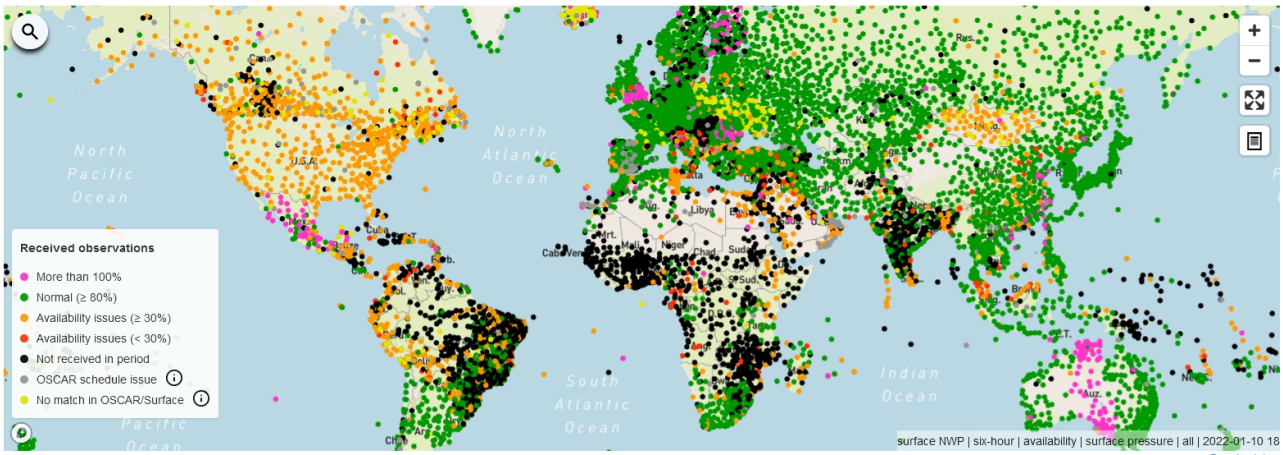
Master’s thesis project in Space, Earth and Environment

Develop a time-schedule-mapping algorithm for worldwide observations of the Earth system



Background:
National Meteorological and Hydrological Services face complex challenges now and in the future, which demand more and better observations and more data sharing between services. Weather and climate observations are shared worldwide through networks of the World Meteorological Organization (WMO) and are monitored in terms of availability, quality and timeliness of the data by the WMO Data Quality Monitoring System. To be able to use observational data efficiently, not only the data, like the temperature reading, itself but also the metadata, for example the altitude of the station, is crucial. A critical element for monitoring is metadata describing how often observations are taken. One aspect of the WMO Data Quality Monitoring System is to check if as many observations have been send to operational data centres as expected from the metadata record.

In order to know how many observations are expected in a given time period a mapping algorithm is needed that maps an abstract schedule, such as daily observations every hour, to an concrete interval, for example January 2022.

Your master thesis aims to develop such an algorithm to map an abstract schedule on to a concrete date interval, which might be used in a productive system of the WMO to assess worldwide data.

Your job:
We are looking for a student who is interested in developing a time-schedule-mapping algorithm in Python aiming to improve the data quality of worldwide exchanged observations. You will be working in a stimulating environment in close contact to colleges of an international organization (WMO) with the opportunity to learn about applied computational science/programming.

Working plan for the project:
1.) Understand the problem from a theoretical perspective. Make a list of requirements for an algorithm to map an abstract schedule to a concrete data interval. Analyse how an abstract schedule can be normalised and represented in an algorithm. Examine the role of equivalence of different (parallel) schedules and how to deal with different base times, different time zones, clock change and leap years.
2.) Carefully review the capabilities of existing Python date libraries with respect to the requirements (as used in the financial sector etc.).

3.) Write a new Python module to map schedules based on your theoretical considerations and optimisation.
4.) Test your code and compare it to the existing library and an easy un-optimized solution.
5.) Validate your code and test it in the operational system.

Prerequisites:
Experience of programming in Python and basic knowledge in mathematics.
Credits:
30 credits.
Starting time:
Any time.

Contact:
Luisa Ickes
Department of Space, Earth and Environment, Division of Geoscience and Remote Sensing.
luisa.ickes@chalmers.se