SENTINEL-3A CAL/VAL ENVIRONMENT AND APPLICATIONS

Jean-François Piollé, Ifremer
(Eumetsat/Copernicus visiting scientist)

Igor Tomazic, Anne O'Caroll
Eumetsat

« exploitation platform concept » for cal/val

Scientific framework and data

- Collection of datasets (Sentinel-3, reference, other missions)
- ↓ Abstraction layer (readers, models : Cerbere)
- ↓ Middleware for search, indexing, extraction, intercomparison, visualization)
- ↓ Validation, Data analysis, Science, Reporting

Processing Environment

- Regular update of datasets through dataflow management
- Framework for automatic background information processing, extraction and indexing – Automation of tasks

  Distributed processing

Prototype for a consistent open source and cloud based « thematic exploitation platform », cal/val oriented

GENERAL S3 CAL/VAL FRAMEWORK

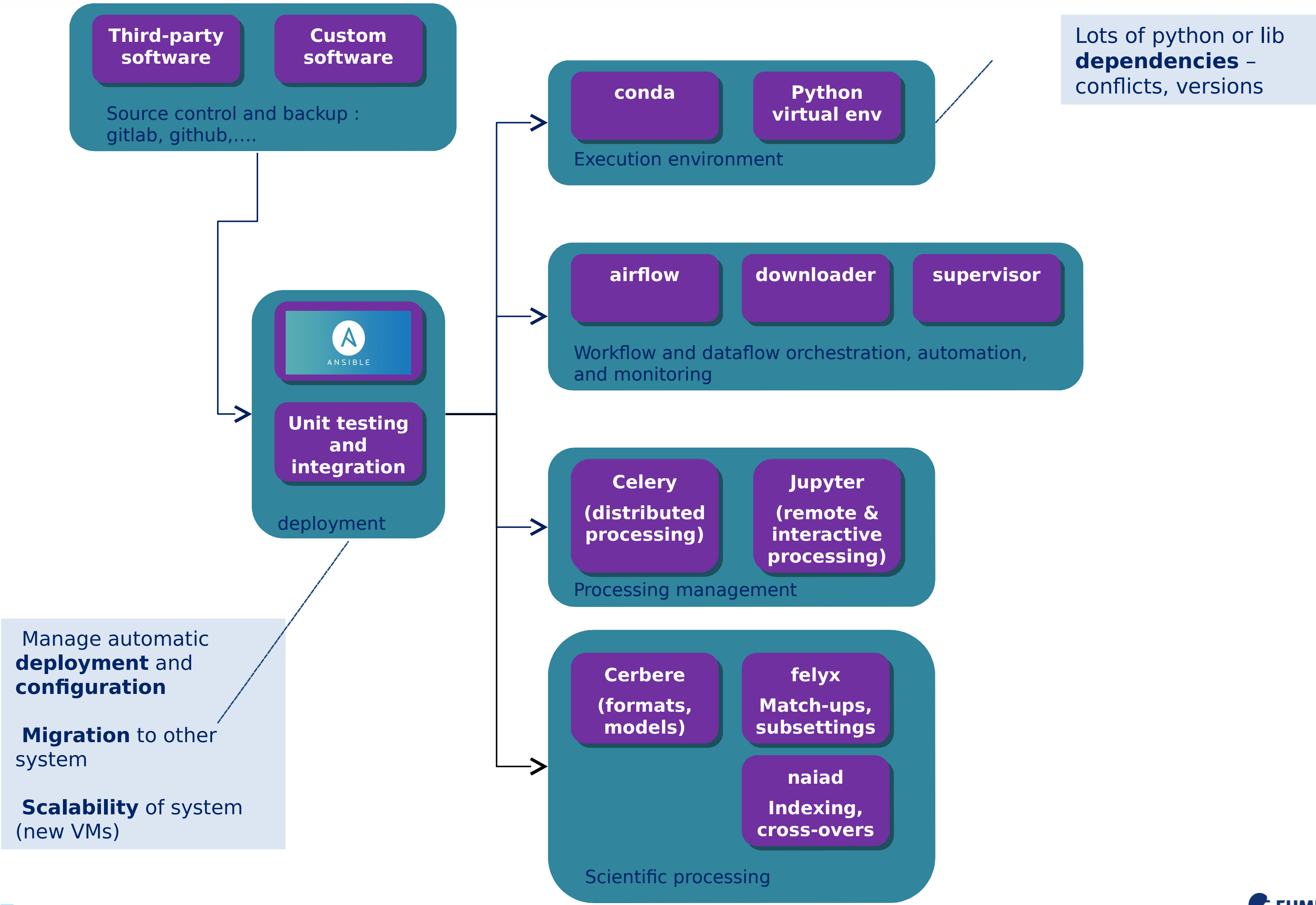
Cloud infrastructure

- Remote platform for Sentinel-3 cal/val
- Virtual desktop

- Dedicated cloud for Sentinel-3 cal/val, shared by SST, ocean colour and altimetry subgroups

- Specs
  - 9 VMs (64 GB RAM, 8 cores)
  - storage : 1 PB
  - Ubuntu

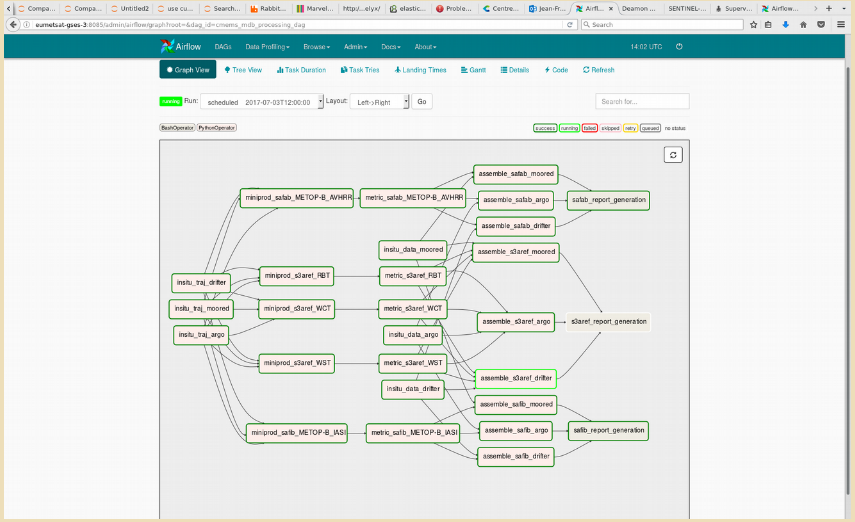
Building the cal/val environment



EUMETSAT

Workflow management : Airflow

- Task scheduler
- Schedules and runs processing workflows
  - Handle dependencies, conditions
  - Workflows implemented in python
  - Tasks distributed on different VMs (celery framework)
  - Alert through email and web interface
  - Quite complex to handle at first but very powerful framework for running and monitoring background tasks – time saver on the long run!!
- Deployed and installed on GSES with Ansible, controlled by supervisor
- High availability, very effective
- Absolutely essential for routine workflows



https://airflow.incubator.apache.org

EUMETSAT

Managing dataflows

Sentinel-3A data	versions
SL_1_RBT___	REF : reference – internal OPE : operational version REP : reprocessings (two) NRT : near real time NTC : non time critical Multiple IPF releases
SL_2_WCT___ (internal)	
SL_2_WST___	
OL_1_ERR___	
OL_1_EFR___	
OL_2_WRR___	
OL_2_WFR___	
SR_2_WAT___	

The data to assess

Comparison data	Source
OSI SAF METOP-B AVHRR L2P	OSI SAF (Ifremer)
OSI SAF SEVIRI L2P	OSI SAF (Ifremer)
OSI SAF METOP-B IASI L2P	OSI SAF (Ifremer)
OSI SAF METOP-A IASI L2P	OSI SAF (Ifremer)
NOAA VIIRS L2P	NOAA
METOP-B IASI L1	Eumetsat
METOP-B IASI L2	Eumetsat

Similar products / sensors to compare with

Reference data	Source
OSTIA	UK Met office
CMC	PO.DAAC
OSI SAF MDB NRT	Ifremer
CMEMS in situ data	Ifremer
Radiometer data	NOC
MOBY	NOAA
Aeronet	NASA
Rephy	Ifremer

The « ground truth »

Multiple dataflows to maintain updated

Multiple sources for download to monitor

Scripts + more advanced tools for easier management

Dataflow management :« downloader »

Maintaining complete up-to-date collections of reference and S3A datasets

Ensure completeness, continuous update / replacement

Store data in a common organization (possibly different from provider)

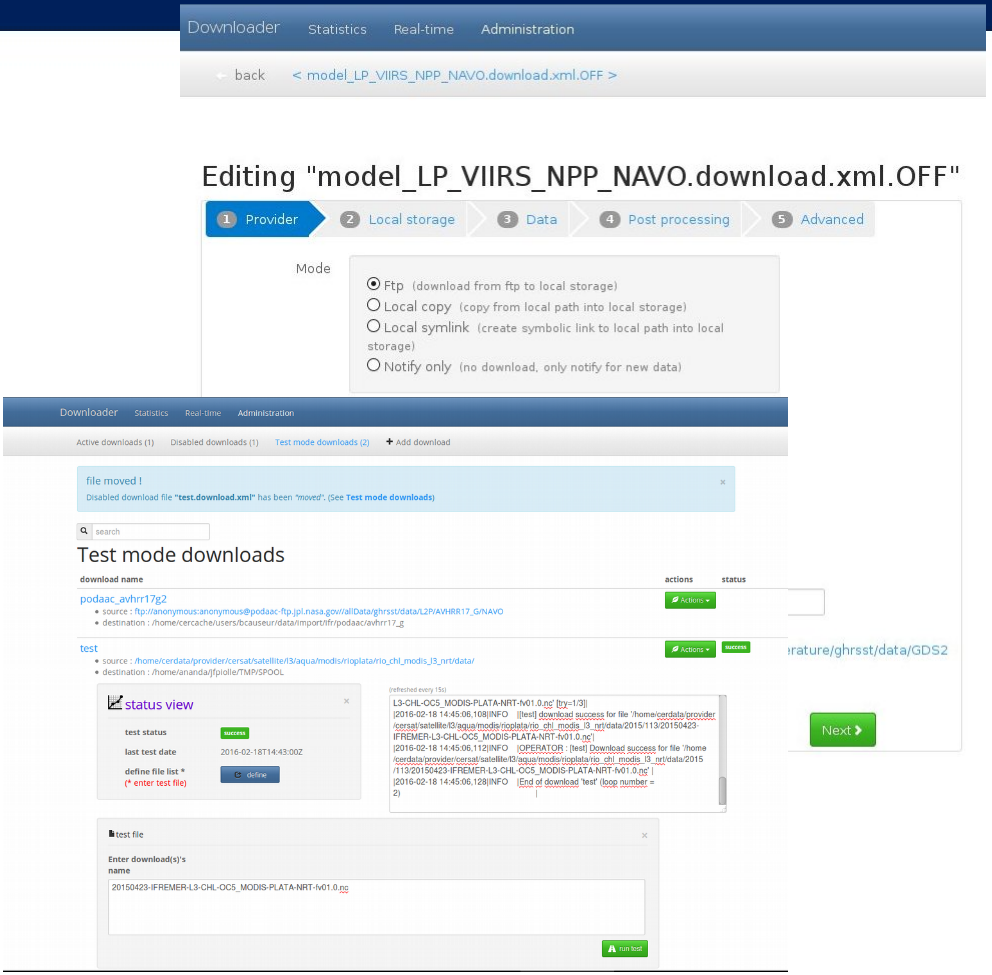
Connect to processing workflows (notifications, spools)

Purge data history (rolling archive) without impact on download

Ensure data integrity (use of checksum when provided)

Monitor data source (access, availability of the data) and raise issues

No on-the-shelf tool existing with enough intelligence to address all this issues



Documentation : https://docs.google.com/document/d/1G8RBiAI3hBoJchjUcgj90qFGa0DPYb1OJBHKQ4A0jzE/edit?usp=sharing

Source code : https://git.cersat.fr/felyx/downloader

EUMETSAT

CALV/VAL TOOLS AND APPLICATION MIDDLEWARE

Cerbere : data abstraction layer in python

- Generic **python API** to access and describe file content (different data formats) and observation patterns
- Abstract layer to build generic tools and applications upon it
- Implemented at Ifremer, also access layer for software like felyx, naiad, cal/val tools and routines
- Generic data file model (similar to netCDF) – **mapper** :
  - Standard geolocation dimensions : row/cell, x/y, lat/lon, time,…
  - Other dimensions
  - Standard geolocation fields
  - Instrumental / geophysical fields : multi-dimensional arrays (incl.)
    - Variables attributes : no explicit scale factor, transformation performed in memory
  - Metadata (global attributes)

- Generic observation patterns - **datamodel**
  - Swath, Grid, Trajectory (along-track) , Image, TimeSeries, GridTimeSeries,….
  - Generic functions
    - save : format to similar format (dimensions, global attributes, etc…) any data following the same observation pattern
    - extraction of subsets, etc…
- Complemented by some companion packages
  - mappers for other formats (Sentinel-3/SAFE, IASI/EPS)
  - generic packages based on the cerbere datamodel concept : ancillary fields, display, resampling/interpolation, ocean parameter calculation,

Doc/tutorial : http://cerbere.readthedocs.io/en/latest/
https://git.cersat.fr/groups/cerbere

EUMETSAT

« big data » technologies

Elasticsearch : nosql type of database (alternative solution to SOLR, Cassandra, Hbase,…), with geospatial extensions : used for geospatial information indexing and search

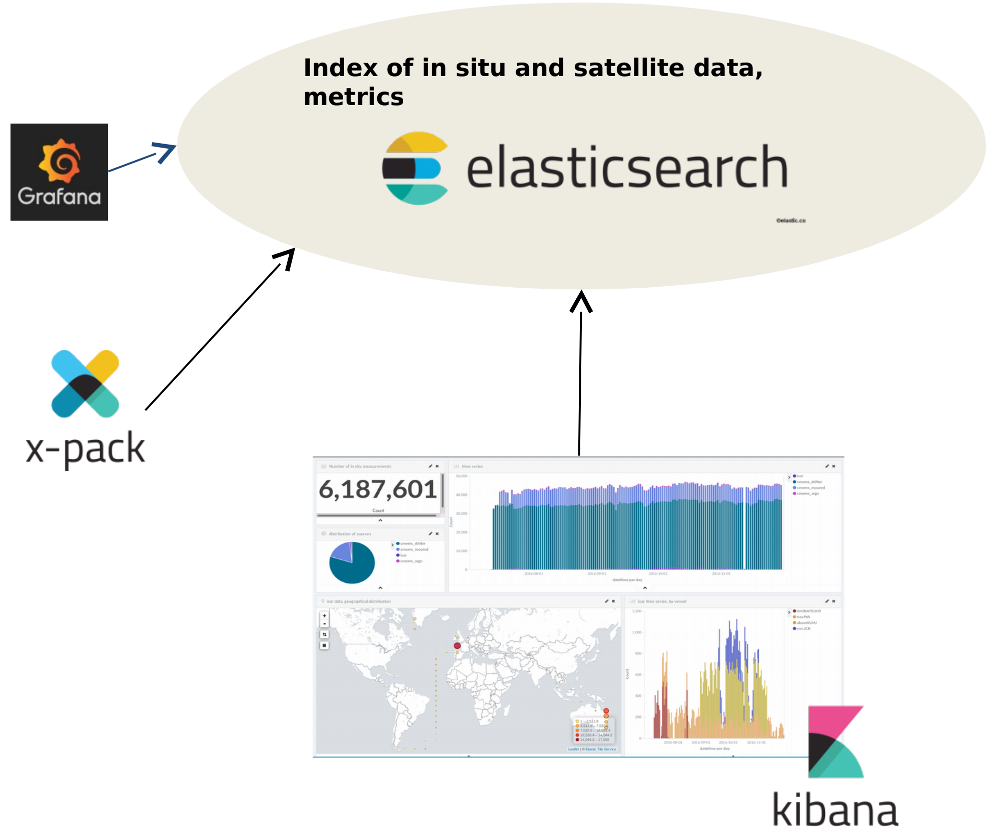
Take advantage of distributed environment (here GSES VMs)

Several third-party tools and analytics tools to leverageits full power)

Analyzing, alerting, finding patterns

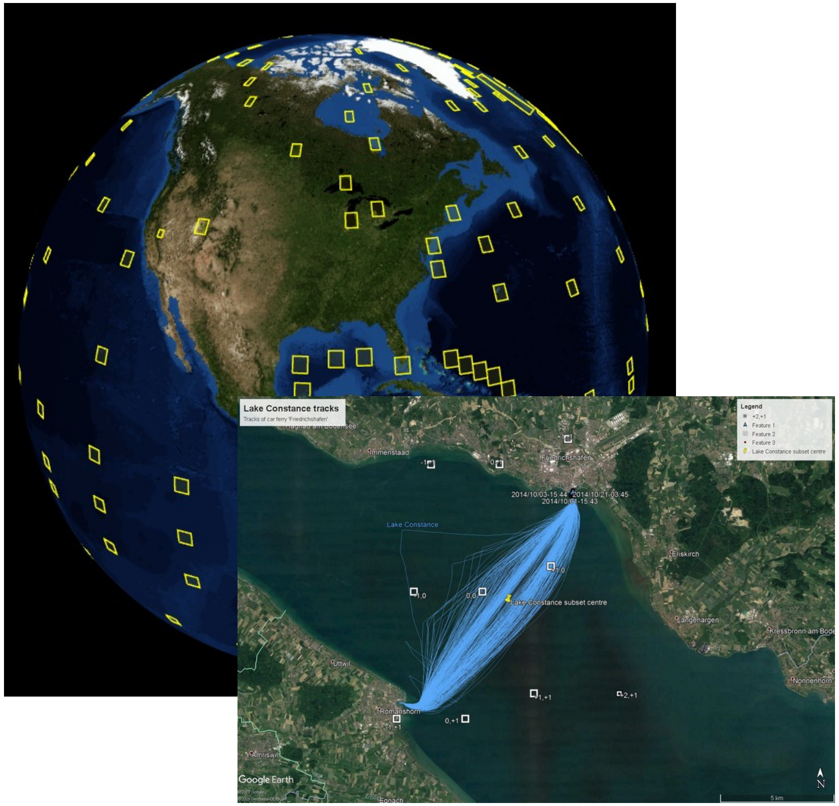
Barely scratching the surface now : machine learning, …

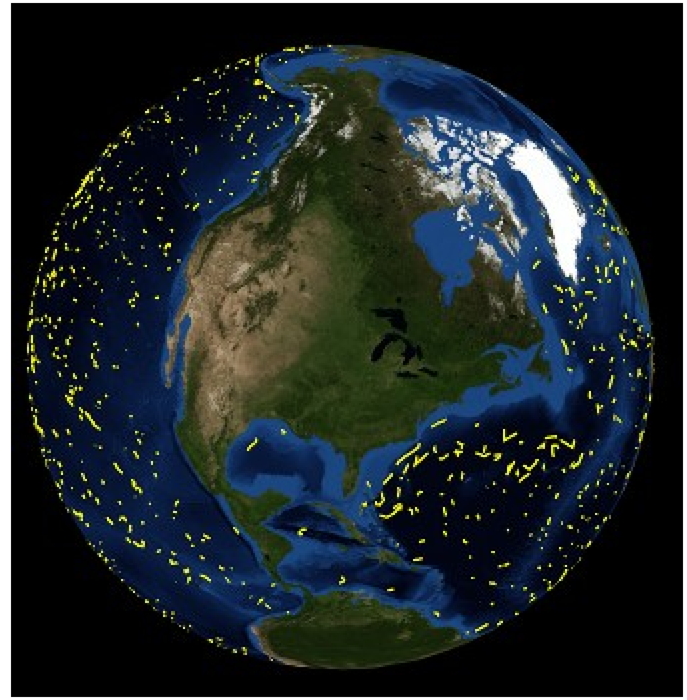
Two main tools based on : **Felyx** and **Naiad**



CALV/VAL TOOLS AND APPLICATION MATCH-UP DATABASES WITH FELYX

Felyx : static vs dynamic sites





Static sites

Extraction of time series

« dummy match-ups » => multi-sensor match-ups with no reference in situ measurement

Local case studies => lakes

Dynamic sites

Match-ups with moving platforms (buoys, ships)

Does allow multi-sensor match-ups too but anchored to in situ measurement

SLSTR MDBs

In situ sources

- Benefit on general frameworks:
  - CMEMS
    - Integration with Copernicus/CMEMS service for the provision of moored and drifting buoys and Argo data : collection and availability of all data in the same format and quality control
    - Canadian & european GDACs for surface drifters being created
    - Expected improvements in quality control and metadata

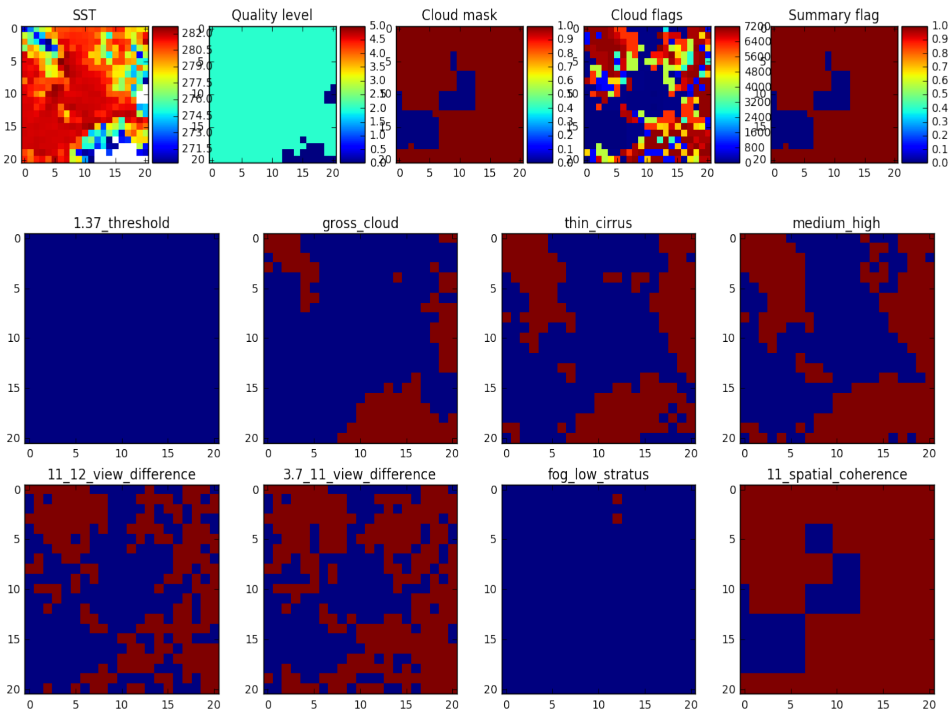
  - in situ radiometer
    - http://www.shipborne-radiometer.org/
    - High quality data
    - Common format and content has been agreed
    - Shared repository will be soon available assembling all these data
    - Currently used in felyx : cruises from ABoM, NOC, RSMAS and DMI

fiducial reference temperature measurements

- All these data formatted in felyx format and available on ftp for ingestion into other MDB (request jfpiolle@ifremer.fr)
- Felyx + in situ data : framework for consistent MDB production for each GHRSST product

Content of match-up

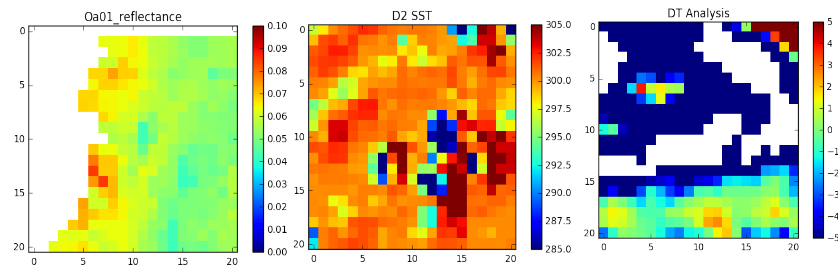
All fields from RBT (L1), WCT and WST (L2) traceability



More than 600 variables from L1 to L2…..

21x21 boxes extracted with all fields for each match-up can be used to test and assess new algorithms or post-processing on a larger scale and time period in a fast way, with in situ information to directly estimate the improvement.

All fields from cross-overs and complementary files



Cross-over fields from OLCI, METOP, VIIRS

Complementary files from post processing of match-ups (prototype SST, quality level, etc…)

Ancillary fields (OSTIA dSST)

In situ buoy history centered on match-up

Existing match-up databases

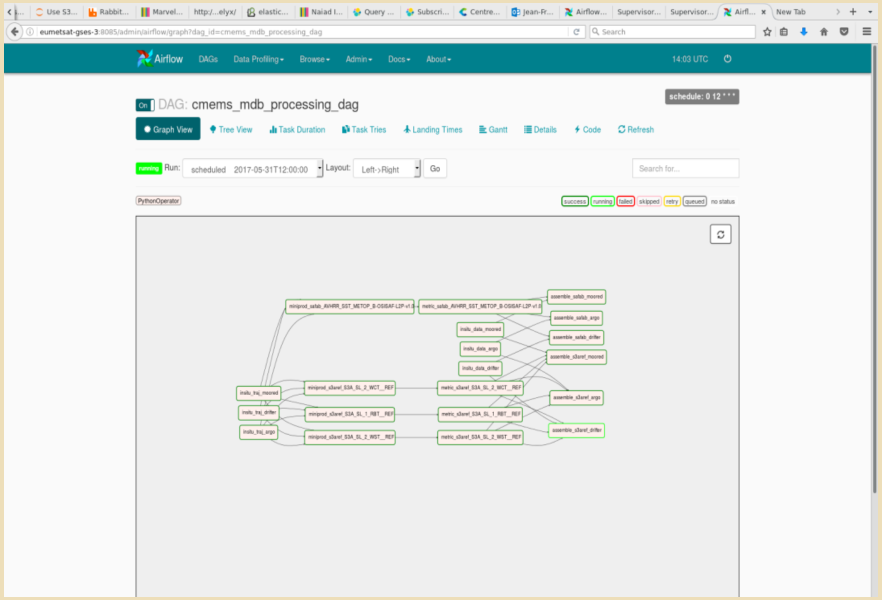
Match-up database	Primary products	Complementary products	Availability
OSI SAF SLSTR MDB	SLSTR NRT products	METOP (about 50%) SEVIRI (about 30%) SST prototype OSTIA	July 2016 - present
Eumetsat reprocessed SLSTR MDB	SLSTR REP v4	OLCI SST prototype OSTIA	July – Nov 2016
Eumetsat reprocessed MDB	SLSTR REP v5	SST prototype OSTIA	Nov 2016 (-*April 2017)*
Eumetsat IASI MDB	Eumetsat & OSI SAF L2P METOP-A METOP-B IASI		June 2017 - onward
Eumetsat METOP-B MDB	OSI SAF L2P METOP-B AVHRR		June 2017 – onward

SLSTR MDB alternative configurations

- Challenge of MDB production is the collection and download of the input data, especially with SLSTR
- All product levels, operation and reference versions, reprocessed dataset
- Sentinel-3A and Sentinel-3B
- All SLSTR MDBs to be processed on EUM S3 cal/val environment

Processing baseline	Timeliness	MDB production
OPE	NTC	Continuous update, full time series
	NRT	On demand, to quantify difference with NTC, ~one month
REF	NTC	On demand, for validation of new PB
	NRT	On demand, for validation of new PB
Reprocessing	NTC	On demand, complete or partial time series

Processing / integration into operation

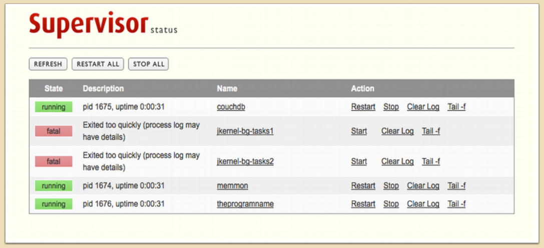


Airflow (https://airflow.incubator.apache.org) is a task scheduler

Processing workflow, from in situ data ingestion to match-up assembling, can be integrated in such system for automated MDB production

Provides automation, control and monitoring

Routine processing



Felyx daemons supervised and run by **supervisor**

http://supervisord.org

supervision

```
felyx-run-mdb  --miniprod --metric --ancillary --assemble  -t all -c cmems_mdb.cfg -v --backlog 0 -l 20180306.log --date 20180306
```

Command-line script to run the complete MDB process for a time range

Several options to run specific steps or configurations

On-demand processing

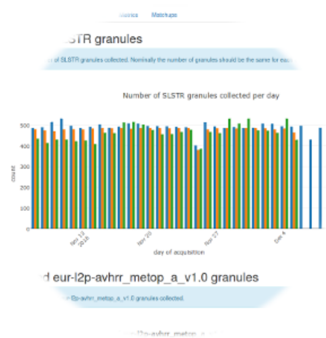
SENTINEL 3

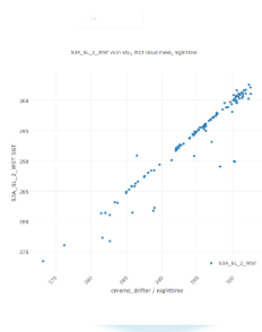
Sentinel-3A SLSTR Match-Up Database

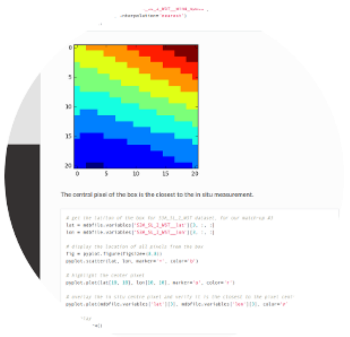
Homepage of Ocean and Sea-Ice Satellite Application Facility (OSI SAF) match-up database for Sentinel-3A SLSTR instrument

This site gives access to colocations between all level SLSTR products and various sources of reference in situ measurements, completed with other third party mission measurements when close enough to SLSTR measurements. It is used for the validation and monitoring of SLSTR products. This project if funded by Eumetsat. The access to these data is restricted and must be requested to jfpiolle@ifremer.fr.

Note: interative plots are best viewed with Chrome navigator.







Production reports

Statistics on the production of the match-ups with respect to input satellite and in situ data, to monitor completeness of match-up production and processing errors.

View latest >>

Analysis reports

Basic daily comparison statistics between SLSTR and in situ sea surface temperature measurements, as computed from the match-ups

View latest >>

Data access

How to access the data, python tools and tutorial to start with the match-up files.

View details >>

The S3A SLSTR MDB is produced at Ifremer / CERSAT for OSI SAF project funded by Eumetsat, using felyx ESA tool.

Tutorials and notebooks

s3analysis
1.0

Search docs

Sentinel-3A MDB databases
s3analysis package
pyfelyx
s3analysis
Reading and using the SLSTR MDB files
Comparison of WST SST in situ using WCT cloud mask
Investigate outliers in WST/WCT comparisons with in situ
Investigate a match-up
Generate diagnostics over the SLSTR MDB content

Docs » Welcome to s3analysis's documentation!   View page source

Welcome to s3analysis's documentation!

Overview

- Sentinel-3A MDB databases
  - Sentinel-3 SLSTR MDB
- s3analysis package
  - Overview
  - Installation

Reference

- `pyfelyx`
- `s3analysis`

Tutorials

- Reading and using the SLSTR MDB files
  - basics
  - understanding the in situ section of the matchups
  - understanding the satellite part of the matchups
  - traceability
  - some advanced functions with `s3analysis`
- Comparison of WST SST in situ using WCT cloud mask
  - getting sst and cloud information
  - Compare with in situ
    - night time statistics
    - day time statistics
  - spatial coherence test
- Investigate outliers in WST/WCT comparisons with in situ
  - interactive display
- Investigate a match-up
  - trace back to original file
- Generate diagnostics over the SLSTR MDB content

Jupyter notebooks for above tutorials can also be downloaded here:

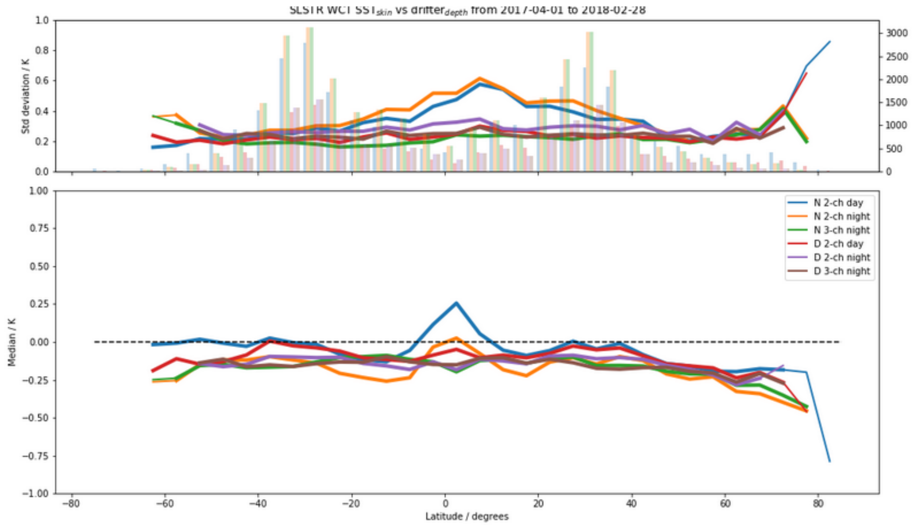
- Using SLSTR MDB files
- Comparison between WST and in situ using WCT mask
- Investigating outliers
- Generate diagnostics

- Content of the match-up is quite complex
- Python package providing :
  - Abstraction over individual files (virtual file)
  - Handy pre-built selection filters
  - Diagnostic plots
- Demonstrated with jupyter notbooks
- Shared methodology for match-up analysis and diagnostic production
- Reproductible results

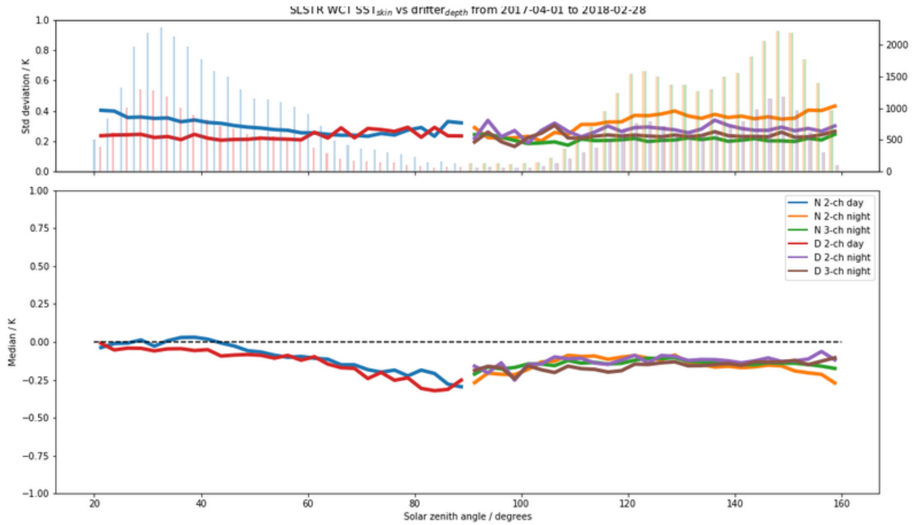
Assessment diagnostics

Dependency diagrams

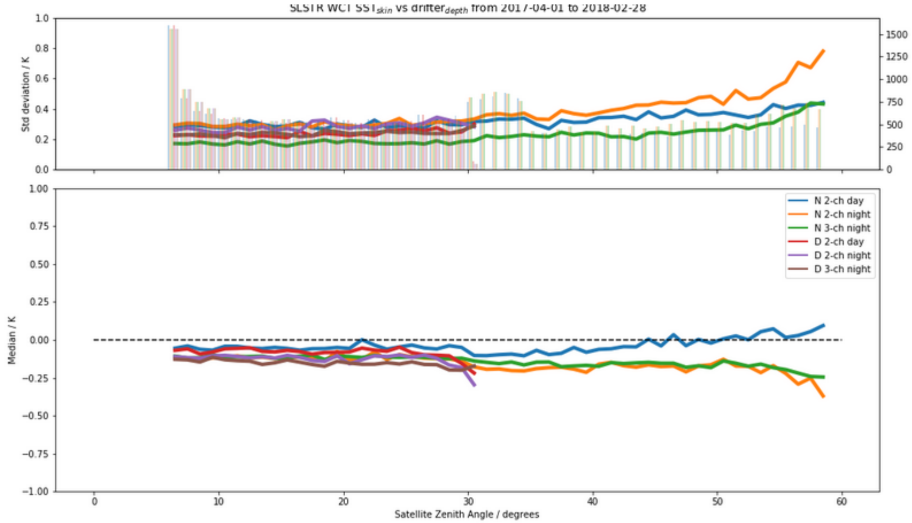
Latitude



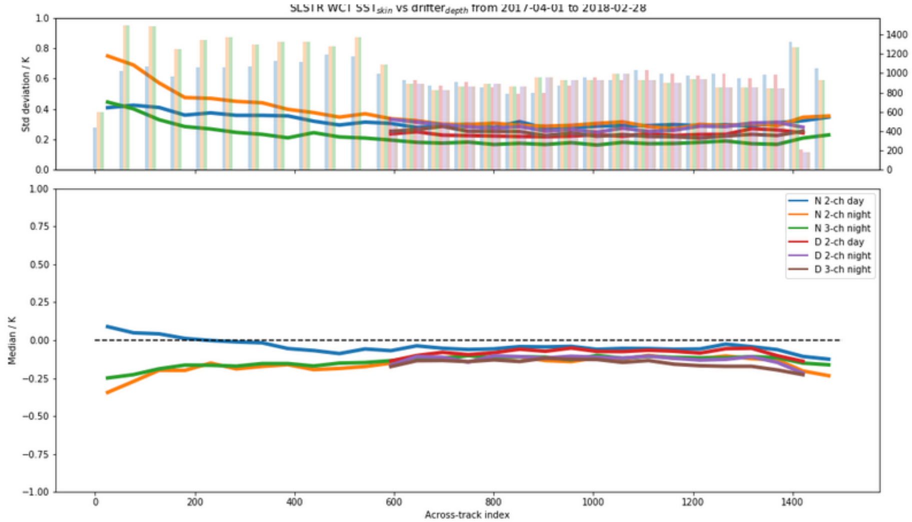
Solar Zenith Angle



Satellite Zenith Angle



Across Track Index



Wind Speed

SLSTR WCT SST$_{skin}$ vs drifter$_{depth}$ from 2017-04-01 to 2018-02-28

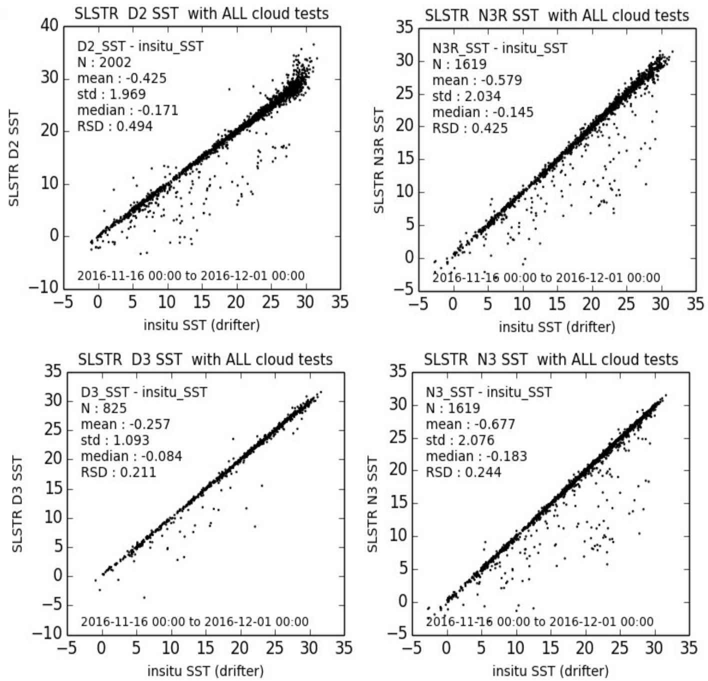
Total Waver Vapour Content

SLSTR WCT SST$_{skin}$ vs drifter$_{depth}$ from 2017-04-01 to 2018-02-28

Application of SLSTR MDB(s)

- Used by different groups at Eumetsat, within S3VT and MPC Sentinel-3

- Major asset in:
  - L1 cloud screening validation (RAL)
  - L2 algorithm/cloud screening improvements
  - L2 Quality level stratification and uncertainties estimation (Univ. Of Leicester)
  - SST validation : OSI SAF (Meto-France / DMI / MetNo), NOAA, Eumetsat
  - Metis intercomparison framework



(WCT_SST – insitu_SST) with cloud clearing REC

algo	N	mean	std	median	RSD
N2	5293	-1.520	3.449	-0.330	0.560
N3R	1911	-0.617	2.098	-0.146	0.430
N3	1911	-0.727	2.165	-0.192	0.245
D2	2653	-0.735	2.654	-0.188	0.561
D3	966	-0.294	1.179	-0.096	0.209

26586 cases, 5299 clear cases 19.9 %

(WCT_SST – insitu_SST) with cloud clearing ALL

algo	N	mean	std	median	RSD
N2	4130	-0.973	2.608	-0.296	0.479
N3R	1619	-0.579	2.034	-0.145	0.425
N3	1619	-0.677	2.076	-0.183	0.244
D2	2002	-0.425	1.969	-0.171	0.494
D3	825	-0.257	1.093	-0.084	0.211

26586 cases, 4133 clear cases 15.5 %

No correction : skin WCT SST vs bulk insitu SST

Courtesy: Anne Marsouin, Meteo-France

EUMETSAT

CALV/VAL TOOLS AND APPLICATION

NAIAD : DATA INDEXING AND SEARCH

Naiad

- Intended for **satellite to satellite cross-over** detection
- geo-indexing of observation data as temporally bounded geographical shapes
- Command line or API based
- **Main functions**
  - Search file or file subset wrt multiple criteria : spatial, temporal, properties and metadata
  - Cross-search in different datasets, with time window constraint (cross-overs)
- **Main outputs**
  - List of file subsets (file name, indices)

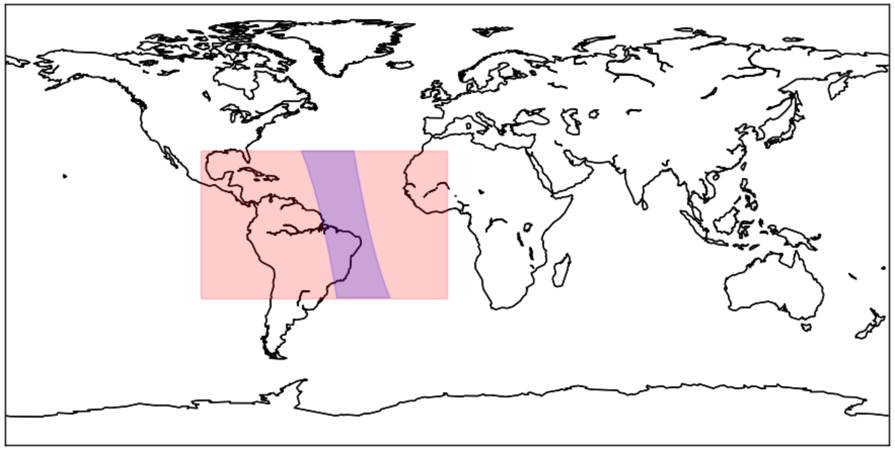
http://naiad.readthedocs.io/en/develop/

Naiad ENHANCED SATELLITE ARCHIVE DATAMINER

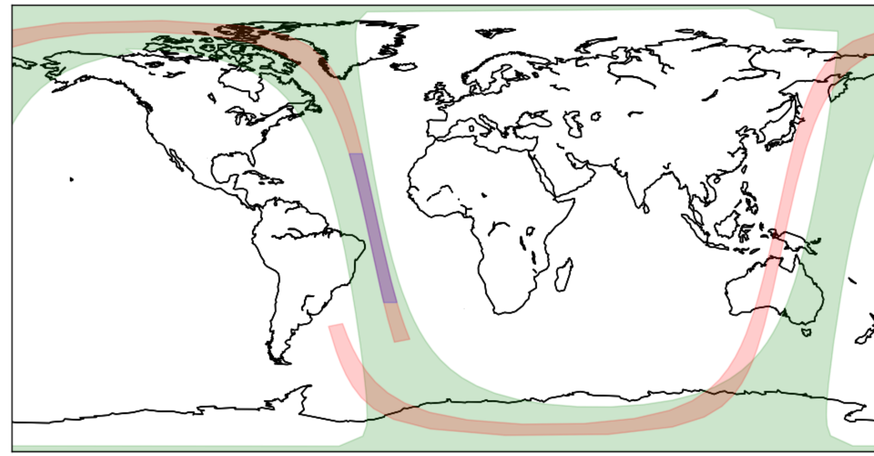
Naiad - queries

Command line tools or python API

Simple search



Cross-over search



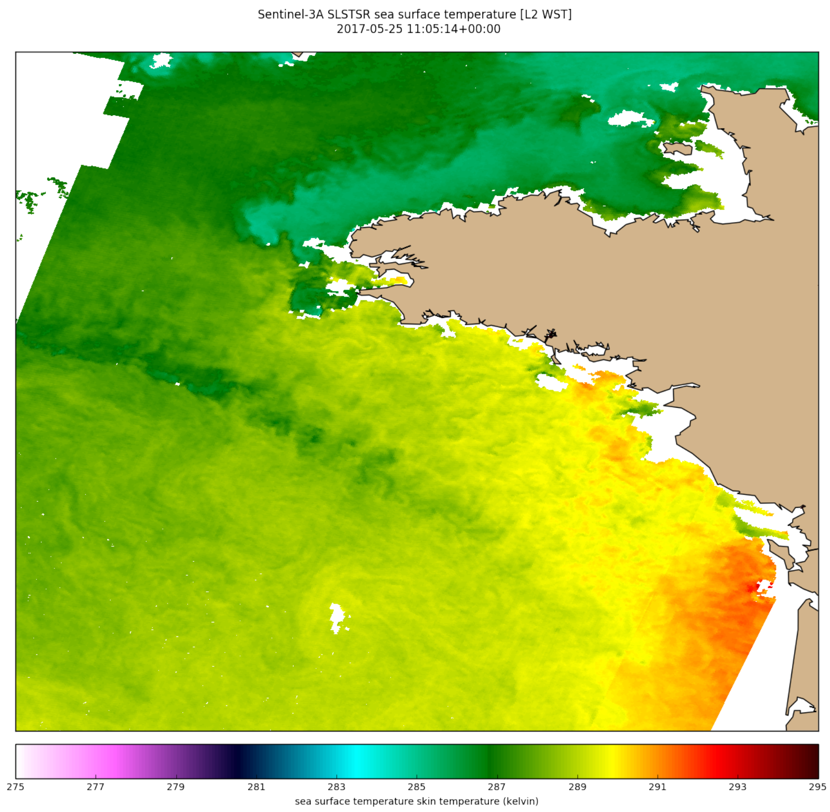
Simple search

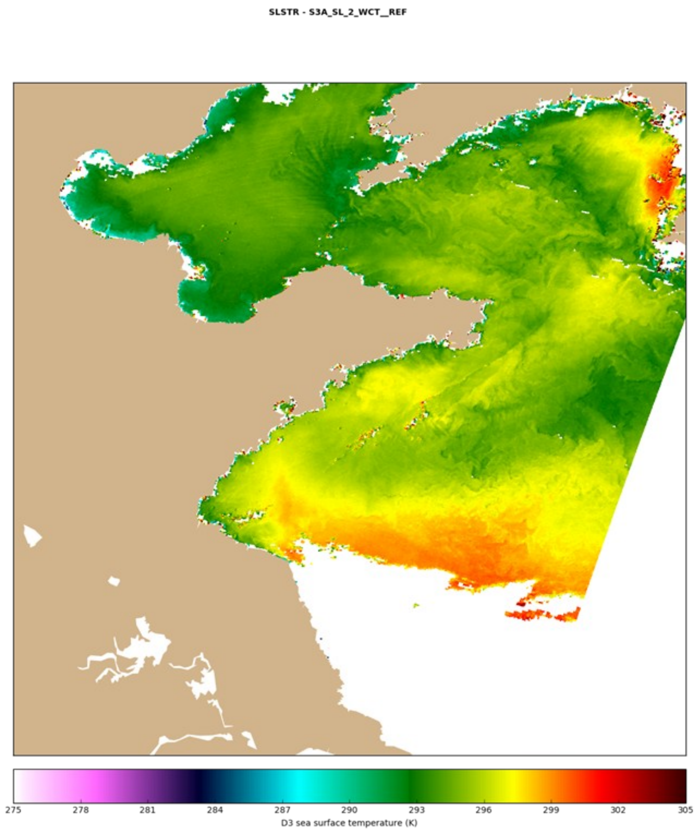
product(s)
time and space criteria
constraints on properties

```
SHOW 0 8
W_XX-EUMETSAT-Darmstadt,HYPERSPECT+SOUNDING,MetOpA+IASI_C_EUMP_20100701004153_19184_eps_o_l1.nc

Reference
Name      : W_XX-EUMETSAT-Darmstadt,HYPERSPECT+SOUNDING,MetOpA+IASI_C_EUMP_20100701004153_19184_eps_o_l1.nc
Time range: 2010-07-01 01:45:32 to 2010-07-01 02:04:44
Slice     : {'cell': slice(40, 119, None), 'row': slice(477, 621, None)}
Geometry  :POLYGON ((-74.3743285021517 30, -68.90709065955365 30, -68.65699768066406 29.06100082397461, -60.12200164794922 -9.196999549865723, -56.58900070190
43 -23.4950008392334, -54.69640015258144 -30, -60.13921621269841 -30, -62.56700134277344 -19.75699996948242, -70.01499938964844 13.71700000762939, -72.4209976
1962891 23.25600051879883, -74.3743285021517 30))
Crossover :
Name      : 20100701-ATS_NR_2P-UPA-L2P-ATS_NR__2PNPDE20100701_020310_000046842090_00432_43572_3066-v01.nc
Time range: 2010-07-01 02:08:32 to 2010-07-01 02:27:22
Slice     : {'cell': slice(0, 511, None), 'row': slice(2152, 9684, None)}
Geometry  :POLYGON ((-74.3743285021517 30, -68.90709065955365 30, -68.65699768066406 29.06100082397461, -60.12200164794922 -9.196999549865723, -56.58900070190
43 -23.4950008392334, -54.69640015258144 -30, -60.13921621269841 -30, -62.56700134277344 -19.75699996948242, -70.01499938964844 13.71700000762939, -72.4209976
1962891 23.25600051879883, -74.3743285021517 30))
```

Results as images, text or json document

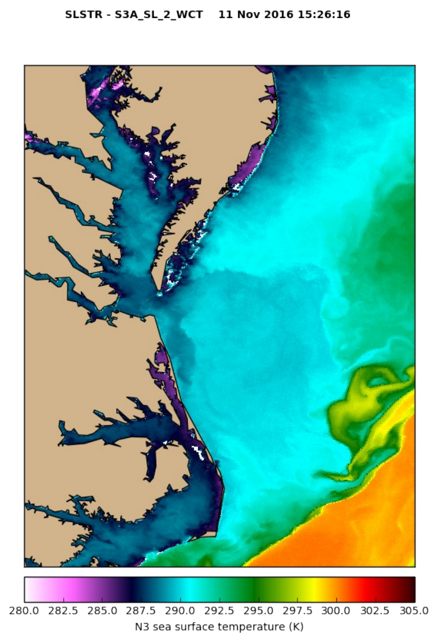
Use case : clear sky image search

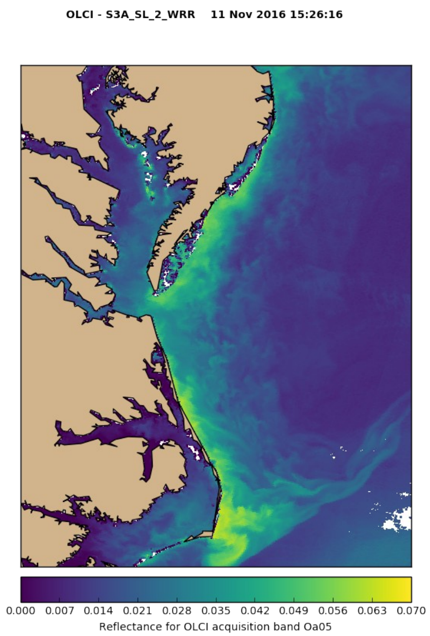


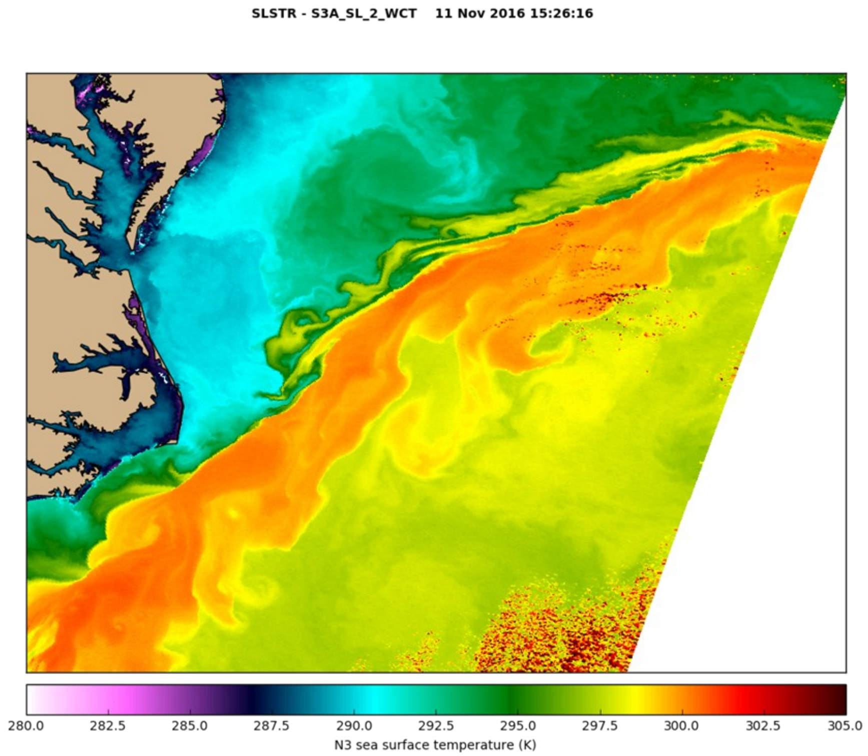


Search based on metadata registered in Elasticsearch

Use case : scene selection, comparison







- clear sky image search
- cross-overs between SLSTR / OLCI or other pairs of sensors

EUMETSAT

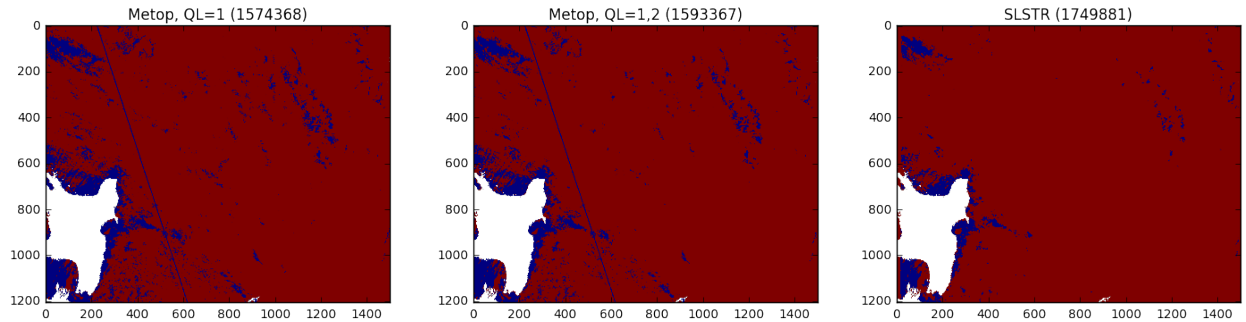
Use case : external cloud mask

Assess quality of SLSTR L1/L2 cloud mask by direct comparison with cloud mask provided by other instruments, resampled on SLSTR image grid

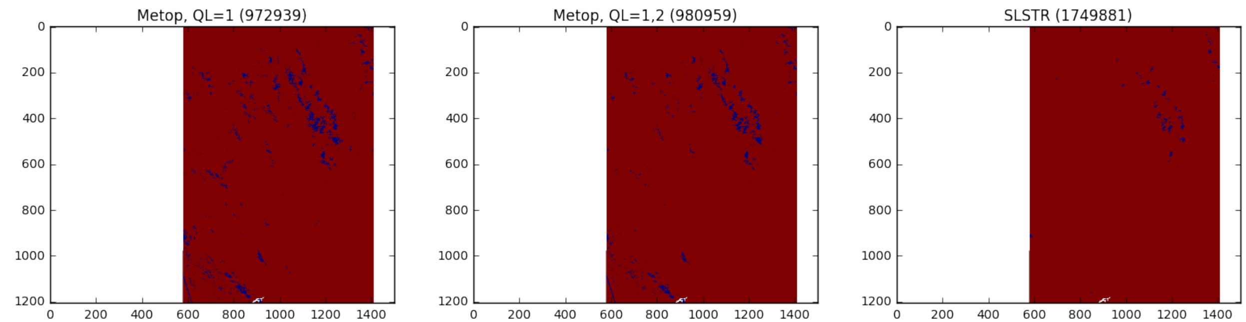
We construct a new cloud mask for SLSTR by remapping the cloud mask from Metop-B cross-overs onto SLSTR granules.

Two masks are generated : native cloud mask from Metop-B (QL=1) and GHRSST mask (QL=1,2)

Intendend for statistical comparison of respective mask extent (NOT at pixel level as clouds may be shifting)



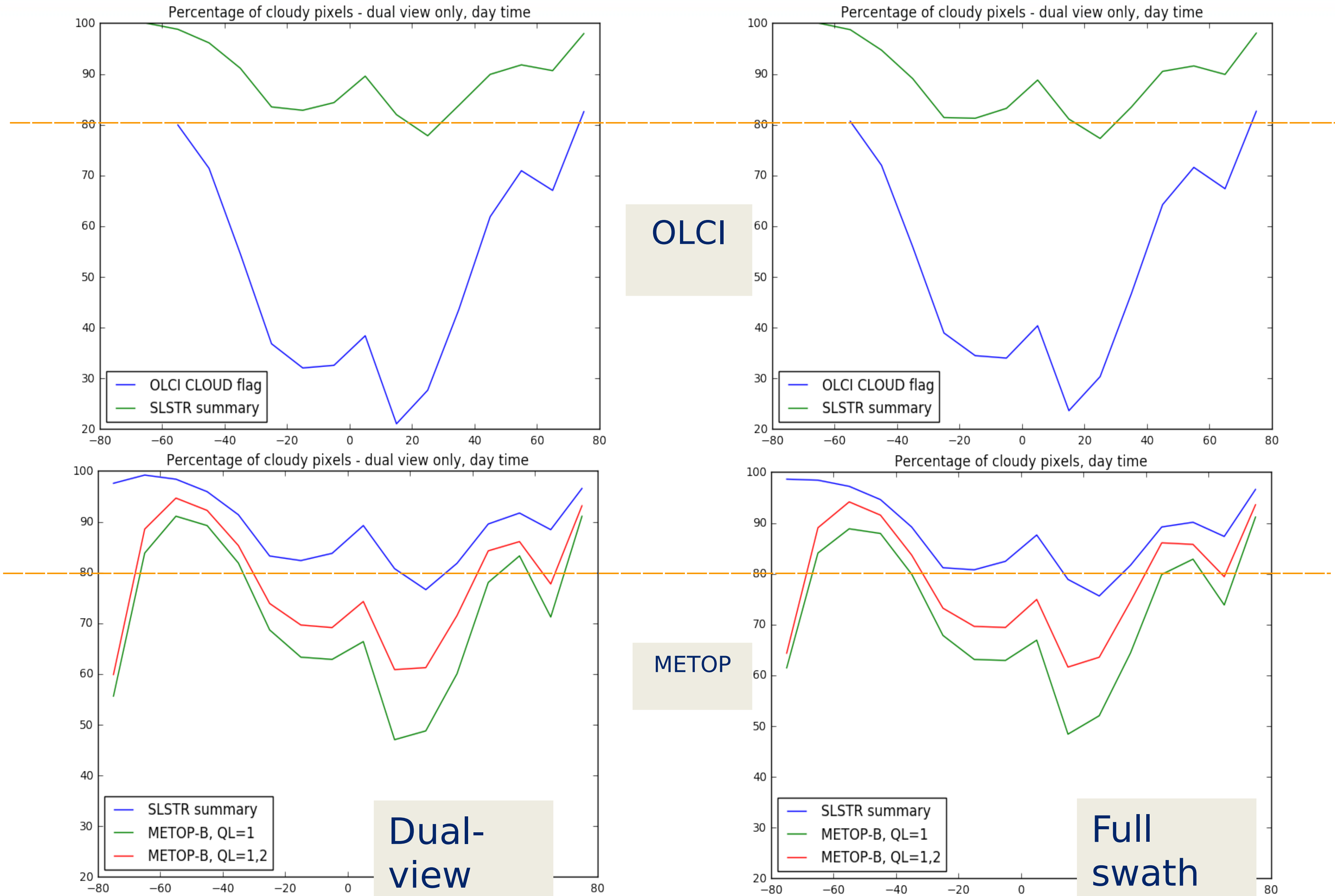
Full swath mask



dual view part of the swath

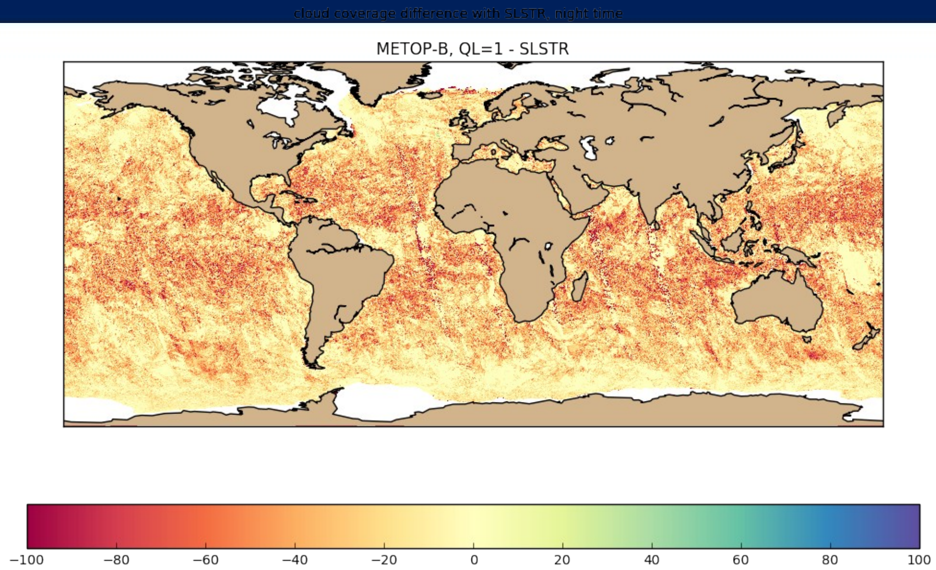
EUMETSAT

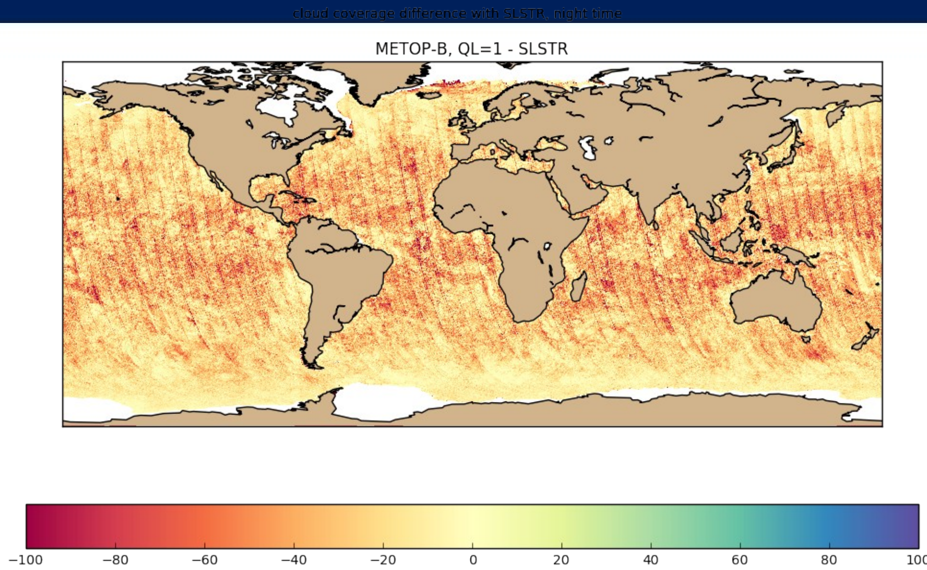
Comparisons of SLSTR with OLCI and Metop cloud masks, per lat

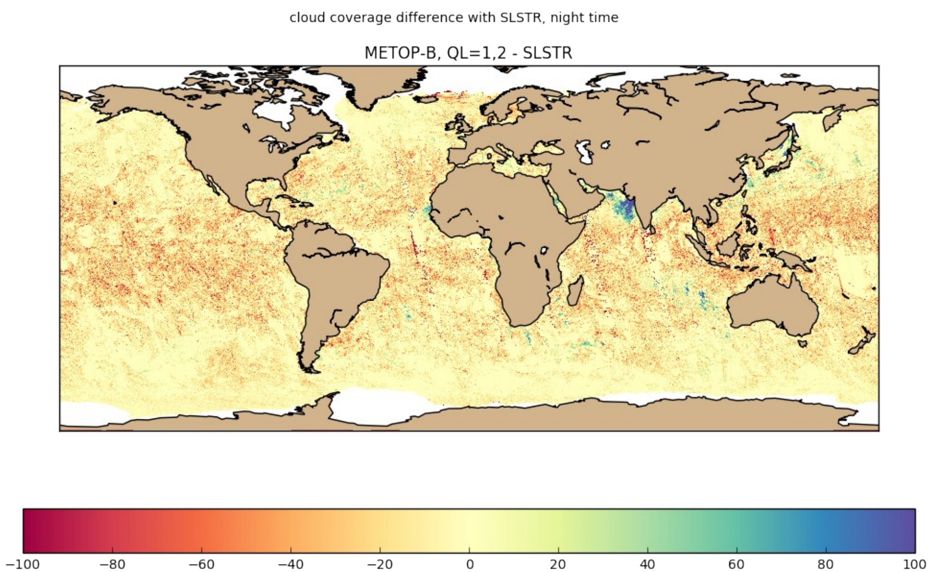


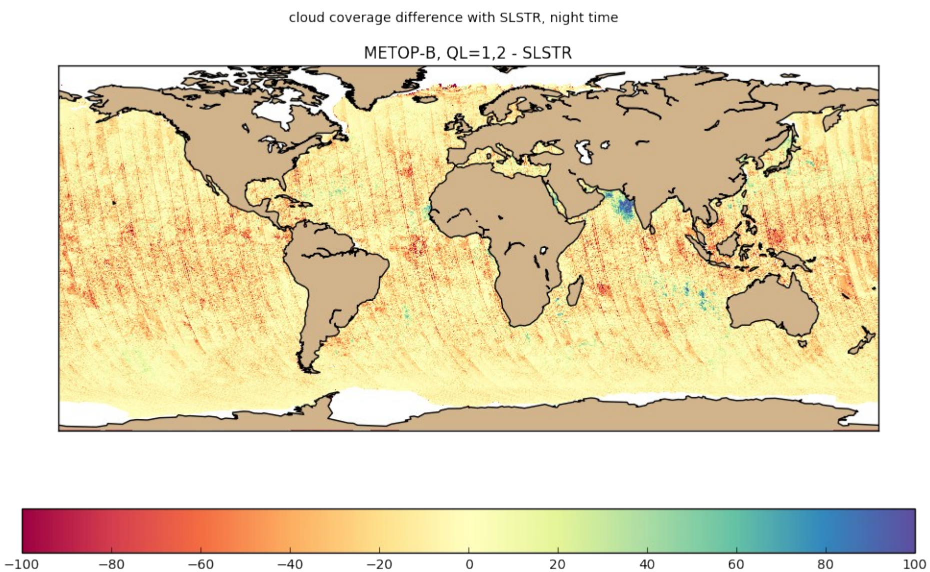
EUMETSAT

SLSTR / Metop-B, night time





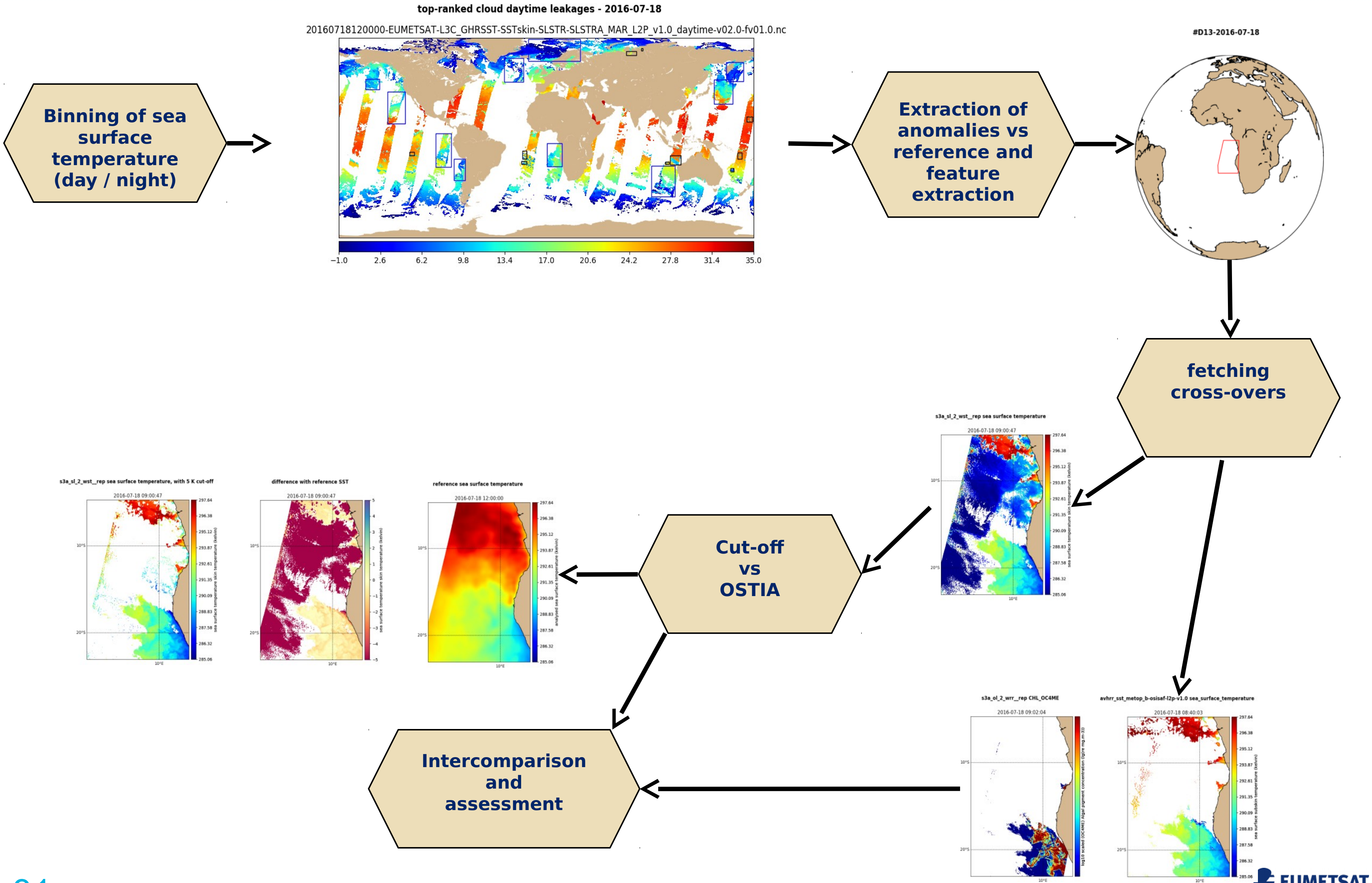




Dual-view SLSTR

Full swath SLSTR

Use case : cloud leakage detection workflow / cut-off adjustment



Catalogue of cloud cases

eumetsat-gses-1/~jfpiolle/s3calval/pages/cloud_summary.html

S3 Cal/Val | Home | Cal/val environment | Data quality | Sentinel-3 operations | Blog

Social

Twitter (Anne O Caroll)

Links

Metis

Airflow

The Sentinel-3A CloudLeaks - summary page

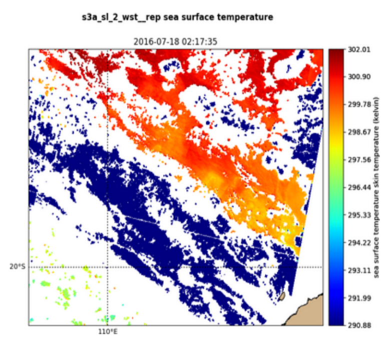
Previous | 07/18/2016 | Next

Cloud leakages [daytime] | Cloud leakages [night time] | Summary

Cases with SST deviation > 15 K

Case #D01-2016-07-18

Go to this cloud leakage case page



Pass start time: July 18th 2016, 02:17:35

Pass end time: July 18th 2016, 02:27:35

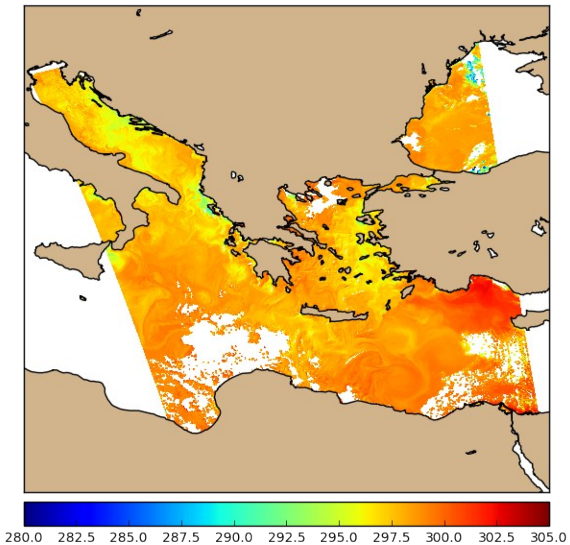
Granules:

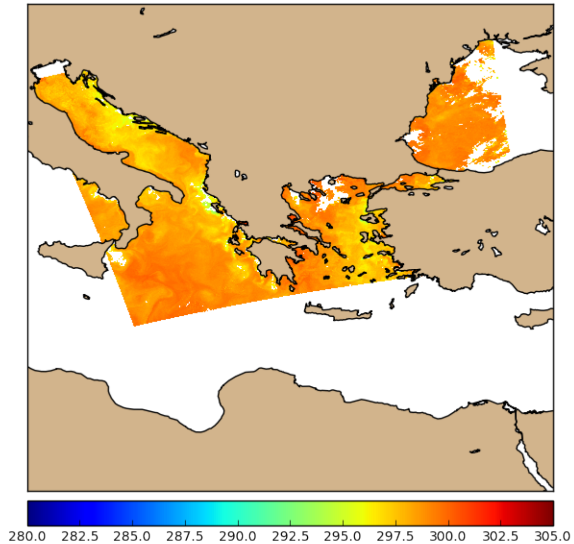
S3A_SL_2_WST____20160718T021736_20160718T022236_20170121T040728_0299_006_274______MR1_R_NT_002.SEN3

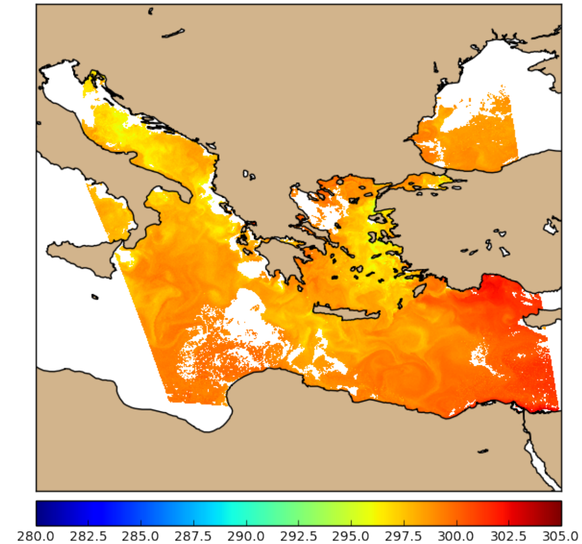
S3A_SL_2_WST____20160718T022236_20160718T022736_20170121T040706_0299_006_274______MR1_R_NT_002.SEN3

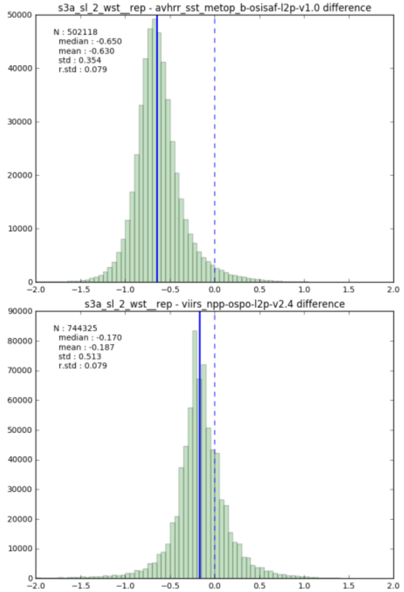
Case #D02-2016-07-18

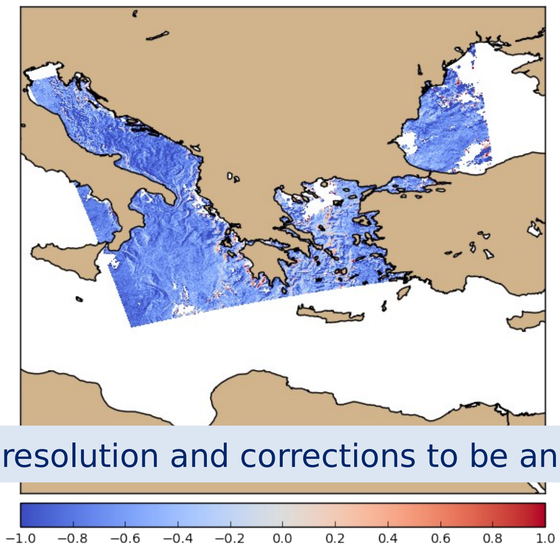
Use case : cross-over comparisons

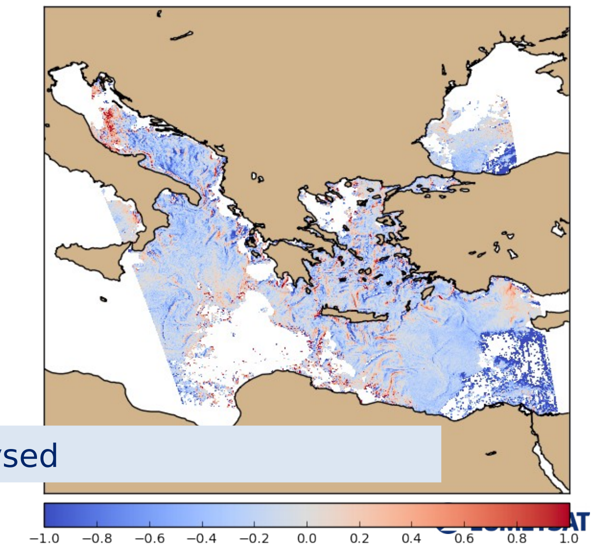












Feature resolution and corrections to be analysed

JUPYTER

EUMETSAT

jupyter

- http://jupyter.org/
- Python (but not only) in your web browser
- Embeds and mixes code, visualisation, explanations, equations in « notebooks »
- Growingly popular for interactive science
- Can run different languages (over 40)
- Can mix in some shell instructions
- Can be exported as html pages, pdf documents, .rst documents, LateX, python script
- Widgets for more interactivity, small task interfaces
- Complemented by **jupyterhub** which is single-user => allow multi-user access : a jupyter notebook server is spawned for each user

In S3 cal/val framework

- Quick visual development
- Sharing results with methodologie
- Learning tools and libraries
- Repeating analysis scenarii (new product release, longer time series,…)
- Advanced data analysis interfaces
- (Dashboards, report production)

Interactive integration of our different pieces of software

Interactive match-up outlier investigation with Jupyter

In [2]:

```
from s3analysis.slstr.mdb.analysis import get_basic_mask
from s3analysis.slstr.cloud import cloud_mask, DEFAULT_CLOUDMASK

# basic validity mask (sat angl < 55., wind speed > 6 m/s)
basic_mask = get_basic_mask(data_sat)

print "Number of match-ups : ", len(basic_mask)

print "Number of valid match-ups : ", (numpy.count_nonzero(basic_mask))
print "Number of invalid match-ups : ", (basic_mask).size - (numpy.count_nonzero(basic_mask))


# select only the match-ups where WST fields are defined
valid_sst = (
        basic_mask &
        ~numpy.ma.getmaskarray(data_sat['WST']["sst_theoretical_uncertainty"]) &
        (data_sat['WST']["quality_level"] > 2) &
        (numpy.ma.fabs(data_sat['WST']["dt_analysis"]) <= 5.) &
        ~numpy.ma.getmaskarray(data_insitu["water_temperature"])
        )


print "Final number of valid clear sky match-ups : ", numpy.count_nonzero(valid_sst)


slstr_sst = data_sat['WST']["sea_surface_temperature"] - 273.15
insitu_sst = data_insitu["water_temperature"]


cloudybox = cloud_mask(data_sat_box['WCT']["cloud_in"])
confidence = (data_sat_box['WCT']['confidence_in'][:] & 16384) > 0
```

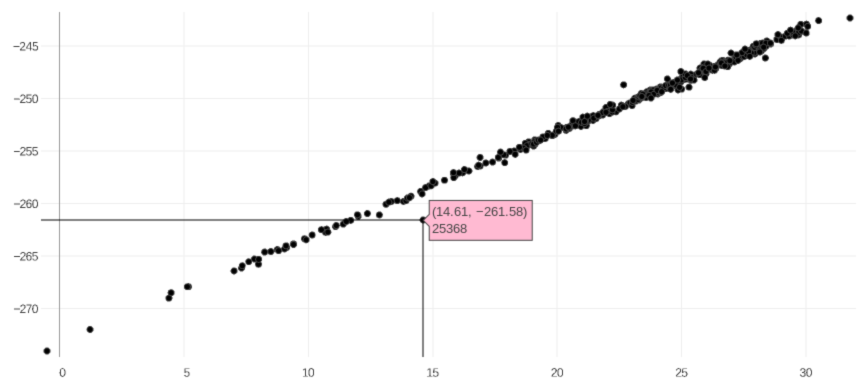
In [3]:

```
# additional filter to keep only nighttime data
night = (data_insitu['solar_zenith_angle'] > 90.)
```

In [4]:

```
# achtung! plotly neeeds to be installed in your environment (pip install plotly

import plotly.graph_objs as go
import numpy as np
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot

# allow inline plot with plotly
init_notebook_mode(connected=True)

# Create a interactive scatterplot SST vs in situ with plotly
trace = go.Scattergl(
    x = insitu_sst[night & valid_sst],
    y = slstr_sst[night & valid_sst] - 273.15,
    text = numpy.arange(len(slstr_sst))[night & valid_sst],
    mode = 'markers',
    marker = dict(
        color = 'FFBAD2',
        line = dict(width = 1)
    )
)
data = [trace]
iplot(data)

print len(insitu_sst[night & valid_sst]), ' match-ups'
```

```
print len(insitu_sst[night & valid_sst]), ' match-ups'
```



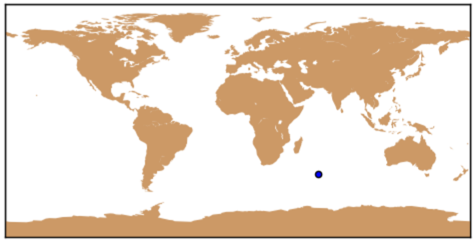
Export to plot.ly »

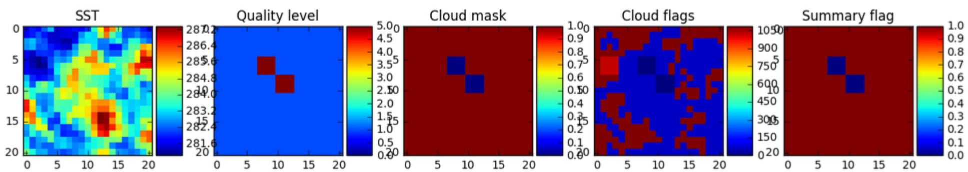
```
459  match-ups
```


EUMETSAT

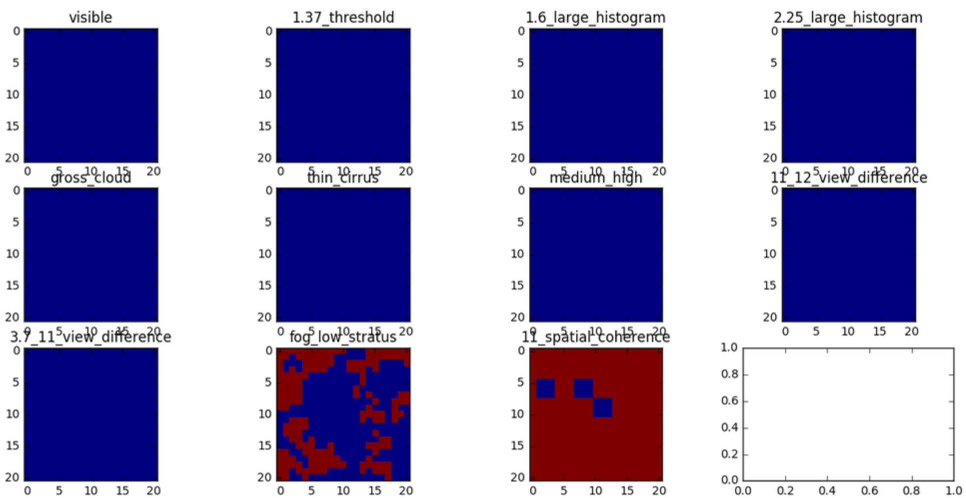
Interactive match-up outlier investigation with Jupyter

```
In situ value : 14.610000 K
SST - in situ difference : -3.040000 K
Traceability:
....WST file :  S3A SL 2 WST    20170628T185854 20170628T190154 20170628T202840 0179 019 198 5220 MAR F NR 002.SEN3
```






```
Used mask flags :  ['visible', '1.37_threshold', '1.6_large_histogram', '2.25_large_histogram', 'gross_cloud', 'thin_cirrus', 'medium_high', '11_12_view_difference', '3.7_11_view_difference', 'fog_low_stratus', '11_spatial_coherence']
```




```python
# display match-up info
print "SST value : %f K" % slstr_sst[choice]
print "In situ value : %f K" % insitu_sst[choice]
print "SST - in situ difference : %f K" % (slstr_sst[choice] - insitu_sst[choice])

print "Traceability:"
print "....WST file : ", data_sat['WST']['origin'][choice]

# locate match-up on map
from mpl_toolkits.basemap import Basemap
m = Basemap()
m.drawmapboundary()
m.fillcontinents(color='#cc9966')
x, y = m(data_insitu['lon'][choice], data_insitu['lat'][choice])
m.scatter(x, y)


# plot cloud and SST
plot_mask(choice)
```


EUMETSAT

Interactive match-up outlier investigation with Jupyter

trace back to original file

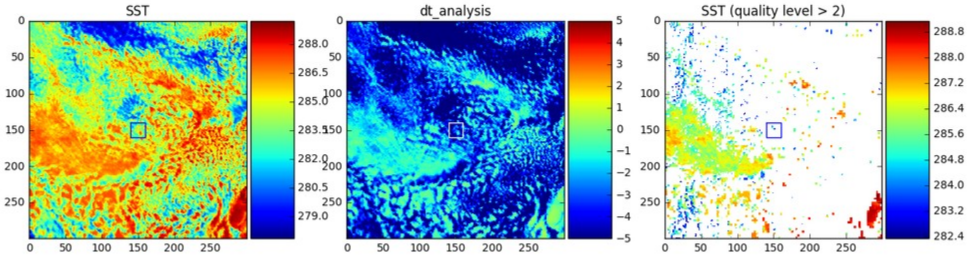
Here we access the content of the original file from which the match-up was extracted, and display a larger area around the match-up location.

This require to have access to the original SLSTR files!

In [6]:
```
print data_sat['WST']['dynamic_target_center_index'][choice]
```

```
[681 200]
```

In [7]:
```
# get full path name
from naiad.utils.filelocator import FileLocator
locator = FileLocator()
fname = locator.get_full_path(data_sat['WST']['origin'][choice], 's3a_sl_2_wst__ref')

# define large subset
row, cell = data_sat['WST']['dynamic_target_center_index'][choice]
boxwidth = 300
boxheight = 300
larger_box = {'row': slice(max(0, row - boxheight / 2), row + boxheight / 2),
              'cell': slice(max(0, cell - boxwidth / 2), cell + boxheight / 2)
             }

# load data into a cerbere swa=th object
from cerbere.mapper.safeslfile import SAFESLWSTFile
from cerbere.datamodel.swath import Swath
wstfile = SAFESLWSTFile(fname)
swath = Swath()
swath.load(wstfile)
```



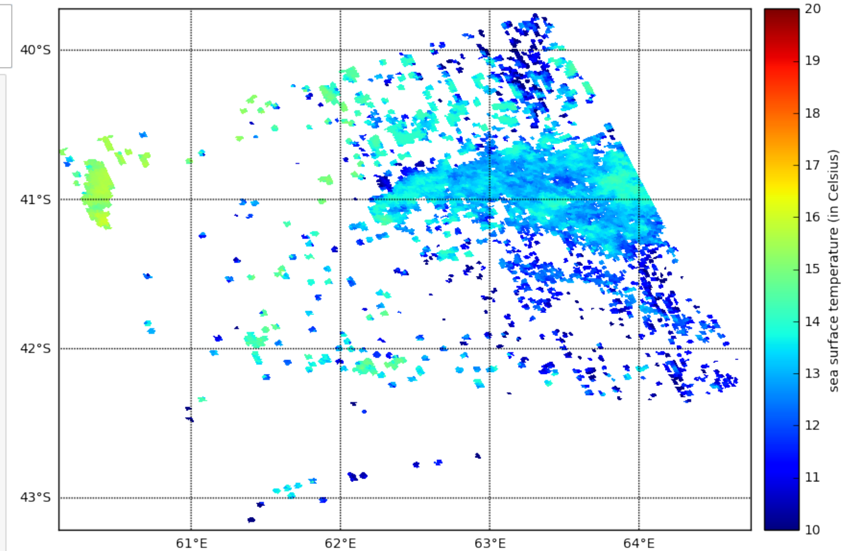
EUMETSAT

fetch Metop image

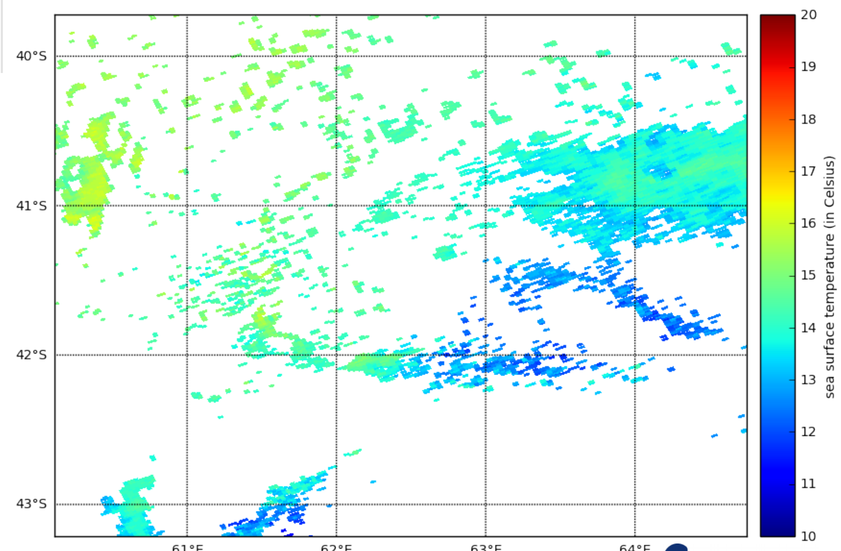
In [19]:

```python
# import necessary packages
import shapely

from naiad.utils.filelocator import FileLocator
from naiad.queries.server import Server
from naiad.queries.search import SpatioTemporalSearch

from cerbere.mapper.ghrsstncfile import GHRSSTNCFile
from cerbere.datamodel.swath import Swath
from cerplot.mapping import CerMap

%matplotlib inline

# provides Naiad server URL
es = Server("http://eumetsat-gses-5:9200/")

# =========== DEFINE HERE YOUR SEARCH CRITERIA ===========

# define the geographical search box
lats = swath.get_lat(slices=larger_box)
lons = swath.get_lon(slices=larger_box)
area = shapely.geometry.box(lons.min(), lats.min(), lons.max(), lats.max())

# define start and end of search interval
start = swath.get_start_time() - datetime.timedelta(hours=1)
end = swath.get_end_time() + datetime.timedelta(hours=1)

# define the naiad indice (product name) to crawl
product = 'avhrr_sst_metop_b-osisaf-l2p-v1.0'

# compose the query
search = SpatioTemporalSearch([product], area, start, end)

# execute the query
res = search.run(es)
```

SLSTR WST - 2017-06-28T18:58:53+00:00

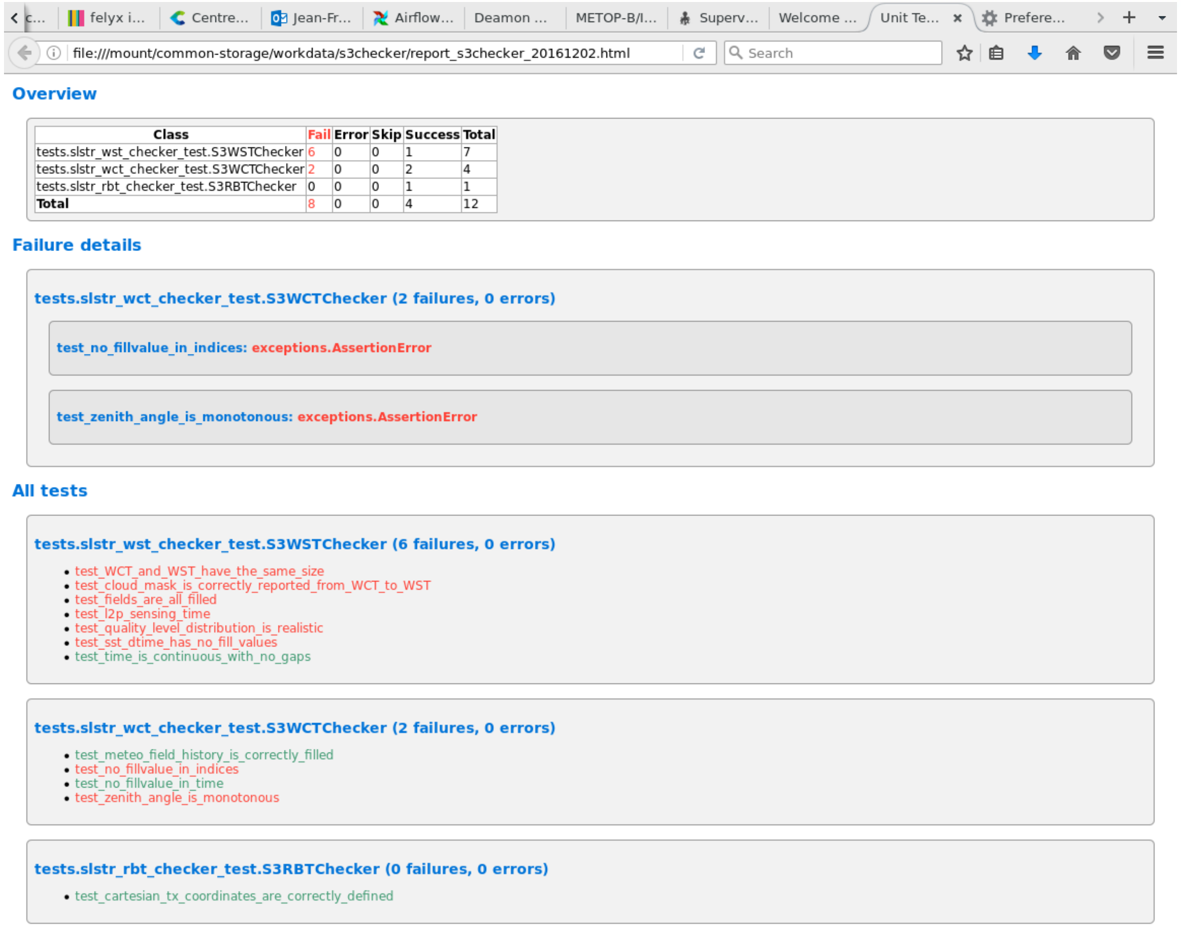


METOP AVHRR - 2017-06-28T18:46:03+00:00



MISC

Fix / regression verification : « s3checker »



Overview

Class	Fail	Error	Skip	Success	Total
tests.slstr_wst_checker_test.S3WSTChecker	6	0	0	1	7
tests.slstr_wct_checker_test.S3WCTChecker	2	0	0	2	4
tests.slstr_rbt_checker_test.S3RBTChecker	0	0	0	1	1
Total	8	0	0	4	12

Failure details

tests.slstr_wct_checker_test.S3WCTChecker (2 failures, 0 errors)

test_no_fillvalue_in_indices: exceptions.AssertionError

test_zenith_angle_is_monotonous: exceptions.AssertionError

All tests

tests.slstr_wst_checker_test.S3WSTChecker (6 failures, 0 errors)

- test_WCT_and_WST_have_the_same_size
- test_cloud_mask_is_correctly_reported_from_WCT_to_WST
- test_fields_are_all_filled
- test_l2p_sensing_time
- test_quality_level_distribution_is_realistic
- test_sst_dtime_has_no_fill_values
- test_time_is_continuous_with_no_gaps

tests.slstr_wct_checker_test.S3WCTChecker (2 failures, 0 errors)

- test_meteo_field_history_is_correctly_filled
- test_no_fillvalue_in_indices
- test_no_fillvalue_in_time
- test_zenith_angle_is_monotonous

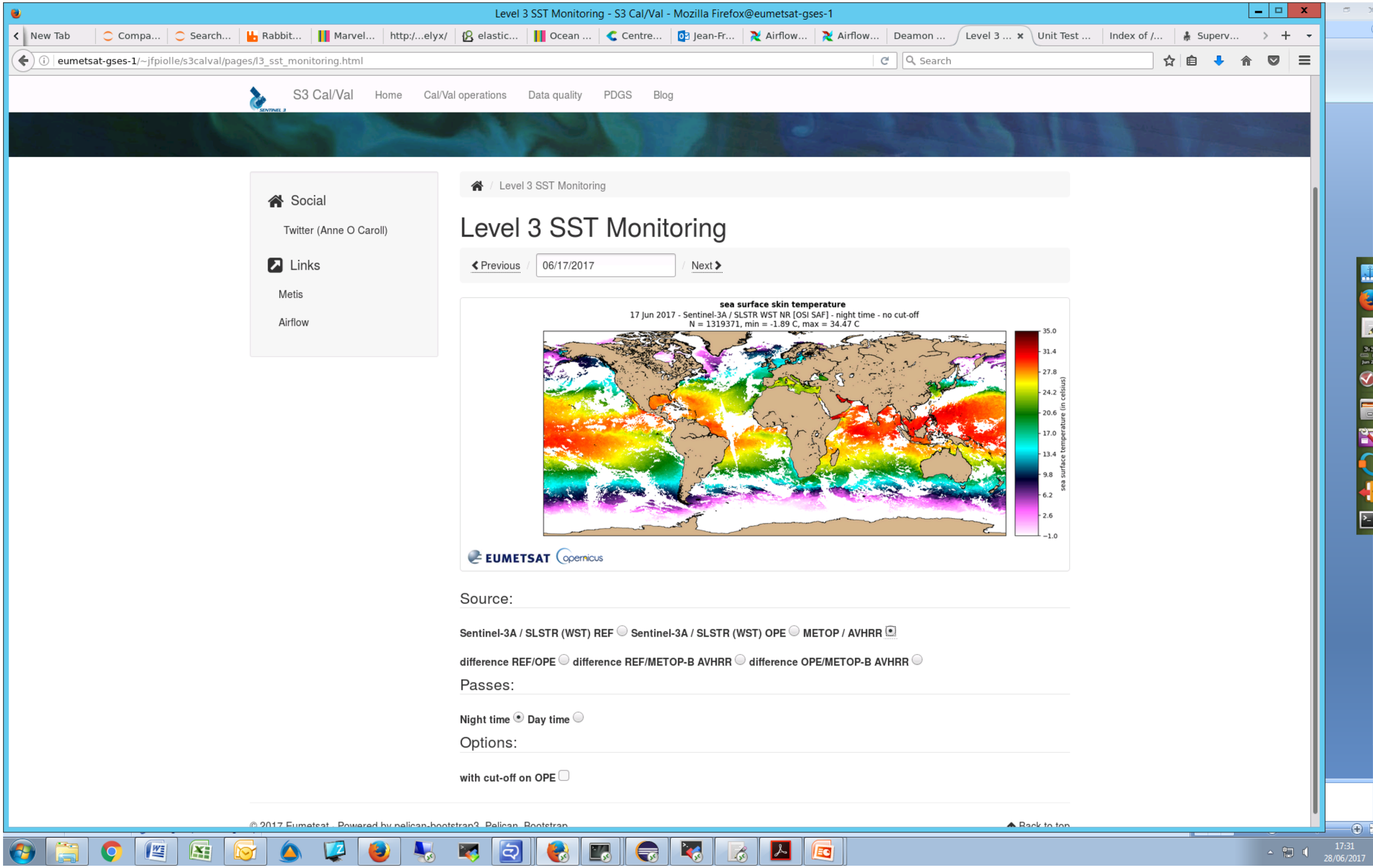
tests.slstr_rbt_checker_test.S3RBTChecker (0 failures, 0 errors)

- test_cartesian_tx_coordinates_are_correctly_defined

Based on python unitary test (**unittest**) framework – uses also scientific packages implemented for S3 data analysis

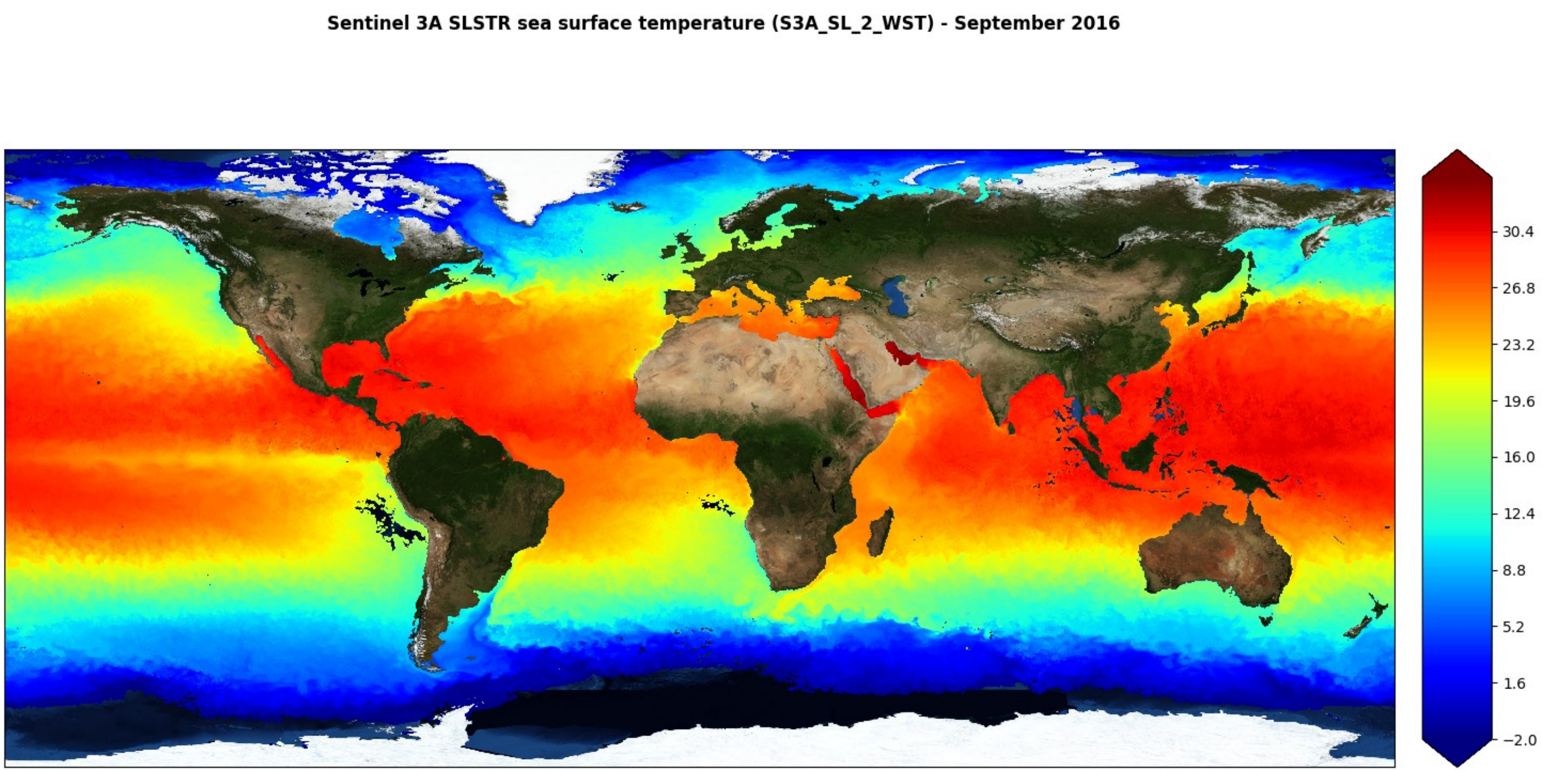
EUMETSAT

Configurable regridder/binner



EUMETSAT

Configurable regridder/binner



EUMETSAT Copernicus

Control tool and outreach

CONCLUSION

conclusion

- Demonstration of usage of open source tools for a collaborative framework, in the context of SLSTR cal/val
- Here focus on solution but was an asset in S3VT, esp. match-ups which were publically available
- Emphasis on existing generic tools, avoid reinventing the wheel
- Some tools have flaws but sharing among community would help improving them
- Embedded platform and tool concept can be applied to other aspects than cal/val, in support to GHRSST task teams, ex: CDAF implementation prototype
- Leverage on DAC dataset collections and resources for implementation of such services
- Extended version : https://www.eumetsat.int/website/home/VisitingScientists/SciencePresentations/DAT_3571206.html

EUMETSAT