days, as it is a subset of the main data set. The minimum and maximum values have not changed compared to those of the complete WFMD data, with the exception of September 18th where the minimum value has increased to 0.00669 [mol m^{-2}]. The maximum values have not changed in comparison. The mean values and standard deviations have mostly changed very insignificantly, except for the standard deviation of November 4th, which has decreased by 0.00120 [mol m^{-2}].

WFMD collocated					
	N	min [mol m^{-2}]	max [mol m^{-2}]	mean [mol m^{-2}]	St.Dev. [mol m^{-2}]
20180606	360 477	0.00502	0.1373	0.02739	0.00565
20180918	428 944	0.00669	0.1029	0.02888	0.00648
20181104	321 373	0.00372	0.18409	0.02694	0.00536

Table 4: Statistical overview of the collocated WFMD data for 20180606, 20180918 and 20181104. N represents the number of measurements (passing the quality filter), min is the minimum value, max the maximum value, mean the mean of the data and St.Dev. the standard deviation.

Table 5 gives a further comparison between the collocated data of TROPOMI/WFMD and TROPOMI/

Absolute Difference (WFMD - SICOR)				
	mean [mol m^{-2}]	St.Dev. [mol m^{-2}]	min [mol m^{-2}]	max [mol m^{-2}]
20180606	0.00027	0.00145	-0.02673	0.01317
20180918	-0.0002	0.00148	-0.02440	0.01223
20181104	0.00013	0.00140	-0.02645	0.01706

Table 5: Statistical overview of the absolute difference between the collocated WFMD and the SICOR data for 20180606, 20180918 and 20181104. Mean is the mean value of the differences, min is the largest difference in negative direction (by how much SICOR is higher) and max the largest difference in positive direction (by how much WFMD is higher), and St.Dev. is the standard deviation of the absolute differences between the two data sets.

SICOR. The number of observations is the same as for Table 2 and Table 4. For the comparisons in this table, the absolute difference between the two data sets is considered, wherefore one should notice that the significance lies in the absolute value of the figures, and that it is insignificant if the referred values are positive of negative. The mean of the difference between the two data sets is quite low, with a maximum of 0.00027 [mol m^{-2}] on June 6^{th}, suggesting the retrievals have a tendency to not differ too much between the sets. The standard deviations of the differences are very close throughout all three of the considered days, suggesting consistency in the spread of the data. The absolute difference in the retrievals of the data sets ranges from 0.01317 [mol m^{-2}] to 0.02673 [mol m^{-2}].

5 Discussion

This section discusses the findings shown in the Section 4 (Results) in order to further place them in the context of the overall comparison of the two data products and to fulfil the aim of this thesis.

Looking at the collocated data sets of the three days that are under consideration for this comparison, from Figures 13, 15 and 17 (b), it can already be recognized that the products have a good correspondence. For all three days, the latitudinal distribution of the data shows good overlaps, indicating that enhancements as well as areas of low carbon monoxide levels are similar in both products. All three days also show that the SICOR retrievals can sometimes be distinguished below and above the WFM-DOAS retrievals, indicating smaller differences in the absolute measurements. No latitudinal shift can be distinguished from these plots, suggesting that the enhancements are seen at the same latitudes for both products, even if the retrieved carbon monoxide values differ a little.

The similarity of the two data products can further be observed from the global maps in Figures 7, 9 and 11 when looking at both panels (a) and (b). For all three days, both the areas of carbon monoxide enhancement and of lower carbon monoxide levels can be observed both from the collocated SICOR and WFMD retrieval maps. Clear enhancements, which are seen for both the collocated data products, are in central North America, the African west coast near Angola, in proximity of the Nile delta around Cairo in northern Africa, around the major cities in north India and Pakistan, in the Chinese industrial area, and Siberian Russia for June 6th, 2018. For September 18th, 2018, these are found on the Brazilian east coast, South Africa and southern Madagascar, and for November 4th, 2018, the major enhancements, all visible in both collocated data sets, are in northwest Australia, South Africa, central Africa (near the Central African Republic) and on the northwestern African coast near Senegal. On June 6th, 2018, areas of extremely low carbon monoxide retrievals, that are visible in both collocated data products, are found near Argentina, in South Africa and in central Australia. On September 18th, 2018, these are found to be in Greenland, the western United States and northern India (Himalaya and Tibet region), and on November 4th, 2018, they are in the Antarctic.

In comparing the differences, both by the global maps of relative and absolute differences shown in Figures 8, 10 and 12, as well as the latitudinal distribution of the differences shown in Figures 14, 16 and 18 (a), it can be seen that there is a rather steady difference, ranging about 0.0002 [mol m^{-2}] on average in both directions, which is about 10% of a difference (Table 5). For June 6th, 2018, and September 18th, 2018, the operational SICOR product has a tendency to be higher, while for November 4th, 2018, it seems to be more balance, with the scientific WFM-DOAS product being a little higher. Overall, as can be deduced from Table 5, the mean relative difference reaches a maximum of 1% among the three days, the relative standard deviation is found to be around 5%. The collocated data products are also be highly correlated, as can be read in Figures 14, 16 and 18 (b), with a correlation coefficient (R) of about 0.97 to 0.98 on all three days.

These outcomes and this comparison solely focus on the collocated data sets, which means that no information in relation to the quality in comparison can be given about the remaining retrievals. The number of collocated retrievals is 360 777 for Juneth, 428 944 for September 18th and 321 373 for November 4th. Nevertheless, the full operational SICOR data product includes 3 585 415 retrieval points for June 6th, 3 647 895 for September 18th and 3 056 429 for November 4th. This leaves about 3 million retrieval points per day of which the quality cannot be determined through this comparison. As the scientific WFM-DOAS product has less retrieval points, due to only retrieving cloud-free readings, it also leaves less unconsidered retrievals (between about 40 and 75 thousand), but nevertheless, these are existing data points contained in the data product, which are not considered in this comparison.

6 Conclusions

Carbon monoxide is a trace gas that has significant impacts on human health, air pollution and the global climate. Although it is not considered a greenhouse gas, it is able to alter the concentrations and emissions of other greenhouse gases, wherefore it is referred to as an indirect greenhouse gas.

There are currently two algorithms used to generate global carbon monoxide products from the TROPOspheric Monitoring Instrument (TROPOMI) onboard the Sentinel-5 Precursor satellite, namely the operational Copernicus Programme Shortwave Infrared CO Retrieval (SICOR) algorithm and the scientific Weighting Function Modified Differential Optical Absorption Spectroscopy (WFM-DOAS) algorithm developed at the Institute of Environmental Physics (IUP) at the University of Bremen.

The objective of this thesis is to compare these two data products. Global comparisons have been carried out for daily data on the days June 6th, September 18th and November 4th, 2018, in order to quantify the systematic differences. To achieve this, different aspects of the retrievals are considered, and retrievals passing the suggested quality measurements, referred to as quality flags, of the corresponding products have been considered. Global maps of the retrievals are generated and analyzed, as well as maps of absolute and relative differences, and the latitudinal distribution. The level of agreement and disagreement has been quantified by computing mean differences, standard deviations of differences and their linear correlation.

In the introduction, areas of significantly high and low carbon monoxide concentrations have been pointed out. A number of these have also been observed in the analyzed maps, among the three considered days. Regions of high concentrations that are also visible on the TROPOMI maps for June 6th, September 18th or November 4th, 2018, are located on the African west coast, northern India and Pakistan, the industrial area on the Chinese coast, central and southern Africa, Madagascar and Australia. Regions of very low carbon monoxide concentrations, where the analyzed days match with the global distributions described in the introduction, are in the Himalaya and Tibetan area and on the west coast of the United States.

For the three investigated days, the mean differences are very small, reaching a maximum of 1%, the standard deviation of the differences is below 10%, namely closer to 5%, and the linear correlation coefficient is about 0.97, indicating that the two data products have a high level of agreement. However, as shown by spatial maps and latitudinal difference plots, differences can be larger during certain times and at certain locations, although on average they are relatively low. There is no clear pattern for the distribution of the differences, but it can be observed that the SICOR product is higher at the southernmost latitudes for September 18th and November 4th, 2018. For June 6th, 2018, there is no collocated data available between -90°and -60°of latitude, wherefore no statement in this regard can be made for this day. These comparison results have been obtained after collocating the observations in order to be able to compute the difference for individual ground pixels, as the two data products show differences in their spatial coverage. This is due to the operational product aiming to also provide retrievals for partially cloudy scenes, whereas the scientific product limits its retrievals to cloud-free scenes. Due to this reason, the number of retrievals passing the quality measures for the operational product is roughly eight times larger than the scientific product, leaving the collocated data set to be a lot smaller than the size of the operational data product, and also slightly smaller than the full scientific product.

Possible extensions to this project could consist in analyzing data for more days, to obtain an understanding of the trends in global carbon monoxide levels, and to gain a better insight on the similarities and differences throughout a larger period of time. Furthermore, data for entire months or years could be plotted together to have a better coverage for comparison. Additionally, observations of particular areas and regions could be compared for smaller differences to become more visible and apparent.

References

Airbus Defence Space Dutch Technology (2016). Tropomi. https://www.airbusdefenceandspacenetherlands.nl/project/tropomi/.

Apituley, A., Padergnana, M., Sneep, M., Veefkind, J. P., Loyola, D., Landgraf, J., and Borsdorff, T. (2018). Sentinel-5 Precursor/TROPOMI Level 2 Product User Manual Carbon Monoxide.

Borsdorff, T., aan de Brugh, J., Hu, H., Aben, I., Hasekamp, O., and Landgraf, J. (2018a). Measuring Carbon Monoxide With TROPOMI: First Results and a Comparison With ECMWF-IFS Analysis Data. *Geophysical Research Letters*, 45(6):2826–2832.

Borsdorff, T., aan de Brugh, J., Hu, H., Hasekamp, O., Sussmann, R., Rettinger, M., Hase, F., Gross, J., Schneider, M., Garcia, O., Stremme, W., Grutter, M., Feist, D. G., Arnold, S. G., De Mazière, M., Kumar Sha, M., Pollard, D. F., Kiel, M., Roehl, C., Wennberg, P. O., Toon, G. C., and Landgraf, J. (2018b). Mapping Carbon Monoxide Pollution from Space Down to City Scales with Daily Global Coverage. *Atmospheric Measurement Techniques*, 11:5507–5518.

Buchwitz, M., Khlystova, I., Bovensmann, H., and Burrows, J. P. (2007). Three Years of Global Carbon Monoxide from SCIAMACHY: Comparison with MOPITT and First Results Related to the Detection of Enhanced CO Over cities. *Atmospheric Chemistry and Physics*, 7.

Burrows, J. P., Hölzle, E., Goede, A. P. H., V., H., and Fricke, W. (1995). SCIAMACHY — Scanning Imaging Absorption Spectrometer for Atmospheric Chartography. *Acta Astronautica*, 35(7):445 – 451. Earth Observation.

Daniel, J. S. and Solomon, S. (1998). On the Climate Forcing of Carbon Monoxide. *Journal of Geophysical Research: Atmospheres*, 103(D11):13249–13260.

ESA (2014). Earth Observation Programmes. http://www.esa.int/About_Us/Business_with_ESA/Business_Opportunities/Earth_Observation_Programmes.

ESA (2019a). Copernicus Overview. http://www.esa.int/Applications/Observing_the_Earth/Copernicus/Overview3.

ESA (2019b). Copernicus Overview. http://www.esa.int/Applications/Observing_the_Earth/Copernicus/Overview4.

Goldstein, M. (2009). Carbon Monoxide Poisoning. *Journal of emergency nursing: JEN : official publication of the Emergency Department Nurses Association*, 34:538–42.

Hille, K. (2015). NPP Mission Overview.

Holloway, T., Levi II, H., and Kasibhatla, P. (2000). Global Distribution of Carbon Monoxide. *Journal of Geophysical Research*, 105:12123–12148.

Khalil, M. A. K. and Rasmussen, R. A. (1990). The Global Cycle of Carbon Monoxide: Trends and Mass Balance. *Chemosphere*, 20(1):227 – 242.

Khlystova, I. G. (2010). *Analysis and Interpretation of Satellite Measurements in the Near-Infrared Spectral*

Region with the Focus on Carbon Monoxide. PhD thesis.

KNMI R&D Satellite Observations (2011). Tropomi. http://www.tropomi.eu.

Landgraf, J., aan de Brugh, J., Scheepmaker, R. A., Borsdorff, T., Houweling, S., and Hasekamp, O. P. (2018). Algorithm Theoretical Baseline Document for Sentinel-5 Precursor: Carbon Monoxide Total Column Retrieval.

Levrini, G. (2020). Private conversation about Copernicus program and Sentinel satellite missions. Copernicus Space Segment Programme Manager, European Space Agency.

Levy, R. J. (2015). Carbon Monoxide Pollution and Neurodevelopment: A Public Health Concern. *Neurotoxicology and teratology*, 49.

Murgatroyd, R. J. (2019). Troposphere. *AccessScience, McGraw-Hill Education*.

Raub, J., Organization, W. H., and on Chemical Safety, I. P. (1999). Carbon Monoxide.

Schneising, O. (2019). Product User Guide (PUG) TROPOMI/WFMD XCH_4 and XCO.

Schneising, O., Buchwitz, M., Reuter, M., Bovensmann, H., and Burrows, J. (2020). Severe Californian Wildfires in November 2018 Observed from Space: the Carbon monoxide Perspective. *Atmospheric Chemistry and Physics*, 20(6):3317–3332.

Schneising, O., Buchwitz, M., Reuter, M., Bovensmann, H., Burrows, J. P., Borsdorff, T., Deutscher, N. M., Feist, D. G., Griffith, D. W. T., Hase, F., Hermans, C., Iraci, L. T., Kivi, R., Landgraf, J., Morino, I., Notholt, J., Petri, C., Pollard, D. F., Roche, S., Shiomi, K., Strong, K., Sussmann, R., Velazco, V. A., Warneke, T., and Wunch, D. (2019). A Scientific Algorithm to Simultaneously Retrieve Carbon Monoxide and Methane from TROPOMI onboard Sentinel-5 Precursor. *Atmospheric Measurement Techniques*, 12(12):6771–6802.

Sentinel-5P Mission Performance Centre (2018). Sentinel-5P Level 1b Product Readme File.

Sinha, A. and Tuomi, R. (1996). A Comparison of Climate Forcings Due to Chlorofluorocarbons and Carbon Monoxide. *Geophysical Research Letters*, 23(1):65–68.

SRON (2019). Tropomi. https://www.sron.nl/tropomi-swir-monitoring.

Veefkind, J., Aben, I., McMullan, K., Förster, H., de Vries, J., Otter, G., Claas, J., Eskes, H., de Haan, J., Kleipool, Q., van Weele, M., Hasekamp, O., Hoogeveen, R., Landgraf, J., Snel, R., Tol, P., Ingmann, P., Voors, R., Kruizinga, B., Vink, R., Visser, H., and Levelt, P. (2012). TROPOMI on ESA Sentinel-5 Precursor: A GMES Mission for Global Observatios of the Atmospheric Composition for Climate, Air Quality and Ozone Layer Applications. *Remote Sensing of Environment*, 120:70 – 83. The Sentinel Missions - New Opportunities for Science.

Appendix

The following are each examples for one run, resulting in data, maps and plots for one day (June 6th, 2018). For further days some variables have been adjusted.

Appendix 1: Reading the Data and Creating Text Files with the Data for Further Analysis

```
from netCDF4 import Dataset
import numpy as np
import pandas as pd

## IUP Product (WFM-DOAS)

IUP_data = Dataset(r"C:ESACCI-GHG-L2-CH4-CO-TROPOMI-WFMD-20180606-fv1.nc",
    mode='r')
lat_IUP = IUP_data.variables['latitude'][:]
lon_IUP = IUP_data.variables['longitude'][:]
co_IUP = IUP_data.variables['co_column'][:]
grp_IUP = IUP_data.variables['ground_pixel'][:]
scl_IUP = IUP_data.variables['scanline'][:]
orb_IUP = IUP_data.variables['orbit_number'][:]
lat_c1_IUP = IUP_data.variables['latitude_corners'][:,0]
lat_c2_IUP = IUP_data.variables['latitude_corners'][:,1]
lat_c3_IUP = IUP_data.variables['latitude_corners'][:,2]
lat_c4_IUP = IUP_data.variables['latitude_corners'][:,3]
lon_c1_IUP = IUP_data.variables['longitude_corners'][:,0]
lon_c2_IUP = IUP_data.variables['longitude_corners'][:,1]
lon_c3_IUP = IUP_data.variables['longitude_corners'][:,2]
lon_c4_IUP = IUP_data.variables['longitude_corners'][:,3]

lat_IUP = np.array(lat_IUP)
lon_IUP = np.array(lon_IUP)
co_IUP = np.array(co_IUP)
grp_IUP = np.array(grp_IUP) -1   # to match with ESA
scl_IUP = np.array(scl_IUP) -1   # to match with ESA
lat_c1_IUP = np.array(lat_c1_IUP)
lat_c2_IUP = np.array(lat_c2_IUP)
lat_c3_IUP = np.array(lat_c3_IUP)
lat_c4_IUP = np.array(lat_c4_IUP)
lon_c1_IUP = np.array(lon_c1_IUP)
lon_c2_IUP = np.array(lon_c2_IUP)
```

```
lon_c3_IUP = np.array(lon_c3_IUP)
lon_c4_IUP = np.array(lon_c4_IUP)

n_IUP = co_IUP.size

S5P_20180606_co_IUP = open(r"C:S5P_20180606_co_IUP.txt", "w")
S5P_20180606_co_IUP.write("lat, lat_c1, lat_c2, lat_c3, lat_c4, lon, lon_c1, 
    lon_c2, lon_c3, lon_c4, co\n")

for j in np.arange(n_IUP):
  S5P_20180606_co_IUP.write("%10.5f %10.5f %10.5f %10.5f %10.5f %10.5f %10.5f 
    %10.5f %10.5f %10.5f %10.5f\n" % (lat_IUP[j], lat_c1_IUP[j], lat_c2_IUP[j
    ], lat_c3_IUP[j], lat_c4_IUP[j], lon_IUP[j], lon_c1_IUP[j], lon_c2_IUP[j],
     lon_c3_IUP[j], lon_c4_IUP[j], co_IUP[j]))

S5P_20180606_co_IUP.close()

## ESA Product (SICOR)

orbit01 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T002647_20180606T021015_03343_01_010202_
20190207T155143.nc", mode='r')

prod_01 = orbit01.groups['PRODUCT']
sdata_01 = prod_01.groups['SUPPORT_DATA']
geoloc_01 = sdata_01.groups['GEOLOCATIONS']

lat_0001 = prod_01.variables['latitude'][:]
lon_0001 = prod_01.variables['longitude'][:]
co_0001 = prod_01.variables['carbonmonoxide_total_column'][:]
qa_0001 = prod_01.variables['qa_value'][:]
grp_0001 = prod_01.variables['ground_pixel'][:]
scl_0001 = prod_01.variables['scanline'][:]
lat_c_0001 = geoloc_01.variables['latitude_bounds'][:]
lon_c_0001 = geoloc_01.variables['longitude_bounds'][:]


lat_001 = np.array(lat_0001)[0, :, :]
lon_001 = np.array(lon_0001)[0, :, :]
co_001 = np.array(co_0001)[0, :, :]
```

```python
qa_001 = np.array(qa_0001)[0, :, :]
grp_001_ = np.array(grp_0001)[:]
scl_001_ = np.array(scl_0001)[:]
lat_c1_001 = np.array(lat_c_0001)[0,:,:,0]
lat_c2_001 = np.array(lat_c_0001)[0,:,:,1]
lat_c3_001 = np.array(lat_c_0001)[0,:,:,2]
lat_c4_001 = np.array(lat_c_0001)[0,:,:,3]
lon_c1_001 = np.array(lon_c_0001)[0,:,:,0]
lon_c2_001 = np.array(lon_c_0001)[0,:,:,1]
lon_c3_001 = np.array(lon_c_0001)[0,:,:,2]
lon_c4_001 = np.array(lon_c_0001)[0,:,:,3]

n_grp_01 = grp_001_.size
n_scl_01 = scl_001_.size

scl_001 = qa_001 * 0
grp_001 = qa_001 * 0

for ii in range (n_scl_01):
  for jj in range (n_grp_01):
    scl_001[ii][jj] = scl_001_[ii]
    grp_001[ii][jj] = grp_001_[jj]

(i,j) = (qa_001 >= 0.5).nonzero()
lat_01 = lat_001[(i,j)]
lon_01 = lon_001[(i,j)]
co_01 = co_001[(i,j)]
grp_01 = grp_001[(i,j)]
scl_01 = scl_001[(i,j)]
lat_c1_01 = lat_c1_001[(i,j)]
lat_c2_01 = lat_c2_001[(i,j)]
lat_c3_01 = lat_c3_001[(i,j)]
lat_c4_01 = lat_c4_001[(i,j)]
lon_c1_01 = lon_c1_001[(i,j)]
lon_c2_01 = lon_c2_001[(i,j)]
lon_c3_01 = lon_c3_001[(i,j)]
lon_c4_01 = lon_c4_001[(i,j)]

lat_01 = pd.DataFrame(lat_01)
lon_01 = pd.DataFrame(lon_01)
```

```
co_01 = pd.DataFrame(co_01)
grp_01 = pd.DataFrame(grp_01)
scl_01 = pd.DataFrame(scl_01)
orb_01 = 0*lat_01 + 3343
lat_c1_01 = pd.DataFrame(lat_c1_01)
lat_c2_01 = pd.DataFrame(lat_c2_01)
lat_c3_01 = pd.DataFrame(lat_c3_01)
lat_c4_01 = pd.DataFrame(lat_c4_01)
lon_c1_01 = pd.DataFrame(lon_c1_01)
lon_c2_01 = pd.DataFrame(lon_c2_01)
lon_c3_01 = pd.DataFrame(lon_c3_01)
lon_c4_01 = pd.DataFrame(lon_c4_01)


orbit02 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T020817_20180606T035144_03344_01_010202_
20190207T155639.nc", mode='r')

prod_02 = orbit02.groups['PRODUCT']
sdata_02 = prod_02.groups['SUPPORT_DATA']
geoloc_02 = sdata_02.groups['GEOLOCATIONS']

lat_0002 = prod_02.variables['latitude'][:]
lon_0002 = prod_02.variables['longitude'][:]
co_0002 = prod_02.variables['carbonmonoxide_total_column'][:]
qa_0002 = prod_02.variables['qa_value'][:]
grp_0002 = prod_02.variables['ground_pixel'][:]
scl_0002 = prod_02.variables['scanline'][:]
lat_c_0002 = geoloc_02.variables['latitude_bounds'][:]
lon_c_0002 = geoloc_02.variables['longitude_bounds'][:]

lat_002 = np.array(lat_0002)[0, :, :]
lon_002 = np.array(lon_0002)[0, :, :]
co_002 = np.array(co_0002)[0, :, :]
qa_002 = np.array(qa_0002)[0, :, :]
grp_002_ = np.array(grp_0002)[:]
scl_002_ = np.array(scl_0002)[:]
lat_c1_002 = np.array(lat_c_0002)[0,:,:,0]
lat_c2_002 = np.array(lat_c_0002)[0,:,:,1]
lat_c3_002 = np.array(lat_c_0002)[0,:,:,2]
```

```
lat_c4_002 = np.array(lat_c_0002)[0,:,:,3]
lon_c1_002 = np.array(lon_c_0002)[0,:,:,0]
lon_c2_002 = np.array(lon_c_0002)[0,:,:,1]
lon_c3_002 = np.array(lon_c_0002)[0,:,:,2]
lon_c4_002 = np.array(lon_c_0002)[0,:,:,3]

n_grp_02 = grp_002_.size
n_scl_02 = scl_002_.size

scl_002 = qa_002 * 0
grp_002 = qa_002 * 0

for ii in range (n_scl_02):
  for jj in range (n_grp_02):
    scl_002[ii][jj] = scl_002_[ii]
    grp_002[ii][jj] = grp_002_[jj]

(i,j) = (qa_002 >= 0.5).nonzero()
lat_02 = lat_002[(i,j)]
lon_02 = lon_002[(i,j)]
co_02 = co_002[(i,j)]
grp_02 = grp_002[(i,j)]
scl_02 = scl_002[(i,j)]
lat_c1_02 = lat_c1_002[(i,j)]
lat_c2_02 = lat_c2_002[(i,j)]
lat_c3_02 = lat_c3_002[(i,j)]
lat_c4_02 = lat_c4_002[(i,j)]
lon_c1_02 = lon_c1_002[(i,j)]
lon_c2_02 = lon_c2_002[(i,j)]
lon_c3_02 = lon_c3_002[(i,j)]
lon_c4_02 = lon_c4_002[(i,j)]

lat_02 = pd.DataFrame(lat_02)
lon_02 = pd.DataFrame(lon_02)
co_02 = pd.DataFrame(co_02)
grp_02 = pd.DataFrame(grp_02)
scl_02 = pd.DataFrame(scl_02)
orb_02 = 0*lat_02 + 3344
lat_c1_02 = pd.DataFrame(lat_c1_02)
lat_c2_02 = pd.DataFrame(lat_c2_02)
```

```
lat_c3_02 = pd.DataFrame(lat_c3_02)
lat_c4_02 = pd.DataFrame(lat_c4_02)
lon_c1_02 = pd.DataFrame(lon_c1_02)
lon_c2_02 = pd.DataFrame(lon_c2_02)
lon_c3_02 = pd.DataFrame(lon_c3_02)
lon_c4_02 = pd.DataFrame(lon_c4_02)


orbit03 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T034946_20180606T053314_03345_01_010202_
20190207T160913.nc", mode='r')

prod_03 = orbit03.groups['PRODUCT']
sdata_03 = prod_03.groups['SUPPORT_DATA']
geoloc_03 = sdata_03.groups['GEOLOCATIONS']

lat_0003 = prod_03.variables['latitude'][:]
lon_0003 = prod_03.variables['longitude'][:]
co_0003 = prod_03.variables['carbonmonoxide_total_column'][:]
qa_0003 = prod_03.variables['qa_value'][:]
grp_0003 = prod_03.variables['ground_pixel'][:]
scl_0003 = prod_03.variables['scanline'][:]
lat_c_0003 = geoloc_03.variables['latitude_bounds'][:]
lon_c_0003 = geoloc_03.variables['longitude_bounds'][:]

lat_003 = np.array(lat_0003)[0, :, :]
lon_003 = np.array(lon_0003)[0, :, :]
co_003 = np.array(co_0003)[0, :, :]
qa_003 = np.array(qa_0003)[0, :, :]
grp_003_ = np.array(grp_0003)[:]
scl_003_ = np.array(scl_0003)[:]
lat_c1_003 = np.array(lat_c_0003)[0,:,:,0]
lat_c2_003 = np.array(lat_c_0003)[0,:,:,1]
lat_c3_003 = np.array(lat_c_0003)[0,:,:,2]
lat_c4_003 = np.array(lat_c_0003)[0,:,:,3]
lon_c1_003 = np.array(lon_c_0003)[0,:,:,0]
lon_c2_003 = np.array(lon_c_0003)[0,:,:,1]
lon_c3_003 = np.array(lon_c_0003)[0,:,:,2]
lon_c4_003 = np.array(lon_c_0003)[0,:,:,3]
```

```
n_grp_03 = grp_003_.size
n_scl_03 = scl_003_.size

scl_003 = qa_003 * 0
grp_003 = qa_003 * 0

for ii in range(n_scl_03):
  for jj in range(n_grp_03):
    scl_003[ii][jj] = scl_003_[ii]
    grp_003[ii][jj] = grp_003_[jj]

(i,j) = (qa_003 >= 0.5).nonzero()
lat_03 = lat_003[(i,j)]
lon_03 = lon_003[(i,j)]
co_03 = co_003[(i,j)]
grp_03 = grp_003[(i,j)]
scl_03 = scl_003[(i,j)]
lat_c1_03 = lat_c1_003[(i,j)]
lat_c2_03 = lat_c2_003[(i,j)]
lat_c3_03 = lat_c3_003[(i,j)]
lat_c4_03 = lat_c4_003[(i,j)]
lon_c1_03 = lon_c1_003[(i,j)]
lon_c2_03 = lon_c2_003[(i,j)]
lon_c3_03 = lon_c3_003[(i,j)]
lon_c4_03 = lon_c4_003[(i,j)]

lat_03 = pd.DataFrame(lat_03)
lon_03 = pd.DataFrame(lon_03)
co_03 = pd.DataFrame(co_03)
grp_03 = pd.DataFrame(grp_03)
scl_03 = pd.DataFrame(scl_03)
orb_03 = 0*lat_03 + 3345
lat_c1_03 = pd.DataFrame(lat_c1_03)
lat_c2_03 = pd.DataFrame(lat_c2_03)
lat_c3_03 = pd.DataFrame(lat_c3_03)
lat_c4_03 = pd.DataFrame(lat_c4_03)
lon_c1_03 = pd.DataFrame(lon_c1_03)
lon_c2_03 = pd.DataFrame(lon_c2_03)
lon_c3_03 = pd.DataFrame(lon_c3_03)
lon_c4_03 = pd.DataFrame(lon_c4_03)
```

```
orbit04 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T053116_20180606T071410_03346_01_010202_
20190207T162115.nc", mode='r')

prod_04 = orbit04.groups['PRODUCT']
sdata_04 = prod_04.groups['SUPPORT_DATA']
geoloc_04 = sdata_04.groups['GEOLOCATIONS']

lat_0004 = prod_04.variables['latitude'][:]
lon_0004 = prod_04.variables['longitude'][:]
co_0004 = prod_04.variables['carbonmonoxide_total_column'][:]
qa_0004 = prod_04.variables['qa_value'][:]
grp_0004 = prod_04.variables['ground_pixel'][:]
scl_0004 = prod_04.variables['scanline'][:]
lat_c_0004 = geoloc_04.variables['latitude_bounds'][:]
lon_c_0004 = geoloc_04.variables['longitude_bounds'][:]

lat_004 = np.array(lat_0004)[0, :, :]
lon_004 = np.array(lon_0004)[0, :, :]
co_004 = np.array(co_0004)[0, :, :]
qa_004 = np.array(qa_0004)[0, :, :]
grp_004_ = np.array(grp_0004)[:]
scl_004_ = np.array(scl_0004)[:]
lat_c1_004 = np.array(lat_c_0004)[0,:,:,0]
lat_c2_004 = np.array(lat_c_0004)[0,:,:,1]
lat_c3_004 = np.array(lat_c_0004)[0,:,:,2]
lat_c4_004 = np.array(lat_c_0004)[0,:,:,3]
lon_c1_004 = np.array(lon_c_0004)[0,:,:,0]
lon_c2_004 = np.array(lon_c_0004)[0,:,:,1]
lon_c3_004 = np.array(lon_c_0004)[0,:,:,2]
lon_c4_004 = np.array(lon_c_0004)[0,:,:,3]

n_grp_04 = grp_004_.size
n_scl_04 = scl_004_.size

scl_004 = qa_004 * 0
grp_004 = qa_004 * 0
```

```
for ii in range (n_scl_04):
  for jj in range (n_grp_04):
    scl_004[ii][jj] = scl_004_[ii]
    grp_004[ii][jj] = grp_004_[jj]

(i,j) = (qa_004 >= 0.5).nonzero()
lat_04 = lat_004[(i,j)]
lon_04 = lon_004[(i,j)]
co_04 = co_004[(i,j)]
grp_04 = grp_004[(i,j)]
scl_04 = scl_004[(i,j)]
lat_c1_04 = lat_c1_004[(i,j)]
lat_c2_04 = lat_c2_004[(i,j)]
lat_c3_04 = lat_c3_004[(i,j)]
lat_c4_04 = lat_c4_004[(i,j)]
lon_c1_04 = lon_c1_004[(i,j)]
lon_c2_04 = lon_c2_004[(i,j)]
lon_c3_04 = lon_c3_004[(i,j)]
lon_c4_04 = lon_c4_004[(i,j)]

lat_04 = pd.DataFrame(lat_04)
lon_04 = pd.DataFrame(lon_04)
co_04 = pd.DataFrame(co_04)
grp_04 = pd.DataFrame(grp_04)
scl_04 = pd.DataFrame(scl_04)
orb_04 = 0*lat_04 + 3346
lat_c1_04 = pd.DataFrame(lat_c1_04)
lat_c2_04 = pd.DataFrame(lat_c2_04)
lat_c3_04 = pd.DataFrame(lat_c3_04)
lat_c4_04 = pd.DataFrame(lat_c4_04)
lon_c1_04 = pd.DataFrame(lon_c1_04)
lon_c2_04 = pd.DataFrame(lon_c2_04)
lon_c3_04 = pd.DataFrame(lon_c3_04)
lon_c4_04 = pd.DataFrame(lon_c4_04)


orbit05 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T071246_20180606T085614_03347_01_010202_
20190207T162812.nc", mode='r')
```

```
prod_05 = orbit05.groups['PRODUCT']
sdata_05 = prod_05.groups['SUPPORT_DATA']
geoloc_05 = sdata_05.groups['GEOLOCATIONS']

lat_0005 = prod_05.variables['latitude'][:]
lon_0005 = prod_05.variables['longitude'][:]
co_0005 = prod_05.variables['carbonmonoxide_total_column'][:]
qa_0005 = prod_05.variables['qa_value'][:]
grp_0005 = prod_05.variables['ground_pixel'][:]
scl_0005 = prod_05.variables['scanline'][:]
lat_c_0005 = geoloc_05.variables['latitude_bounds'][:]
lon_c_0005 = geoloc_05.variables['longitude_bounds'][:]

lat_005 = np.array(lat_0005)[0, :, :]
lon_005 = np.array(lon_0005)[0, :, :]
co_005 = np.array(co_0005)[0, :, :]
qa_005 = np.array(qa_0005)[0, :, :]
grp_005_ = np.array(grp_0005)[:]
scl_005_ = np.array(scl_0005)[:]
lat_c1_005 = np.array(lat_c_0005)[0,:,:,0]
lat_c2_005 = np.array(lat_c_0005)[0,:,:,1]
lat_c3_005 = np.array(lat_c_0005)[0,:,:,2]
lat_c4_005 = np.array(lat_c_0005)[0,:,:,3]
lon_c1_005 = np.array(lon_c_0005)[0,:,:,0]
lon_c2_005 = np.array(lon_c_0005)[0,:,:,1]
lon_c3_005 = np.array(lon_c_0005)[0,:,:,2]
lon_c4_005 = np.array(lon_c_0005)[0,:,:,3]

n_grp_05 = grp_005_.size
n_scl_05 = scl_005_.size

scl_005 = qa_005 * 0
grp_005 = qa_005 * 0

for ii in range (n_scl_05):
  for jj in range (n_grp_05):
    scl_005[ii][jj] = scl_005_[ii]
    grp_005[ii][jj] = grp_005_[jj]

(i,j) = (qa_005 >= 0.5).nonzero()
```

```
lat_05 = lat_005[(i,j)]
lon_05 = lon_005[(i,j)]
co_05 = co_005[(i,j)]
grp_05 = grp_005[(i,j)]
scl_05 = scl_005[(i,j)]
lat_c1_05 = lat_c1_005[(i,j)]
lat_c2_05 = lat_c2_005[(i,j)]
lat_c3_05 = lat_c3_005[(i,j)]
lat_c4_05 = lat_c4_005[(i,j)]
lon_c1_05 = lon_c1_005[(i,j)]
lon_c2_05 = lon_c2_005[(i,j)]
lon_c3_05 = lon_c3_005[(i,j)]
lon_c4_05 = lon_c4_005[(i,j)]

lat_05 = pd.DataFrame(lat_05)
lon_05 = pd.DataFrame(lon_05)
co_05 = pd.DataFrame(co_05)
grp_05 = pd.DataFrame(grp_05)
scl_05 = pd.DataFrame(scl_05)
orb_05 = 0*lat_05 + 3347
lat_c1_05 = pd.DataFrame(lat_c1_05)
lat_c2_05 = pd.DataFrame(lat_c2_05)
lat_c3_05 = pd.DataFrame(lat_c3_05)
lat_c4_05 = pd.DataFrame(lat_c4_05)
lon_c1_05 = pd.DataFrame(lon_c1_05)
lon_c2_05 = pd.DataFrame(lon_c2_05)
lon_c3_05 = pd.DataFrame(lon_c3_05)
lon_c4_05 = pd.DataFrame(lon_c4_05)



orbit06 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T085416_20180606T103744_03348_01_010202_
20190207T164047.nc", mode='r')

prod_06 = orbit06.groups['PRODUCT']
sdata_06 = prod_06.groups['SUPPORT_DATA']
geoloc_06 = sdata_06.groups['GEOLOCATIONS']

lat_0006 = prod_06.variables['latitude'][:]
lon_0006 = prod_06.variables['longitude'][:]
```

```
co_0006 = prod_06.variables['carbonmonoxide_total_column'][:]
qa_0006 = prod_06.variables['qa_value'][:]
grp_0006 = prod_06.variables['ground_pixel'][:]
scl_0006 = prod_06.variables['scanline'][:]
lat_c_0006 = geoloc_06.variables['latitude_bounds'][:]
lon_c_0006 = geoloc_06.variables['longitude_bounds'][:]

lat_006 = np.array(lat_0006)[0, :, :]
lon_006 = np.array(lon_0006)[0, :, :]
co_006 = np.array(co_0006)[0, :, :]
qa_006 = np.array(qa_0006)[0, :, :]
grp_006_ = np.array(grp_0006)[:]
scl_006_ = np.array(scl_0006)[:]
lat_c1_006 = np.array(lat_c_0006)[0,:,:,0]
lat_c2_006 = np.array(lat_c_0006)[0,:,:,1]
lat_c3_006 = np.array(lat_c_0006)[0,:,:,2]
lat_c4_006 = np.array(lat_c_0006)[0,:,:,3]
lon_c1_006 = np.array(lon_c_0006)[0,:,:,0]
lon_c2_006 = np.array(lon_c_0006)[0,:,:,1]
lon_c3_006 = np.array(lon_c_0006)[0,:,:,2]
lon_c4_006 = np.array(lon_c_0006)[0,:,:,3]

n_grp_06 = grp_006_.size
n_scl_06 = scl_006_.size

scl_006 = qa_006 * 0
grp_006 = qa_006 * 0

for ii in range (n_scl_06):
  for jj in range (n_grp_06):
    scl_006[ii][jj] = scl_006_[ii]
    grp_006[ii][jj] = grp_006_[jj]

(i,j) = (qa_006 >= 0.5).nonzero()
lat_06 = lat_006[(i,j)]
lon_06 = lon_006[(i,j)]
co_06 = co_006[(i,j)]
grp_06 = grp_006[(i,j)]
scl_06 = scl_006[(i,j)]
lat_c1_06 = lat_c1_006[(i,j)]
```

```
lat_c2_06 = lat_c2_006[(i,j)]
lat_c3_06 = lat_c3_006[(i,j)]
lat_c4_06 = lat_c4_006[(i,j)]
lon_c1_06 = lon_c1_006[(i,j)]
lon_c2_06 = lon_c2_006[(i,j)]
lon_c3_06 = lon_c3_006[(i,j)]
lon_c4_06 = lon_c4_006[(i,j)]

lat_06 = pd.DataFrame(lat_06)
lon_06 = pd.DataFrame(lon_06)
co_06 = pd.DataFrame(co_06)
grp_06 = pd.DataFrame(grp_06)
scl_06 = pd.DataFrame(scl_06)
orb_06 = 0*lat_06 + 3348
lat_c1_06 = pd.DataFrame(lat_c1_06)
lat_c2_06 = pd.DataFrame(lat_c2_06)
lat_c3_06 = pd.DataFrame(lat_c3_06)
lat_c4_06 = pd.DataFrame(lat_c4_06)
lon_c1_06 = pd.DataFrame(lon_c1_06)
lon_c2_06 = pd.DataFrame(lon_c2_06)
lon_c3_06 = pd.DataFrame(lon_c3_06)
lon_c4_06 = pd.DataFrame(lon_c4_06)


orbit07 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T103546_20180606T121914_03349_01_010202_
20190207T165649.nc", mode='r')

prod_07 = orbit07.groups['PRODUCT']
sdata_07 = prod_07.groups['SUPPORT_DATA']
geoloc_07 = sdata_07.groups['GEOLOCATIONS']

lat_0007 = prod_07.variables['latitude'][:]
lon_0007 = prod_07.variables['longitude'][:]
co_0007 = prod_07.variables['carbonmonoxide_total_column'][:]
qa_0007 = prod_07.variables['qa_value'][:]
grp_0007 = prod_07.variables['ground_pixel'][:]
scl_0007 = prod_07.variables['scanline'][:]
lat_c_0007 = geoloc_07.variables['latitude_bounds'][:]
lon_c_0007 = geoloc_07.variables['longitude_bounds'][:]
```

```
lat_007 = np.array(lat_0007)[0, :, :]
lon_007 = np.array(lon_0007)[0, :, :]
co_007 = np.array(co_0007)[0, :, :]
qa_007 = np.array(qa_0007)[0, :, :]
grp_007_ = np.array(grp_0007)[:]
scl_007_ = np.array(scl_0007)[:]
lat_c1_007 = np.array(lat_c_0007)[0,:,:,0]
lat_c2_007 = np.array(lat_c_0007)[0,:,:,1]
lat_c3_007 = np.array(lat_c_0007)[0,:,:,2]
lat_c4_007 = np.array(lat_c_0007)[0,:,:,3]
lon_c1_007 = np.array(lon_c_0007)[0,:,:,0]
lon_c2_007 = np.array(lon_c_0007)[0,:,:,1]
lon_c3_007 = np.array(lon_c_0007)[0,:,:,2]
lon_c4_007 = np.array(lon_c_0007)[0,:,:,3]

n_grp_07 = grp_007_.size
n_scl_07 = scl_007_.size

scl_007 = qa_007 * 0
grp_007 = qa_007 * 0

for ii in range (n_scl_07):
  for jj in range (n_grp_07):
    scl_007[ii][jj] = scl_007_[ii]
    grp_007[ii][jj] = grp_007_[jj]

(i,j) = (qa_007 >= 0.5).nonzero()
lat_07 = lat_007[(i,j)]
lon_07 = lon_007[(i,j)]
co_07 = co_007[(i,j)]
grp_07 = grp_007[(i,j)]
scl_07 = scl_007[(i,j)]
lat_c1_07 = lat_c1_007[(i,j)]
lat_c2_07 = lat_c2_007[(i,j)]
lat_c3_07 = lat_c3_007[(i,j)]
lat_c4_07 = lat_c4_007[(i,j)]
lon_c1_07 = lon_c1_007[(i,j)]
lon_c2_07 = lon_c2_007[(i,j)]
lon_c3_07 = lon_c3_007[(i,j)]
```

```
lon_c4_07 = lon_c4_007[(i,j)]

lat_07 = pd.DataFrame(lat_07)
lon_07 = pd.DataFrame(lon_07)
co_07 = pd.DataFrame(co_07)
grp_07 = pd.DataFrame(grp_07)
scl_07 = pd.DataFrame(scl_07)
orb_07 = 0*lat_07 + 3349
lat_c1_07 = pd.DataFrame(lat_c1_07)
lat_c2_07 = pd.DataFrame(lat_c2_07)
lat_c3_07 = pd.DataFrame(lat_c3_07)
lat_c4_07 = pd.DataFrame(lat_c4_07)
lon_c1_07 = pd.DataFrame(lon_c1_07)
lon_c2_07 = pd.DataFrame(lon_c2_07)
lon_c3_07 = pd.DataFrame(lon_c3_07)
lon_c4_07 = pd.DataFrame(lon_c4_07)



orbit08 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T121716_20180606T140044_03350_01_010202_
20190207T171959.nc", mode='r')

prod_08 = orbit08.groups['PRODUCT']
sdata_08 = prod_08.groups['SUPPORT_DATA']
geoloc_08 = sdata_08.groups['GEOLOCATIONS']

lat_0008 = prod_08.variables['latitude'][:]
lon_0008 = prod_08.variables['longitude'][:]
co_0008 = prod_08.variables['carbonmonoxide_total_column'][:]
qa_0008 = prod_08.variables['qa_value'][:]
grp_0008 = prod_08.variables['ground_pixel'][:]
scl_0008 = prod_08.variables['scanline'][:]
lat_c_0008 = geoloc_08.variables['latitude_bounds'][:]
lon_c_0008 = geoloc_08.variables['longitude_bounds'][:]

lat_008 = np.array(lat_0008)[0, :, :]
lon_008 = np.array(lon_0008)[0, :, :]
co_008 = np.array(co_0008)[0, :, :]
qa_008 = np.array(qa_0008)[0, :, :]
grp_008_ = np.array(grp_0008)[:]
```

```
scl_008_ = np.array(scl_0008)[:]
lat_c1_008 = np.array(lat_c_0008)[0,:,:,0]
lat_c2_008 = np.array(lat_c_0008)[0,:,:,1]
lat_c3_008 = np.array(lat_c_0008)[0,:,:,2]
lat_c4_008 = np.array(lat_c_0008)[0,:,:,3]
lon_c1_008 = np.array(lon_c_0008)[0,:,:,0]
lon_c2_008 = np.array(lon_c_0008)[0,:,:,1]
lon_c3_008 = np.array(lon_c_0008)[0,:,:,2]
lon_c4_008 = np.array(lon_c_0008)[0,:,:,3]

n_grp_08 = grp_008_.size
n_scl_08 = scl_008_.size

scl_008 = qa_008 * 0
grp_008 = qa_008 * 0

for ii in range (n_scl_08):
  for jj in range (n_grp_08):
    scl_008[ii][jj] = scl_008_[ii]
    grp_008[ii][jj] = grp_008_[jj]

(i,j) = (qa_008 >= 0.5).nonzero()
lat_08 = lat_008[(i,j)]
lon_08 = lon_008[(i,j)]
co_08 = co_008[(i,j)]
grp_08 = grp_008[(i,j)]
scl_08 = scl_008[(i,j)]
lat_c1_08 = lat_c1_008[(i,j)]
lat_c2_08 = lat_c2_008[(i,j)]
lat_c3_08 = lat_c3_008[(i,j)]
lat_c4_08 = lat_c4_008[(i,j)]
lon_c1_08 = lon_c1_008[(i,j)]
lon_c2_08 = lon_c2_008[(i,j)]
lon_c3_08 = lon_c3_008[(i,j)]
lon_c4_08 = lon_c4_008[(i,j)]

lat_08 = pd.DataFrame(lat_08)
lon_08 = pd.DataFrame(lon_08)
co_08 = pd.DataFrame(co_08)
grp_08 = pd.DataFrame(grp_08)
```

```
scl_08 = pd.DataFrame(scl_08)
orb_08 = 0*lat_08 + 3350
lat_c1_08 = pd.DataFrame(lat_c1_08)
lat_c2_08 = pd.DataFrame(lat_c2_08)
lat_c3_08 = pd.DataFrame(lat_c3_08)
lat_c4_08 = pd.DataFrame(lat_c4_08)
lon_c1_08 = pd.DataFrame(lon_c1_08)
lon_c2_08 = pd.DataFrame(lon_c2_08)
lon_c3_08 = pd.DataFrame(lon_c3_08)
lon_c4_08 = pd.DataFrame(lon_c4_08)


orbit09 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T143822_20180606T145537_03351_01_010202_
20190207T172910.nc", mode='r')

prod_09 = orbit09.groups['PRODUCT']
sdata_09 = prod_09.groups['SUPPORT_DATA']
geoloc_09 = sdata_09.groups['GEOLOCATIONS']

lat_0009 = prod_09.variables['latitude'][:]
lon_0009 = prod_09.variables['longitude'][:]
co_0009 = prod_09.variables['carbonmonoxide_total_column'][:]
qa_0009 = prod_09.variables['qa_value'][:]
grp_0009 = prod_09.variables['ground_pixel'][:]
scl_0009 = prod_09.variables['scanline'][:]
lat_c_0009 = geoloc_09.variables['latitude_bounds'][:]
lon_c_0009 = geoloc_09.variables['longitude_bounds'][:]

lat_009 = np.array(lat_0009)[0, :, :]
lon_009 = np.array(lon_0009)[0, :, :]
co_009 = np.array(co_0009)[0, :, :]
qa_009 = np.array(qa_0009)[0, :, :]
grp_009_ = np.array(grp_0009)[:]
scl_009_ = np.array(scl_0009)[:]
lat_c1_009 = np.array(lat_c_0009)[0,:,:,0]
lat_c2_009 = np.array(lat_c_0009)[0,:,:,1]
lat_c3_009 = np.array(lat_c_0009)[0,:,:,2]
lat_c4_009 = np.array(lat_c_0009)[0,:,:,3]
lon_c1_009 = np.array(lon_c_0009)[0,:,:,0]
```

```
lon_c2_009 = np.array(lon_c_0009)[0,:,:,1]
lon_c3_009 = np.array(lon_c_0009)[0,:,:,2]
lon_c4_009 = np.array(lon_c_0009)[0,:,:,3]

n_grp_09 = grp_009_.size
n_scl_09 = scl_009_.size

scl_009 = qa_009 * 0
grp_009 = qa_009 * 0

for ii in range (n_scl_09):
  for jj in range (n_grp_09):
    scl_009[ii][jj] = scl_009_[ii]
    grp_009[ii][jj] = grp_009_[jj]

(i,j) = (qa_009 >= 0.5).nonzero()
lat_09 = lat_009[(i,j)]
lon_09 = lon_009[(i,j)]
co_09 = co_009[(i,j)]
grp_09 = grp_009[(i,j)]
scl_09 = scl_009[(i,j)]
lat_c1_09 = lat_c1_009[(i,j)]
lat_c2_09 = lat_c2_009[(i,j)]
lat_c3_09 = lat_c3_009[(i,j)]
lat_c4_09 = lat_c4_009[(i,j)]
lon_c1_09 = lon_c1_009[(i,j)]
lon_c2_09 = lon_c2_009[(i,j)]
lon_c3_09 = lon_c3_009[(i,j)]
lon_c4_09 = lon_c4_009[(i,j)]

lat_09 = pd.DataFrame(lat_09)
lon_09 = pd.DataFrame(lon_09)
co_09 = pd.DataFrame(co_09)
grp_09 = pd.DataFrame(grp_09)
scl_09 = pd.DataFrame(scl_09)
orb_09 = 0*lat_09 + 3351
lat_c1_09 = pd.DataFrame(lat_c1_09)
lat_c2_09 = pd.DataFrame(lat_c2_09)
lat_c3_09 = pd.DataFrame(lat_c3_09)
lat_c4_09 = pd.DataFrame(lat_c4_09)
```

```
lon_c1_09 = pd.DataFrame(lon_c1_09)
lon_c2_09 = pd.DataFrame(lon_c2_09)
lon_c3_09 = pd.DataFrame(lon_c3_09)
lon_c4_09 = pd.DataFrame(lon_c4_09)


orbit10 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T154016_20180606T172340_03352_01_010202_
20190207T173851.nc", mode='r')

prod_10 = orbit10.groups['PRODUCT']
sdata_10 = prod_10.groups['SUPPORT_DATA']
geoloc_10 = sdata_10.groups['GEOLOCATIONS']

lat_0010 = prod_10.variables['latitude'][:]
lon_0010 = prod_10.variables['longitude'][:]
co_0010 = prod_10.variables['carbonmonoxide_total_column'][:]
qa_0010 = prod_10.variables['qa_value'][:]
grp_0010 = prod_10.variables['ground_pixel'][:]
scl_0010 = prod_10.variables['scanline'][:]
lat_c_0010 = geoloc_10.variables['latitude_bounds'][:]
lon_c_0010 = geoloc_10.variables['longitude_bounds'][:]

lat_010 = np.array(lat_0010)[0, :, :]
lon_010 = np.array(lon_0010)[0, :, :]
co_010 = np.array(co_0010)[0, :, :]
qa_010 = np.array(qa_0010)[0, :, :]
grp_010_ = np.array(grp_0010)[:]
scl_010_ = np.array(scl_0010)[:]
lat_c1_010 = np.array(lat_c_0010)[0,:,:,0]
lat_c2_010 = np.array(lat_c_0010)[0,:,:,1]
lat_c3_010 = np.array(lat_c_0010)[0,:,:,2]
lat_c4_010 = np.array(lat_c_0010)[0,:,:,3]
lon_c1_010 = np.array(lon_c_0010)[0,:,:,0]
lon_c2_010 = np.array(lon_c_0010)[0,:,:,1]
lon_c3_010 = np.array(lon_c_0010)[0,:,:,2]
lon_c4_010 = np.array(lon_c_0010)[0,:,:,3]

n_grp_10 = grp_010_.size
n_scl_10 = scl_010_.size
```

```python
scl_010 = qa_010 * 0
grp_010 = qa_010 * 0

for ii in range (n_scl_10):
  for jj in range (n_grp_10):
    scl_010[ii][jj] = scl_010_[ii]
    grp_010[ii][jj] = grp_010_[jj]

(i,j) = (qa_010 >= 0.5).nonzero()
lat_10 = lat_010[(i,j)]
lon_10 = lon_010[(i,j)]
co_10 = co_010[(i,j)]
grp_10 = grp_010[(i,j)]
scl_10 = scl_010[(i,j)]
lat_c1_10 = lat_c1_010[(i,j)]
lat_c2_10 = lat_c2_010[(i,j)]
lat_c3_10 = lat_c3_010[(i,j)]
lat_c4_10 = lat_c4_010[(i,j)]
lon_c1_10 = lon_c1_010[(i,j)]
lon_c2_10 = lon_c2_010[(i,j)]
lon_c3_10 = lon_c3_010[(i,j)]
lon_c4_10 = lon_c4_010[(i,j)]

lat_10 = pd.DataFrame(lat_10)
lon_10 = pd.DataFrame(lon_10)
co_10 = pd.DataFrame(co_10)
grp_10 = pd.DataFrame(grp_10)
scl_10 = pd.DataFrame(scl_10)
orb_10 = 0*lat_10 + 3352
lat_c1_10 = pd.DataFrame(lat_c1_10)
lat_c2_10 = pd.DataFrame(lat_c2_10)
lat_c3_10 = pd.DataFrame(lat_c3_10)
lat_c4_10 = pd.DataFrame(lat_c4_10)
lon_c1_10 = pd.DataFrame(lon_c1_10)
lon_c2_10 = pd.DataFrame(lon_c2_10)
lon_c3_10 = pd.DataFrame(lon_c3_10)
lon_c4_10 = pd.DataFrame(lon_c4_10)
```

```
orbit11 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T172145_20180606T190513_03353_01_010202_
20190207T175957.nc", mode='r')

prod_11 = orbit11.groups['PRODUCT']
sdata_11 = prod_11.groups['SUPPORT_DATA']
geoloc_11 = sdata_11.groups['GEOLOCATIONS']

lat_0011 = prod_11.variables['latitude'][:]
lon_0011 = prod_11.variables['longitude'][:]
co_0011 = prod_11.variables['carbonmonoxide_total_column'][:]
qa_0011 = prod_11.variables['qa_value'][:]
grp_0011 = prod_11.variables['ground_pixel'][:]
scl_0011 = prod_11.variables['scanline'][:]
lat_c_0011 = geoloc_11.variables['latitude_bounds'][:]
lon_c_0011 = geoloc_11.variables['longitude_bounds'][:]

lat_011 = np.array(lat_0011)[0, :, :]
lon_011 = np.array(lon_0011)[0, :, :]
co_011 = np.array(co_0011)[0, :, :]
qa_011 = np.array(qa_0011)[0, :, :]
grp_011_ = np.array(grp_0011)[:]
scl_011_ = np.array(scl_0011)[:]
lat_c1_011 = np.array(lat_c_0011)[0,:,:,0]
lat_c2_011 = np.array(lat_c_0011)[0,:,:,1]
lat_c3_011 = np.array(lat_c_0011)[0,:,:,2]
lat_c4_011 = np.array(lat_c_0011)[0,:,:,3]
lon_c1_011 = np.array(lon_c_0011)[0,:,:,0]
lon_c2_011 = np.array(lon_c_0011)[0,:,:,1]
lon_c3_011 = np.array(lon_c_0011)[0,:,:,2]
lon_c4_011 = np.array(lon_c_0011)[0,:,:,3]

n_grp_11 = grp_011_.size
n_scl_11 = scl_011_.size

scl_011 = qa_011 * 0
grp_011 = qa_011 * 0

for ii in range (n_scl_11):
  for jj in range (n_grp_11):
```

```
    scl_011[ii][jj] = scl_011_[ii]
    grp_011[ii][jj] = grp_011_[jj]

(i,j) = (qa_011 >= 0.5).nonzero()
lat_11 = lat_011[(i,j)]
lon_11 = lon_011[(i,j)]
co_11 = co_011[(i,j)]
grp_11 = grp_011[(i,j)]
scl_11 = scl_011[(i,j)]
lat_c1_11 = lat_c1_011[(i,j)]
lat_c2_11 = lat_c2_011[(i,j)]
lat_c3_11 = lat_c3_011[(i,j)]
lat_c4_11 = lat_c4_011[(i,j)]
lon_c1_11 = lon_c1_011[(i,j)]
lon_c2_11 = lon_c2_011[(i,j)]
lon_c3_11 = lon_c3_011[(i,j)]
lon_c4_11 = lon_c4_011[(i,j)]

lat_11 = pd.DataFrame(lat_11)
lon_11 = pd.DataFrame(lon_11)
co_11 = pd.DataFrame(co_11)
grp_11 = pd.DataFrame(grp_11)
scl_11 = pd.DataFrame(scl_11)
orb_11 = 0*lat_11 + 3353
lat_c1_11 = pd.DataFrame(lat_c1_11)
lat_c2_11 = pd.DataFrame(lat_c2_11)
lat_c3_11 = pd.DataFrame(lat_c3_11)
lat_c4_11 = pd.DataFrame(lat_c4_11)
lon_c1_11 = pd.DataFrame(lon_c1_11)
lon_c2_11 = pd.DataFrame(lon_c2_11)
lon_c3_11 = pd.DataFrame(lon_c3_11)
lon_c4_11 = pd.DataFrame(lon_c4_11)



orbit12 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T190315_20180606T204643_03354_01_010202_
20190207T184522.nc", mode='r')

prod_12 = orbit12.groups['PRODUCT']
sdata_12 = prod_12.groups['SUPPORT_DATA']
```

```
geoloc_12 = sdata_12.groups['GEOLOCATIONS']

lat_0012 = prod_12.variables['latitude'][:]
lon_0012 = prod_12.variables['longitude'][:]
co_0012 = prod_12.variables['carbonmonoxide_total_column'][:]
qa_0012 = prod_12.variables['qa_value'][:]
grp_0012 = prod_12.variables['ground_pixel'][:]
scl_0012 = prod_12.variables['scanline'][:]
lat_c_0012 = geoloc_12.variables['latitude_bounds'][:]
lon_c_0012 = geoloc_12.variables['longitude_bounds'][:]

lat_012 = np.array(lat_0012)[0, :, :]
lon_012 = np.array(lon_0012)[0, :, :]
co_012 = np.array(co_0012)[0, :, :]
qa_012 = np.array(qa_0012)[0, :, :]
grp_012_ = np.array(grp_0012)[:]
scl_012_ = np.array(scl_0012)[:]
lat_c1_012 = np.array(lat_c_0012)[0,:,:,0]
lat_c2_012 = np.array(lat_c_0012)[0,:,:,1]
lat_c3_012 = np.array(lat_c_0012)[0,:,:,2]
lat_c4_012 = np.array(lat_c_0012)[0,:,:,3]
lon_c1_012 = np.array(lon_c_0012)[0,:,:,0]
lon_c2_012 = np.array(lon_c_0012)[0,:,:,1]
lon_c3_012 = np.array(lon_c_0012)[0,:,:,2]
lon_c4_012 = np.array(lon_c_0012)[0,:,:,3]

n_grp_12 = grp_012_.size
n_scl_12 = scl_012_.size

scl_012 = qa_012 * 0
grp_012 = qa_012 * 0

for ii in range (n_scl_12):
  for jj in range (n_grp_12):
    scl_012[ii][jj] = scl_012_[ii]
    grp_012[ii][jj] = grp_012_[jj]

(i,j) = (qa_012 >= 0.5).nonzero()
lat_12 = lat_012[(i,j)]
lon_12 = lon_012[(i,j)]
```

```
co_12 = co_012[(i,j)]
grp_12 = grp_012[(i,j)]
scl_12 = scl_012[(i,j)]
lat_c1_12 = lat_c1_012[(i,j)]
lat_c2_12 = lat_c2_012[(i,j)]
lat_c3_12 = lat_c3_012[(i,j)]
lat_c4_12 = lat_c4_012[(i,j)]
lon_c1_12 = lon_c1_012[(i,j)]
lon_c2_12 = lon_c2_012[(i,j)]
lon_c3_12 = lon_c3_012[(i,j)]
lon_c4_12 = lon_c4_012[(i,j)]

lat_12 = pd.DataFrame(lat_12)
lon_12 = pd.DataFrame(lon_12)
co_12 = pd.DataFrame(co_12)
grp_12 = pd.DataFrame(grp_12)
scl_12 = pd.DataFrame(scl_12)
orb_12 = 0*lat_12 + 3354
lat_c1_12 = pd.DataFrame(lat_c1_12)
lat_c2_12 = pd.DataFrame(lat_c2_12)
lat_c3_12 = pd.DataFrame(lat_c3_12)
lat_c4_12 = pd.DataFrame(lat_c4_12)
lon_c1_12 = pd.DataFrame(lon_c1_12)
lon_c2_12 = pd.DataFrame(lon_c2_12)
lon_c3_12 = pd.DataFrame(lon_c3_12)
lon_c4_12 = pd.DataFrame(lon_c4_12)



orbit13 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T204445_20180606T222813_03355_01_010202_
20190207T185801.nc", mode='r')

prod_13 = orbit13.groups['PRODUCT']
sdata_13 = prod_13.groups['SUPPORT_DATA']
geoloc_13 = sdata_13.groups['GEOLOCATIONS']

lat_0013 = prod_13.variables['latitude'][:]
lon_0013 = prod_13.variables['longitude'][:]
co_0013 = prod_13.variables['carbonmonoxide_total_column'][:]
qa_0013 = prod_13.variables['qa_value'][:]
```

```
grp_0013 = prod_13.variables['ground_pixel'][:]
scl_0013 = prod_13.variables['scanline'][:]
lat_c_0013 = geoloc_13.variables['latitude_bounds'][:]
lon_c_0013 = geoloc_13.variables['longitude_bounds'][:]

lat_013 = np.array(lat_0013)[0, :, :]
lon_013 = np.array(lon_0013)[0, :, :]
co_013 = np.array(co_0013)[0, :, :]
qa_013 = np.array(qa_0013)[0, :, :]
grp_013_ = np.array(grp_0013)[:]
scl_013_ = np.array(scl_0013)[:]
lat_c1_013 = np.array(lat_c_0013)[0,:,:,0]
lat_c2_013 = np.array(lat_c_0013)[0,:,:,1]
lat_c3_013 = np.array(lat_c_0013)[0,:,:,2]
lat_c4_013 = np.array(lat_c_0013)[0,:,:,3]
lon_c1_013 = np.array(lon_c_0013)[0,:,:,0]
lon_c2_013 = np.array(lon_c_0013)[0,:,:,1]
lon_c3_013 = np.array(lon_c_0013)[0,:,:,2]
lon_c4_013 = np.array(lon_c_0013)[0,:,:,3]

n_grp_13 = grp_013_.size
n_scl_13 = scl_013_.size

scl_013 = qa_013 * 0
grp_013 = qa_013 * 0

for ii in range (n_scl_13):
  for jj in range (n_grp_13):
    scl_013[ii][jj] = scl_013_[ii]
    grp_013[ii][jj] = grp_013_[jj]

(i,j) = (qa_013 >= 0.5).nonzero()
lat_13 = lat_013[(i,j)]
lon_13 = lon_013[(i,j)]
co_13 = co_013[(i,j)]
grp_13 = grp_013[(i,j)]
scl_13 = scl_013[(i,j)]
lat_c1_13 = lat_c1_013[(i,j)]
lat_c2_13 = lat_c2_013[(i,j)]
lat_c3_13 = lat_c3_013[(i,j)]
```

```
lat_c4_13 = lat_c4_013[(i,j)]
lon_c1_13 = lon_c1_013[(i,j)]
lon_c2_13 = lon_c2_013[(i,j)]
lon_c3_13 = lon_c3_013[(i,j)]
lon_c4_13 = lon_c4_013[(i,j)]

lat_13 = pd.DataFrame(lat_13)
lon_13 = pd.DataFrame(lon_13)
co_13 = pd.DataFrame(co_13)
grp_13 = pd.DataFrame(grp_13)
scl_13 = pd.DataFrame(scl_13)
orb_13 = 0*lat_13 + 3355
lat_c1_13 = pd.DataFrame(lat_c1_13)
lat_c2_13 = pd.DataFrame(lat_c2_13)
lat_c3_13 = pd.DataFrame(lat_c3_13)
lat_c4_13 = pd.DataFrame(lat_c4_13)
lon_c1_13 = pd.DataFrame(lon_c1_13)
lon_c2_13 = pd.DataFrame(lon_c2_13)
lon_c3_13 = pd.DataFrame(lon_c3_13)
lon_c4_13 = pd.DataFrame(lon_c4_13)



orbit14 = Dataset(r"C:
    S5P_RPRO_L2__CO_____20180606T222615_20180607T000943_03356_01_010202_
20190207T190506.nc", mode='r')

prod_14 = orbit14.groups['PRODUCT']
sdata_14 = prod_14.groups['SUPPORT_DATA']
geoloc_14 = sdata_14.groups['GEOLOCATIONS']

lat_0014 = prod_14.variables['latitude'][:]
lon_0014 = prod_14.variables['longitude'][:]
co_0014 = prod_14.variables['carbonmonoxide_total_column'][:]
qa_0014 = prod_14.variables['qa_value'][:]
grp_0014 = prod_14.variables['ground_pixel'][:]
scl_0014 = prod_14.variables['scanline'][:]
lat_c_0014 = geoloc_14.variables['latitude_bounds'][:]
lon_c_0014 = geoloc_14.variables['longitude_bounds'][:]

lat_014 = np.array(lat_0014)[0, :, :]
```

```
lon_014 = np.array(lon_0014)[0, :, :]
co_014 = np.array(co_0014)[0, :, :]
qa_014 = np.array(qa_0014)[0, :, :]
grp_014_ = np.array(grp_0014)[:]
scl_014_ = np.array(scl_0014)[:]
lat_c1_014 = np.array(lat_c_0014)[0,:,:,0]
lat_c2_014 = np.array(lat_c_0014)[0,:,:,1]
lat_c3_014 = np.array(lat_c_0014)[0,:,:,2]
lat_c4_014 = np.array(lat_c_0014)[0,:,:,3]
lon_c1_014 = np.array(lon_c_0014)[0,:,:,0]
lon_c2_014 = np.array(lon_c_0014)[0,:,:,1]
lon_c3_014 = np.array(lon_c_0014)[0,:,:,2]
lon_c4_014 = np.array(lon_c_0014)[0,:,:,3]

n_grp_14 = grp_014_.size
n_scl_14 = scl_014_.size

scl_014 = qa_014 * 0
grp_014 = qa_014 * 0

for ii in range (n_scl_14):
  for jj in range (n_grp_14):
    scl_014[ii][jj] = scl_014_[ii]
    grp_014[ii][jj] = grp_014_[jj]

(i,j) = (qa_014 >= 0.5).nonzero()
lat_14 = lat_014[(i,j)]
lon_14 = lon_014[(i,j)]
co_14 = co_014[(i,j)]
grp_14 = grp_014[(i,j)]
scl_14 = scl_014[(i,j)]
lat_c1_14 = lat_c1_014[(i,j)]
lat_c2_14 = lat_c2_014[(i,j)]
lat_c3_14 = lat_c3_014[(i,j)]
lat_c4_14 = lat_c4_014[(i,j)]
lon_c1_14 = lon_c1_014[(i,j)]
lon_c2_14 = lon_c2_014[(i,j)]
lon_c3_14 = lon_c3_014[(i,j)]
lon_c4_14 = lon_c4_014[(i,j)]
```

```
lat_14 = pd.DataFrame(lat_14)
lon_14 = pd.DataFrame(lon_14)
co_14 = pd.DataFrame(co_14)
grp_14 = pd.DataFrame(grp_14)
scl_14 = pd.DataFrame(scl_14)
orb_14 = 0*lat_14 + 3356
lat_c1_14 = pd.DataFrame(lat_c1_14)
lat_c2_14 = pd.DataFrame(lat_c2_14)
lat_c3_14 = pd.DataFrame(lat_c3_14)
lat_c4_14 = pd.DataFrame(lat_c4_14)
lon_c1_14 = pd.DataFrame(lon_c1_14)
lon_c2_14 = pd.DataFrame(lon_c2_14)
lon_c3_14 = pd.DataFrame(lon_c3_14)
lon_c4_14 = pd.DataFrame(lon_c4_14)


lat_ESA = pd.concat([lat_01, lat_02, lat_03, lat_04, lat_05, lat_06, lat_07,
            lat_08, lat_09, lat_10, lat_11, lat_12, lat_13, lat_14])
lat_ESA = np.array(lat_ESA)

lon_ESA = pd.concat([lon_01, lon_02, lon_03, lon_04, lon_05, lon_06, lon_07,
            lon_08, lon_09, lon_10, lon_11, lon_12, lon_13, lon_14])
lon_ESA = np.array(lon_ESA)

co_ESA = pd.concat([co_01, co_02, co_03, co_04, co_05, co_06, co_07,
            co_08, co_09, co_10, co_11, co_12, co_13, co_14])
co_ESA = np.array(co_ESA)

grp_ESA = pd.concat([grp_01, grp_02, grp_03, grp_04, grp_05, grp_06, grp_07,
            grp_08, grp_09, grp_10, grp_11, grp_12, grp_13, grp_14])
grp_ESA = np.array(grp_ESA)

scl_ESA = pd.concat([scl_01, scl_02, scl_03, scl_04, scl_05, scl_06, scl_07,
            scl_08, scl_09, scl_10, scl_11, scl_12, scl_13, scl_14])
scl_ESA = np.array(scl_ESA)

orb_ESA = pd.concat([orb_01, orb_02, orb_03, orb_04, orb_05, orb_06, orb_07,
            orb_08, orb_09, orb_10, orb_11, orb_12, orb_13, orb_14])
orb_ESA = np.array(orb_ESA)
```

```
lat_c1_ESA = pd.concat([lat_c1_01, lat_c1_02, lat_c1_03, lat_c1_04, lat_c1_05,
     lat_c1_06, lat_c1_07,
              lat_c1_08, lat_c1_09, lat_c1_10, lat_c1_11, lat_c1_12, lat_c1_13,
    lat_c1_14])
lat_c1_ESA = np.array(lat_c1_ESA)

lat_c2_ESA = pd.concat([lat_c2_01, lat_c2_02, lat_c2_03, lat_c2_04, lat_c2_05,
     lat_c2_06, lat_c2_07,
              lat_c2_08, lat_c2_09, lat_c2_10, lat_c2_11, lat_c2_12, lat_c2_13,
    lat_c2_14])
lat_c2_ESA = np.array(lat_c2_ESA)

lat_c3_ESA = pd.concat([lat_c3_01, lat_c3_02, lat_c3_03, lat_c3_04, lat_c3_05,
     lat_c3_06, lat_c3_07,
              lat_c3_08, lat_c3_09, lat_c3_10, lat_c3_11, lat_c3_12, lat_c3_13,
    lat_c3_14])
lat_c3_ESA = np.array(lat_c3_ESA)

lat_c4_ESA = pd.concat([lat_c4_01, lat_c4_02, lat_c4_03, lat_c4_04, lat_c4_05,
     lat_c4_06, lat_c4_07,
              lat_c4_08, lat_c4_09, lat_c4_10, lat_c4_11, lat_c4_12, lat_c4_13,
    lat_c4_14])
lat_c4_ESA = np.array(lat_c4_ESA)

lon_c1_ESA = pd.concat([lon_c1_01, lon_c1_02, lon_c1_03, lon_c1_04, lon_c1_05,
     lon_c1_06, lon_c1_07,
              lon_c1_08, lon_c1_09, lon_c1_10, lon_c1_11, lon_c1_12, lon_c1_13,
    lon_c1_14])
lon_c1_ESA = np.array(lon_c1_ESA)

lon_c2_ESA = pd.concat([lon_c2_01, lon_c2_02, lon_c2_03, lon_c2_04, lon_c2_05,
     lon_c2_06, lon_c2_07,
              lon_c2_08, lon_c2_09, lon_c2_10, lon_c2_11, lon_c2_12, lon_c2_13,
    lon_c2_14])
lon_c2_ESA = np.array(lon_c2_ESA)

lon_c3_ESA = pd.concat([lon_c3_01, lon_c3_02, lon_c3_03, lon_c3_04, lon_c3_05,
     lon_c3_06, lon_c3_07,
              lon_c3_08, lon_c3_09, lon_c3_10, lon_c3_11, lon_c3_12, lon_c3_13,
    lon_c3_14])
```

```
lon_c3_ESA = np.array(lon_c3_ESA)

lon_c4_ESA = pd.concat([lon_c4_01, lon_c4_02, lon_c4_03, lon_c4_04, lon_c4_05,
     lon_c4_06, lon_c4_07,
              lon_c4_08, lon_c4_09, lon_c4_10, lon_c4_11, lon_c4_12, lon_c4_13,
    lon_c4_14])
lon_c4_ESA = np.array(lon_c4_ESA)

n_ESA = co_ESA.size

S5P_20180606_co_ESA = open(r"C:S5P_20180606_co_ESA.txt", "w")
S5P_20180606_co_ESA.write("lat, lat_c1, lat_c2, lat_c3, lat_c4, lon, lon_c1,
    lon_c2, lon_c3, lon_c4, co\n")

for k in np.arange(n_ESA):
  S5P_20180606_co_ESA.write("%10.5f %10.5f %10.5f %10.5f %10.5f %10.5f %10.5f
    %10.5f %10.5f %10.5f %10.5f\n" % (lat_ESA[k], lat_c1_ESA[k], lat_c2_ESA[k
    ], lat_c3_ESA[k], lat_c4_ESA[k], lon_ESA[k], lon_c1_ESA[k], lon_c2_ESA[k],
     lon_c3_ESA[k], lon_c4_ESA[k], co_ESA[k]))

S5P_20180606_co_ESA.close()

# condition: only data for both ESA and IUP

lat_ESA = lat_ESA[:,0]
lon_ESA = lon_ESA[:,0]
co_ESA = co_ESA[:,0]
lat_c1_ESA = lat_c1_ESA[:,0]
lat_c2_ESA = lat_c2_ESA[:,0]
lat_c3_ESA = lat_c3_ESA[:,0]
lat_c4_ESA = lat_c4_ESA[:,0]
lon_c1_ESA = lon_c1_ESA[:,0]
lon_c2_ESA = lon_c2_ESA[:,0]
lon_c3_ESA = lon_c3_ESA[:,0]
lon_c4_ESA = lon_c4_ESA[:,0]

lat_ESA_col = []
lon_ESA_col = []
co_ESA_col = []
lat_IUP_col = []
```

```
lon_IUP_col = []
co_IUP_col = []
lat_c1_ESA_col = []
lat_c2_ESA_col = []
lat_c3_ESA_col = []
lat_c4_ESA_col = []
lon_c1_ESA_col = []
lon_c2_ESA_col = []
lon_c3_ESA_col = []
lon_c4_ESA_col = []

S5P_20180606_co_col = open(r"C:S5P_20180606_co_col.txt", "w")
S5P_20180606_co_col.write("lat, lat_c1, lat_c2, lat_c3, lat_c4, lon, lon_c1, 
    lon_c2, lon_c3, lon_c4, co_ESA, co_IUP\n")

for j in np.arange(n_ESA):
  mask = (scl_IUP == scl_ESA[j]) & (grp_IUP == grp_ESA[j]) & (orb_IUP ==
    orb_ESA[j])

  if len(lat_IUP[mask]) > 0:

    lat_ESA_col = np.append(lat_ESA_col, lat_ESA[j])
    lat_c1_ESA_col = np.append(lat_c1_ESA_col, lat_c1_ESA[j])
    lat_c2_ESA_col = np.append(lat_c2_ESA_col, lat_c2_ESA[j])
    lat_c3_ESA_col = np.append(lat_c3_ESA_col, lat_c3_ESA[j])
    lat_c4_ESA_col = np.append(lat_c4_ESA_col, lat_c4_ESA[j])
    lon_ESA_col = np.append(lon_ESA_col, lon_ESA[j])
    lon_c1_ESA_col = np.append(lon_c1_ESA_col, lon_c1_ESA[j])
    lon_c2_ESA_col = np.append(lon_c2_ESA_col, lon_c2_ESA[j])
    lon_c3_ESA_col = np.append(lon_c3_ESA_col, lon_c3_ESA[j])
    lon_c4_ESA_col = np.append(lon_c4_ESA_col, lon_c4_ESA[j])
    co_ESA_col = np.append(co_ESA_col, co_ESA[j])
    lat_IUP_col = np.append(lat_IUP_col, lat_IUP[mask])
    lon_IUP_col = np.append(lon_IUP_col, lon_IUP[mask])
    co_IUP_col = np.append(co_IUP_col, co_IUP[mask])

    S5P_20180606_co_col.write("%10.5f %10.5f %10.5f %10.5f %10.5f %10.5f %10.5f
     %10.5f %10.5f %10.5f %10.5f %10.5f\n" % (lat_ESA[j], lat_c1_ESA[j],
     lat_c2_ESA[j], lat_c3_ESA[j], lat_c4_ESA[j], lon_ESA[j], lon_c1_ESA[j],
     lon_c2_ESA[j], lon_c3_ESA[j], lon_c4_ESA[j], co_ESA[j], co_IUP[mask]))
```

```
S5P_20180606_co_col.close()
```

Appendix 2: Global Maps

```
from mpl_toolkits.basemap import Basemap

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon
from matplotlib import cm

lat_col, lat_c1_col, lat_c2_col, lat_c3_col, lat_c4_col, lon_col, lon_c1_col,
    lon_c2_col, lon_c3_col, lon_c4_col, ESA_col, IUP_col = np.genfromtxt(r"
    S5P_20180606_co_col.txt", unpack=True, skip_header=1)
lat_IUP, lat_c1_IUP, lat_c2_IUP, lat_c3_IUP, lat_c4_IUP, lon_IUP, lon_c1_IUP,
    lon_c2_IUP, lon_c3_IUP, lon_c4_IUP, co_IUP = np.genfromtxt(r"
    S5P_20180606_co_IUP.txt", unpack=True, skip_header=1)
lat_ESA, lat_c1_ESA, lat_c2_ESA, lat_c3_ESA, lat_c4_ESA, lon_ESA, lon_c1_ESA,
    lon_c2_ESA, lon_c3_ESA, lon_c4_ESA, co_ESA = np.genfromtxt(r"
    S5P_20180606_co_ESA.txt", unpack=True, skip_header=1)


## IUP (collocated)

map = Basemap()
map.drawcoastlines()

map.drawlsmask(land_color='lightgray', ocean_color='whitesmoke', lsmask=None,
    lsmask_lons=None, lsmask_lats=None, lakes=True, resolution='l', grid=5)


clr_min = 0.01
clr_max = 0.04
clr_n = 150

pxcol = cm.get_cmap('viridis', clr_n)


for k in range(len(lat_col)):

  clr = int(np.floor((IUP_col[k]-clr_min)/((clr_max - clr_min)/clr_n)))
```

```
    if IUP_col[k] < clr_min:
        clr = 0
    if IUP_col[k] > clr_max:
        clr = clr_n

    x1,y1 = map(lon_c1_col[k],lat_c1_col[k])
    x2,y2 = map(lon_c2_col[k],lat_c2_col[k])
    x3,y3 = map(lon_c3_col[k],lat_c3_col[k])
    x4,y4 = map(lon_c4_col[k],lat_c4_col[k])

    if ((np.sign(lon_c1_col[k]) == np.sign(lon_c2_col[k])) & (np.sign(lon_c1_col
        [k]) == np.sign(lon_c3_col[k])) &
        (np.sign(lon_c1_col[k]) == np.sign(lon_c4_col[k])) & (np.sign(lon_c2_col[k
        ]) == np.sign(lon_c3_col[k])) &
        (np.sign(lon_c2_col[k]) == np.sign(lon_c4_col[k])) & (np.sign(lon_c3_col[k
        ]) == np.sign(lon_c4_col[k]))):

        poly = Polygon([(x1,y1),(x2,y2),(x3,y3),(x4,y4)], facecolor=pxcol(clr))
        plt.gca().add_patch(poly)


plt.scatter(lon_col, lat_col, c=IUP_col, cmap=pxcol, vmin=clr_min, vmax=
    clr_max, s=0)
plt.colorbar(orientation='horizontal')

plt.axis([-180,180,-90,90])
plt.title('S5P Carbon Monoxide WFMD[@SICOR], 20180606', fontsize='large')
plt.xlabel('CO [mol m$^{-2}$]', fontsize='large')


## ESA (collocated)

map = Basemap()
map.drawcoastlines()

map.drawlsmask(land_color='lightgray', ocean_color='whitesmoke', lsmask=None,
    lsmask_lons=None, lsmask_lats=None, lakes=True, resolution='l', grid=5)
```

```
clr_min = 0.01
clr_max = 0.04
clr_n = 150

pxcol = cm.get_cmap('viridis', clr_n)


for k in range(len(lat_col)):

  clr = int(np.floor((ESA_col[k]-clr_min)/((clr_max - clr_min)/clr_n)))

  if ESA_col[k] < clr_min:
    clr = 0
  if ESA_col[k] > clr_max:
    clr = clr_n

  x1,y1 = map(lon_c1_col[k],lat_c1_col[k])
  x2,y2 = map(lon_c2_col[k],lat_c2_col[k])
  x3,y3 = map(lon_c3_col[k],lat_c3_col[k])
  x4,y4 = map(lon_c4_col[k],lat_c4_col[k])

  if ((np.sign(lon_c1_col[k]) == np.sign(lon_c2_col[k])) & (np.sign(lon_c1_col
    [k]) == np.sign(lon_c3_col[k])) &
   (np.sign(lon_c1_col[k]) == np.sign(lon_c4_col[k])) & (np.sign(lon_c2_col[k
    ]) == np.sign(lon_c3_col[k])) &
   (np.sign(lon_c2_col[k]) == np.sign(lon_c4_col[k])) & (np.sign(lon_c3_col[k
    ]) == np.sign(lon_c4_col[k]))):

    poly = Polygon([(x1,y1),(x2,y2),(x3,y3),(x4,y4)], facecolor=pxcol(clr))
    plt.gca().add_patch(poly)


plt.scatter(lon_col, lat_col, c=ESA_col, cmap=pxcol, vmin=clr_min, vmax=
    clr_max, s=0)
plt.colorbar(orientation='horizontal')

plt.axis([-180,180,-90,90])
plt.title('S5P Carbon Monoxide SICOR[@WFMD], 20180606', fontsize='large')
plt.xlabel('CO [mol m$^{-2}$]', fontsize='large');
```

```
plt.savefig('S5P20180606_co_worldmap_SICORcol', dpi=720)
plt.close()

## absolute difference

diff_col = (IUP_col - ESA_col)

map = Basemap()
map.drawcoastlines()

map.drawlsmask(land_color='lightgray', ocean_color='whitesmoke', lsmask=None,
    lsmask_lons=None, lsmask_lats=None, lakes=True, resolution='l', grid=5)


clr_min = -0.003
clr_max = 0.003
clr_n = 150

pxcol = cm.get_cmap('RdYlBu', clr_n)


for k in range(len(lat_col)):

  clr = int(np.floor((diff_col[k]-clr_min)/((clr_max - clr_min)/clr_n)))

  if diff_col[k] < clr_min:
    clr = 0
  if diff_col[k] > clr_max:
    clr = clr_n

  x1,y1 = map(lon_c1_col[k],lat_c1_col[k])
  x2,y2 = map(lon_c2_col[k],lat_c2_col[k])
  x3,y3 = map(lon_c3_col[k],lat_c3_col[k])
  x4,y4 = map(lon_c4_col[k],lat_c4_col[k])

  if ((np.sign(lon_c1_col[k]) == np.sign(lon_c2_col[k])) & (np.sign(lon_c1_col
    [k]) == np.sign(lon_c3_col[k])) &
   (np.sign(lon_c1_col[k]) == np.sign(lon_c4_col[k])) & (np.sign(lon_c2_col[k
    ]) == np.sign(lon_c3_col[k])) &
```

```
    (np.sign(lon_c2_col[k]) == np.sign(lon_c4_col[k])) & (np.sign(lon_c3_col[k
    ]) == np.sign(lon_c4_col[k]))):

     poly = Polygon([(x1,y1),(x2,y2),(x3,y3),(x4,y4)], facecolor=pxcol(clr))
     plt.gca().add_patch(poly)


plt.scatter(lon_col, lat_col, c=diff_col, cmap=pxcol, vmin=clr_min, vmax=
    clr_max, s=0)
plt.colorbar(orientation='horizontal')

plt.axis([-180,180,-90,90])
plt.title('S5P CO Absolute Difference, 20180606', fontsize='large')
plt.xlabel('$\Delta$ CO [mol m$^{-2}$]', fontsize='large')


## relative difference

rel_diff = ((IUP_col - ESA_col)/ESA_col)*100

map = Basemap()
map.drawcoastlines()

map.drawlsmask(land_color='lightgray', ocean_color='whitesmoke', lsmask=None,
    lsmask_lons=None, lsmask_lats=None, lakes=True, resolution='l', grid=5)


clr_min = -10
clr_max = 10
clr_n = 150

pxcol = cm.get_cmap('RdYlBu', clr_n)


for k in range(len(lat_col)):

  clr = int(np.floor((rel_diff[k]-clr_min)/((clr_max - clr_min)/clr_n)))

  if rel_diff[k] < clr_min:
    clr = 0
```

```
  if rel_diff[k] > clr_max:
    clr = clr_n

  x1,y1 = map(lon_c1_col[k],lat_c1_col[k])
  x2,y2 = map(lon_c2_col[k],lat_c2_col[k])
  x3,y3 = map(lon_c3_col[k],lat_c3_col[k])
  x4,y4 = map(lon_c4_col[k],lat_c4_col[k])

  if ((np.sign(lon_c1_col[k]) == np.sign(lon_c2_col[k])) & (np.sign(lon_c1_col
    [k]) == np.sign(lon_c3_col[k])) &
   (np.sign(lon_c1_col[k]) == np.sign(lon_c4_col[k])) & (np.sign(lon_c2_col[k
    ]) == np.sign(lon_c3_col[k])) &
   (np.sign(lon_c2_col[k]) == np.sign(lon_c4_col[k])) & (np.sign(lon_c3_col[k
    ]) == np.sign(lon_c4_col[k]))):

    poly = Polygon([(x1,y1),(x2,y2),(x3,y3),(x4,y4)], facecolor=pxcol(clr))
    plt.gca().add_patch(poly)


plt.scatter(lon_col, lat_col, c=rel_diff, cmap=pxcol, vmin=clr_min, vmax=
    clr_max, s=0)
plt.colorbar(orientation='horizontal')

plt.axis([-180,180,-90,90])
plt.title('S5P CO Relative Difference, 20180606', fontsize='large')
plt.xlabel('$\Delta$ CO [%]', fontsize='large')
```

Appendix 3: Additional Plots

```
import numpy as np
import matplotlib.pyplot as plt
import statistics as stat
import scipy.stats

## working with text file

lat_col, lat_c1_col, lat_c2_col, lat_c3_col, lat_c4_col, lon_col, lon_c1_col,
    lon_c2_col, lon_c3_col, lon_c4_col, ESA_col, IUP_col = np.genfromtxt(r"
    S5P_20180606_co_col.txt", unpack=True, skip_header=1)
lat_IUP, lat_c1_IUP, lat_c2_IUP, lat_c3_IUP, lat_c4_IUP, lon_IUP, lon_c1_IUP,
    lon_c2_IUP, lon_c3_IUP, lon_c4_IUP, co_IUP = np.genfromtxt(r"
    S5P_20180606_co_IUP.txt", unpack=True, skip_header=1)
lat_ESA, lat_c1_ESA, lat_c2_ESA, lat_c3_ESA, lat_c4_ESA, lon_ESA, lon_c1_ESA,
    lon_c2_ESA, lon_c3_ESA, lon_c4_ESA, co_ESA = np.genfromtxt(r"
    S5P_20180606_co_ESA.txt", unpack=True, skip_header=1)

n_IUP_col = len(IUP_col)
n_IUP_col_str = str(n_IUP_col)

n_ESA_col = len(ESA_col)
n_ESA_col_str = str(n_ESA_col)

n_IUP = len(co_IUP)
n_IUP_str = str(n_IUP)

n_ESA = len(co_ESA)
n_ESA_str = str(n_ESA)

ESA_col = np.array(ESA_col)
IUP_col = np.array(IUP_col)

co_ESA = np.array(co_ESA)
co_IUP = np.array(co_IUP)

### PLOT CO ESA(deeppink) OVER IUP(BLUE)
```

```
# CO ESA ALL

plt.scatter(lat_ESA, co_ESA, color='deeppink', s = 0.1)
plt.axis([-90, 90, -0.15, 0.4])
plt.text(40, 0.38, 'N-OPER = ' + n_ESA_str, horizontalalignment='left',
    verticalalignment = 'top', color = 'deeppink')

# CO IUP ALL

plt.scatter(lat_IUP, co_IUP, color='b', s = 0.1)
plt.axis([-90, 90, -0.15, 0.4])
plt.text(40, 0.34, 'N-WFMD = ' + n_IUP_str, horizontalalignment='left',
    verticalalignment = 'top', color = 'b')

plt.xlabel('latitude [deg]')
plt.ylabel('xco [mol m-2]')
plt.title('S5P carbon monoxide WFMD & OPER (qual=good), 20180606')


### PLOT CO ESA(deeppink) OVER IUP(BLUE) COLLOCATED

# CO ESA collocated

plt.scatter(lat_col, ESA_col, color='deeppink', s = 0.1)
plt.axis([-90, 90, -0.15, 0.4])
plt.text(40, 0.38, 'N-OPER = ' + n_ESA_col_str, horizontalalignment='left',
    verticalalignment = 'top', color = 'deeppink')

# CO IUP collocated

plt.scatter(lat_col, IUP_col, color='b', s = 0.1)
plt.axis([-90, 90, -0.15, 0.4])
plt.text(40, 0.34, 'N-WFMD = ' + n_IUP_col_str, horizontalalignment='left',
    verticalalignment = 'top', color = 'b')

plt.xlabel('latitude [deg]')
plt.ylabel('xco [mol m-2]')
plt.title('S5P carbon monoxide WFMD & OPER (collocated), 20180606')
```

```
### PLOT DIFFERENCE IUP - ESA COLLOCATED

# CO IUP - ESA

diff_IUP_ESA = np.array(IUP_col-ESA_col)
mean_diff_IUP_ESA = str(stat.mean(diff_IUP_ESA))
#print('mean difference: ' + mean_diff_IUP_ESA)

std_diff_IUP_ESA = str(stat.stdev(diff_IUP_ESA))
#print('std difference: ' + std_diff_IUP_ESA)

n_diff_IUP_ESA = len(diff_IUP_ESA)

plt.scatter(lat_col, diff_IUP_ESA, color='c', s = 0.1)
plt.axis([-90, 90, -0.04, 0.04])
plt.xlabel('latitude [deg]')
plt.ylabel('xco (WFMD - OPER) [mol m-2]')
plt.axhline(y=0, color='black', linewidth=0.5, linestyle=':')
plt.title('S5P Carbon Monoxide WFMD - OPER, 20180606')

#mask01 = (lat_col > -90) & (lat_col < -80)
#mean01 = stat.mean(diff_IUP_ESA[mask01])
#std01 = stat.stdev(diff_IUP_ESA[mask01])

#mask02 = (lat_col > -80) & (lat_col < -70)
#mean02 = stat.mean(diff_IUP_ESA[mask02])
#std02 = stat.stdev(diff_IUP_ESA[mask02])

#mask03 = (lat_col > -70) & (lat_col < -60)
#mean03 = stat.mean(diff_IUP_ESA[mask03])
#std03 = stat.stdev(diff_IUP_ESA[mask03])

#mask04 = (lat_col > -60) & (lat_col < -50)
#mean04 = stat.mean(diff_IUP_ESA[mask04])
#std04 = stat.stdev(diff_IUP_ESA[mask04])

mask05 = (lat_col > -50) & (lat_col < -40)
mean05 = stat.mean(diff_IUP_ESA[mask05])
std05 = stat.stdev(diff_IUP_ESA[mask05])
```

```
mask06 = (lat_col > -40) & (lat_col < -30)
mean06 = stat.mean(diff_IUP_ESA[mask06])
std06 = stat.stdev(diff_IUP_ESA[mask06])

mask07 = (lat_col > -30) & (lat_col < -20)
mean07 = stat.mean(diff_IUP_ESA[mask07])
std07 = stat.stdev(diff_IUP_ESA[mask07])

mask08 = (lat_col > -20) & (lat_col < -10)
mean08 = stat.mean(diff_IUP_ESA[mask08])
std08 = stat.stdev(diff_IUP_ESA[mask08])

mask09 = (lat_col > -10) & (lat_col < 0)
mean09 = stat.mean(diff_IUP_ESA[mask09])
std09 = stat.stdev(diff_IUP_ESA[mask09])

mask10 = (lat_col > 0) & (lat_col < 10)
mean10 = stat.mean(diff_IUP_ESA[mask10])
std10 = stat.stdev(diff_IUP_ESA[mask10])

mask11 = (lat_col > 10) & (lat_col < 20)
mean11 = stat.mean(diff_IUP_ESA[mask11])
std11 = stat.stdev(diff_IUP_ESA[mask11])

mask12 = (lat_col > 20) & (lat_col < 30)
mean12 = stat.mean(diff_IUP_ESA[mask12])
std12 = stat.stdev(diff_IUP_ESA[mask12])

mask13 = (lat_col > 30) & (lat_col < 40)
mean13 = stat.mean(diff_IUP_ESA[mask13])
std13 = stat.stdev(diff_IUP_ESA[mask13])

mask14 = (lat_col > 40) & (lat_col < 50)
mean14 = stat.mean(diff_IUP_ESA[mask14])
std14 = stat.stdev(diff_IUP_ESA[mask14])

mask15 = (lat_col > 50) & (lat_col < 60)
mean15 = stat.mean(diff_IUP_ESA[mask15])
std15 = stat.stdev(diff_IUP_ESA[mask15])
```

```
mask16 = (lat_col > 60) & (lat_col < 70)
mean16 = stat.mean(diff_IUP_ESA[mask16])
std16 = stat.stdev(diff_IUP_ESA[mask16])

mask17 = (lat_col > 70) & (lat_col < 80)
mean17 = stat.mean(diff_IUP_ESA[mask17])
std17 = stat.stdev(diff_IUP_ESA[mask17])

mask18 = (lat_col > 80) & (lat_col < 90)
mean18 = stat.mean(diff_IUP_ESA[mask18])
std18 = stat.stdev(diff_IUP_ESA[mask18])

x=-85

#plt.errorbar(x, mean01, yerr=std01, xerr=5, color='black', linewidth=0.65)
#plt.errorbar(x+10, mean02, yerr=std02, xerr=5, color='black', linewidth=0.65)
#plt.errorbar(x+20, mean03, yerr=std03, xerr=5, color='black', linewidth=0.65)
#plt.errorbar(x+30, mean04, yerr=std04, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+40, mean05, yerr=std05, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+50, mean06, yerr=std06, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+60, mean07, yerr=std07, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+70, mean08, yerr=std08, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+80, mean09, yerr=std09, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+90, mean10, yerr=std10, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+100, mean11, yerr=std11, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+110, mean12, yerr=std12, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+120, mean13, yerr=std13, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+130, mean14, yerr=std14, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+140, mean15, yerr=std15, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+150, mean16, yerr=std16, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+160, mean17, yerr=std17, xerr=5, color='black', linewidth=0.65)
plt.errorbar(x+170, mean18, yerr=std18, xerr=5, color='black', linewidth=0.65)


### HEATMAP

# ESA vs IUP HEXBIN // HEATMAP

plt.hexbin(ESA_col, IUP_col, mincnt=1, cmap='rainbow', gridsize=150)
plt.colorbar()
```

```
plt.axis([0.0, 0.12, 0.0, 0.12])
plt.grid(linestyle=':')
x = np.linspace(-0.15, 0.4, 5)
y=x
plt.plot(x, y, '-r', linestyle='--')
plt.xlabel('xco OPER [mol m-2]')
plt.ylabel('xco WFMD [mol m-2]')
plt.title('xco OPER - xco WFMD, 20180606')

N = str(len(IUP_col))
D = str(stat.mean(np.array(IUP_col-ESA_col)))[:7]
S = str(stat.stdev(IUP_col-ESA_col))[:7]
R = str(scipy.stats.pearsonr(ESA_col,IUP_col))[1:][:7]

plt.text(0.1, 0.06, 'N = ' + N, horizontalalignment='left', verticalalignment
    = 'bottom', color = 'black')
plt.text(0.1, 0.05, 'D = ' + D, horizontalalignment='left', verticalalignment
    = 'bottom', color = 'black')
plt.text(0.1, 0.04, 'S = ' + S, horizontalalignment='left', verticalalignment
    = 'bottom', color = 'black')
plt.text(0.1, 0.03, 'R = ' + R, horizontalalignment='left', verticalalignment
    = 'bottom', color = 'black')
```

Statutory Declaration

Family Name, Given/First Name	Krutsch, Ann-Kathrin Andrea
Matriculation number	30001154
What kind of thesis are you submitting: Bachelor-, Master- or PhD-Thesis	Bachelor-Thesis

English: Declaration of Authorship

I hereby declare that the thesis submitted was created and written solely by myself without any external support. Any sources, direct or indirect, are marked as such. I am aware of the fact that the contents of the thesis in digital form may be revised with regard to usage of unauthorized aid as well as whether the whole or parts of it may be identified as plagiarism. I do agree my work to be entered into a database for it to be compared with existing sources, where it will remain in order to enable further comparisons with future theses. This does not grant any rights of reproduction and usage, however.

The Thesis has been written independently and has not been submitted at any other university for the conferral of a PhD degree; neither has the thesis been previously published in full.

German: Erklärung der Autorenschaft (Urheberschaft)

Ich erkläre hiermit, dass die vorliegende Arbeit ohne fremde Hilfe ausschließlich von mir erstellt und geschrieben worden ist. Jedwede verwendeten Quellen, direkter oder indirekter Art, sind als solche kenntlich gemacht worden. Mir ist die Tatsache bewusst, dass der Inhalt der Thesis in digitaler Form geprüft werden kann im Hinblick darauf, ob es sich ganz oder in Teilen um ein Plagiat handelt. Ich bin damit einverstanden, dass meine Arbeit in einer Datenbank eingegeben werden kann, um mit bereits bestehenden Quellen verglichen zu werden und dort auch verbleibt, um mit zukünftigen Arbeiten verglichen werden zu können. Dies berechtigt jedoch nicht zur Verwendung oder Vervielfältigung.

Diese Arbeit wurde in der vorliegenden Form weder einer anderen Prüfungsbehörde vorgelegt noch wurde das Gesamtdokument bisher veröffentlicht.

July 24th, 2020
....

Date, Signature