A tool for the quantitative assessment of long time series of satellite SST A prototype for CDAF framework implementation

Jean-François Piollé (Ifremer), Jon Mittaz (Univ. Of Reading), Prasanjit Dash (NOAA)

Reprocessing of long time series of sea surface temperature (SST) is now undertaken by different agencies for a large range of sensors (AVHRR series, (A)ATSR series, MSG etc.) acquiring routine measurements of SST since the early 80s. One of the motivations of the **GHRSST Climate Data Assessment Framework** (CDAF) was to define a set of **common metrics** to assess the **uncertainties**, **stability** and **sensitivity** of these datasets, evaluating their suitability for detecting changes and trends and therefore their use as Climate Data Records (CDR).

To achieve this assessment in a **simple**, **fair** and **consistent** way, we are designing a **open-source prototype** tool that take as input any GHRSST formatted dataset to produce assessment diagnostics, as defined by various groups and projects (CDR TAG, CCI SST), based on comparisons with reference data (GTMBA for temporal stability assessment, drifting buoys and Argo floats for spatial assessment) integrated with the tool. The current prototype of this tool is based on **felyx** and **jupyter notebooks**, providing producers with a complete solution to evaluate the quality of a dataset with respect to these common metrics and to allow them to complete the CDAF (Climate Data Assessment Framework) in a consistent manner. We used here for demonstration *AVHRR Pathfinder v5.3* as a candidate test dataset.

The tool

The CDAF prototype tool was implemented by quickly assembling existing opensource solutions to deliver an end-to-end proof of concept of an integrated system (tools and data) allowing a fair and objective assessment of different candidate CDR datasets.

Software



Felyx (http://hrdds.ifremer.fr) is an open-source framework for geographical subset extraction and match-up production.



The **Jupyter Notebook** (http://jupyter.org) is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.



Match-ups are extracted with *felyx* using a maximum **2 hours** time difference and maximum distance of **5 km**. The in situ measurements include all uncertainty and

The reference data

Several diagnostics in the CDAF are based on comparisons with in situ data, such as moored buoys (from GTMBA), drifters and Argo floats. We used the **ESA SST CCI Independent Reference Data Set (SIRDS)** which covers the 1980-2016 time frame. The compiled data went through extensive quality check by the SST CCI team and we selected here only the measurements with quality level 5. All variables from the in situ dataset are retained in the final match-ups and can be used to refine the « good » match-up selection.



Temporal distribution of drifters and GTMBA measurements. About 100.000.000 measurements were ingested into felyx. On-the-shelf analytics tools such as Grafana allow to inspect instantaneously the spatial and temporal content of massive databases (here Elasticsearch over which felyx is implemented)

Acknowledgement : Thanks to Gary Corlett from University of Leicester and ESA SST CCI project for providing the SIRDS in situ dataset.

Assessment diagnostics

correction variables existing in the source dataset.

>>> felyx-run-mdb --assemble -t cci_drifter cci_gtmba -c ./conf/matchup_cdaf_pathfinder_workflow.cfg -v --backlog 730 --date 20031231

Application to Pathfinder dataset

For the demonstration, the CDAF prototype tool was ran over NOAA Pathfinder v5.3 dataset, spanning from August 1981 to December 2014.



More than 10.000.000 match-ups with CCI In situ Reference dataset (only using GTMBA and drifters) were extracted

	Multi-threaded / distributed mode (3 servers)
1 year	< 1h to 17h
30 years	About 10 days

Maximum processing time in various configurations for Pathfinder processing. It is very dependent on the year because of heterogenous number of in situ measurements. Felyx processing can be distributed over a cluster making the processing time much faster. This is to be considered for instance for providing the CDAF tool as a service on a cloud rather than a software.

The CDAF tool could be either deployed at the CDR producer premices or set-up at a particular facility hosting or downloading the datasets to evaluate.

Conclusions and perspectives

references:

[1] CDR TAG CDAF Requirements for a Tool, J.Mittaz University of Reading
 [2] Climate Data Assessment Framework, CDR-TAG_CDAF v 1.0.5, 4 June 2014, Christopher J. Merchant, Jonathan Mittaz and Gary K. Corlett

Several diagnostics, described in above references, were defined by the GHRSST Climate Data Record working group (CDR-TAG) addressing uncertainties, sensitivity, systematic and non systematic differences or stability. They can be implemented collaboratively as generic functions in **jupyter python notebooks**, and ran over the match-up datasets processed at the previous step. For this prototype, we implemented the **stability** diagnostics, which can now be tested on different datasets and be further refined.



« Currently this is only possible using the GTMBA and the steps required are :

- a)Identify GTMBA locations where buoy data (with postcalibration corrections applied) are available for >75% of the period to be assessed for stability
- 1)The ICOADS measurements should be quality controlled using the ICOADS trimming flags to discard any observation more than 4.5 standard deviations from the
- climatological median. b)Match satellite SSTs to these GTMBA measurements for
- the maximum possible number of complete years c)Calculate the monthly median satellite-GTMBA difference for each location separately
- d)For each month of the year and location, calculate the multi-year average of the monthly median satellite-GTMBA differences
- e)Deseasonalize the monthly median differences at each location by subtracting the result of step 4 for the appropriate month of the year
- f) Calculate the monthly mean difference across all locations. This results in a single satellite-GTMBA SST time series



We demonstrated a generic solution to perform a fair and shared assessment of GHRSST CDR datasets. More generally this demonstrates how using open and shared ressources, reference data and software help build collaborative frameworks for the analysis of datasets from various sensors, and a possible way forward for GHRSST project to foster interactive and collaborative science, and investigate specific task team issues.

Improvements for a final open solution would include :

- Other diagnostics need now to be implemented, tested and refined, as well as the matchup filtering and in situ data correction (through jupyter notebooks)
- Usage of jupyter dashboards for easy publication and intercomparison of each CDR dataset evaluated, and providing a space for sharing and publishing these results
- Some optimizations in the match-up processing (could by reduced by half)
- Packaging a complete deployable solution on dockers including software and in situ data or setting up a central service in a candidate host (*software as a service* infrastructure)

We propose to iterate on this interactively and collaboratively working remotely on these preliminary jupyter notebooks and pre-processed match-ups, hosted at Ifremer. Other CDR datasets can be added too. Match-up datasets and notebooks are also available for download for your local usage. g)Find the least squares fit (straight line) to the time series of monthly mean differences
h)Quote the 95% confidence interval on the slope of the fit as the stability measure »

Jupyter Stability Assessement vs GTMBA Last Checkpoint: II y a 20 heures (autosaved)		Logo
File Edit View Insert Cell Kernel Widgets Help	Trusted	felyx
B + ≫ 42 B + ↓ H Run ■ C → Raw NBConvert ■		





```
(39943 7)
```

```
In [350]: # calculate the monthly median satellite-GTMBA difference for each location separately
monthly_median = matchups.groupby(
    [matchups['id'], matchups['date'].dt.year,matchups['date'].dt.month]
    )[['sst diff']].median()
```

l]:
identify GTMBA locations where buoy data are available > 75% of the period to be assessed for st
nb months perbuoy = monthly median.groupby(level=0).count()

```
selected_buoys = nb_months_perbuoy[nb_months_perbuoy['sst_diff'] > int(assessment_timeframe * 0.75)].ind
print "Selected buoys with at least {} months of match-ups :".format(int(assessment_timeframe * 0.75))
for b in selected_buoys:
    print b
```

Selected buoys with at least 180 months of match-ups : moored buoy [GTMBA] 52011 moored buoy [GTMBA] 52012 moored buoy [GTMBA] 52084 moored buoy [GTMBA] 52086 moored buoy [GTMBA] 52302 moored buoy [GTMBA] 52307 moored buoy [GTMBA] 52317 moored buoy [GTMBA] 52318 moored buoy [GTMBA] 52319

In [352]: # calculate the multi-year average of the monthly median satellite-GTMBA differences for each location separate
multiyear_monthly_mean = monthly_median.groupby(level=[0,2]).mean()

In [353]: from matplotlib import pyplot as plt

