Detecting changes in essential ecosystem and biodiversity properties- towards a Biosphere Atmosphere Change Index: BACI

Deliverable 2.6: Summary of exploitation of EO merging system, refinements, recommendations (scientific report/ submitted journal publication

Project title:	Detecting changes in essential ecosystem and biodiversity properties- towards a Biosphere Atmosphere Change Index
Project Acronym:	BACI
Grant Agreement number:	640176
Main pillar:	Industrial Leadership
Topic:	EO- 1- 2014: New ideas for Earth-relevant space applications
Start date of the project:	1st April 2015
Duration of the project:	48 months
Dissemination level:	Public
Responsible of the deliverable	Mathias Disney (UCL) Phone: +44(0) 20 7679 0592 Email: mathias.disney@ucl.ac.uk
Contributors	Mat Disney (UCL Geography) Maxim Chernetskiy (UCL MSSL, UCL Geography) Marcel Urban (FSU Jena) Christiane Schmullius (FSU Jena) Esther Conway (CEDA)
Date of submission:	26.03.2019

Summary
This deliverable D2.6 includes two summary documents describing the development and exploitation of the SSV for further applications. The first of these is a Jupyter Python notebook describing the contents of the SSV data files, how to open, read and extract information from the files, generate time series and gives simple examples of how to analyse the resulting data. Also included is a draft journal paper describing the development of the SSV dataset, in preparation for submission to Nature Scientific Data, open access.

Aim/Outcome
Interactive tool to allow users to navigate the contents of the BACI data, extract observations and generate downstream analysis; summary of the technical generation of the SSV dataset.

A merged optical, thermal and radar satellite dataset optimised for land surface change detection

Maxim Chernetskiy[1*], Mathias Disney[2, 3], Jose Gómez-Dans[2, 3], Marcel Urban[4], Philip Lewis[2, 3], Christiane Schmullius[4] and Esther Conway[5]

March 24, 2019

1. Department of Space and Climate Physics Mullard Space Science Laboratory, University College London, Holmbury St. Mary Dorking, Surrey, RH5 6NT, UK, m.chernetskiy@ucl.ac.uk 2. Department of Geography, University College London, Gower Street, London, WC1E 6BT, UK 3. National Centre for Earth Observation (NCEO), Department of Geography, University College London, Gower Street, London, WC1E 6BT, UK 4. Institute of Geography, Department for Earth Observation, Friedrich Schiller University, Grietgasse 6 07743, Jena, Germany 5. Science and Technology Facilities Council (STFC), Harwell, UK

Abstract

The aim of the generation of the dataset has been to produce a description of the surface state, a Surface State Vector (SSV), from a combination of satellite observations across wavelength domains i.e. albedo (visible), Land Surface Temperature (LST) (passive/thermal microwave) and backscatter (active microwave). The resulting dataset provides a unique spatially and temporally consistent (as far as the observations allow) series of observations of the land surface, across optical and microwave domains. The innovation of this approach is in providing a SSV in a common space/time framework, containing information from multiple, independent data streams, with associated uncertainty. The methods used can be used to combine data from multiple different satellite sources. The resulting dataset is intended to make the best use of all available observations to detect changes in the land surface state: the combination of data is likely to show changes that would not be apparent from data in a single wavelength region. The inclusion of uncertainty also allows the strength of the resulting changes to be properly quantified.

Background & Summary

This work was carried out as part of the "Detecting changes in essential ecosystem and biodiversity properties - towards a Biosphere Atmosphere Change Index" (BACI) project [1]. The main aim of the project is development of the "Essential Biodiversity Variables" which allow interpreting changes in biodiversity of ecosystems. The project employs novel machine learning methods for advanced change detection to cope with various sources of information including ground based measurements and remote sensing. The project requires continuous and unified datasets (i.e. consistent in time, space and wavelength domains) of remote sensing data over at least 15 years with spatial resolution 1 km. In this contribution we present description of creating combined datasets of remote sensing information from optical, thermal and Synthetic Aperture Radar (SAR) sensors. Data are generated over various BACI regional sites which are established due to their socio-ecological importance.

One of the complications in combining different Earth Observation (EO) data streams is a requirement of common time and space resolution. This can be archived by merging observations that were made by different EO sensors. Merging is typically involves some kind of spatio-temporal interpolation. However, applying interpolation can increase uncertainties due to different spectral properties of adjacent pixels. This in turn can dramatically influence further processing of data streams such as classification or change detection. This issue can be solved by propagation of uncertainties through whole processing chain. Therefore one of the requirement for this study is using a method which permits estimation of EO uncertainties. So we need to provide gap free time series of Earth Observation (EO) data with their associated uncertainties. In order to normalise reflectance, fill gaps due to cloudiness and calculate uncertainties we use temporal regularisation which is based on Bayesian inference. We exploit temporal regularisation which was presented in ([7], [6]). This technique allows filing gaps in the time series of parameters and explicitly characterise the output uncertainties.

Changes of Earth surface can have different nature and can influence different domains of electromagnetic spectrum. The user requirements are to make a combination of optical, thermal infrared and microwave data. Using different domains of electromagnetic spectrum as an input to an advanced machine learning methods can help in detection of comprehensive changes of earth surface. Currently best archive of global remote sensing data in optical and thermal infrared domains (it provides best balance of acquisition frequency, radiometric quality and spatial resolution) provided by the moderate-resolution imaging spectroradiometer (MODIS). Microwave domain is presented by Envisat/ASAR and Sentinel-1 VV/VH backscatter.

Due to user requirements we need to produce normalised gap free reflectance with one day step. Currently there two available products: GlobAlbedo and MODIS BRDF. The MODIS MCD43 products provide estimation of normalised reflectance and albedo ([2]). These products are creating by inversion of linear Kernel BRDF models. In order to constrain the problem the method uses 16

day time window. The MODIS method does not estimate uncertainties and does not feel gaps in time series. GlobAlbedo products use the same linear models and provide gap-free estimation and uncertainties ([4]). However GlobAlbedo uses simplified version of temporal constraint. Therefore we use regularised linear Kernel BRDF models for normalisation of surface reflectance to nadir view [7]. This method provides both estimation posterior uncertainties and filling gaps in time series. Reflectance at nadir is calculated by inversion and forward run of the Kernel models. In the case of thermal and SAR information we use identity operator i.e. smoother to fill gaps and estimate uncertainty. This allows minimum lost of information and makes data sets compatible.

The main aim of this paper is to demonstrate practical and consistent way of combining optical, thermal and SAR information as an input for advanced change detection methods. The presented approach provides optical, LST and microwave data streams in a common space/time framework with associated uncertainties. This is important for detection of change as some data streams may show changes that are not apparent in others. In the text below, we present a short discussion of different schemes of albedo retrieval and reflectance normalisation. Next, we describe details of the retrieval process. Then we show results of temporal normalisation and notice importance of posterior uncertainties in further analysis. Then we demonstrate a comparison between the results and the products of MODIS BRDF and GlobAlbedo. It should be stressed out that the goal of this work is not to develop physically based method of data merging but to demonstrate methodology of data preparation for complex multivariant change detection schemes.

Figure 1 shows a work flow of this study.

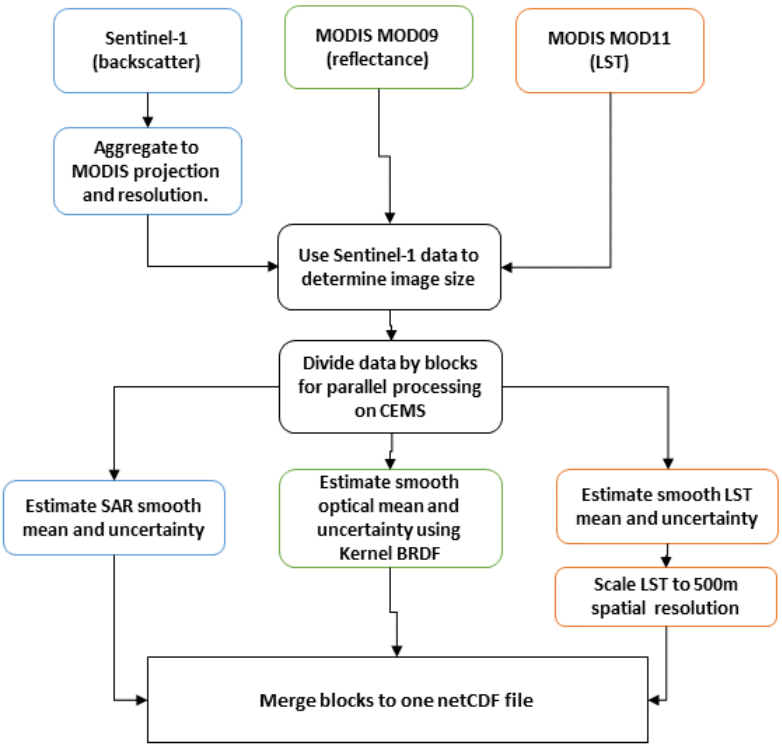


Figure 1: Work flow

Methods

In this work we process three sources of data: surface reflectance, SAR and LST. Due to their nature these data as well as most of the remote sensing information are noisy and have spatial/temporal gaps. A typical way to deal with such data is application one of the filtering procedures. This can reduce high frequency noise and fill gaps. In addition, essential information is uncertainties associated with filtered time series. We can see each available measurement y_{obs} as a random variable with an gaussian distribution determined by mean and variance. We have a state vector $\vec{x}$ which contains parameters which we want to estimate and an operator $H(\vec{x})$ which maps the state from $\vec{x}$ to the space of y_{obs}. Taking into account what we have time series of measurements we can stack variances into main diagonal of covariance matrix $C_{\vec{obs}}$. So we can write a mismatch function as

$$J_{obs} = \frac{1}{2}(y_{\vec{obs}} - H(\vec{x}))^T C_{\vec{obs}}^{-1}(y_{\vec{obs}} - H(\vec{x})) \quad (1)$$

where $y_{\vec{obs}}$ is a vector of measurements.

Since we do not use a model for microwave backscatter and LST i.e. state vector is in the same space as observations we can assume that observational operator is an identity operator $H(\vec{x}) = I(\vec{x})$:

$$J_{obs} = \frac{1}{2}(\vec{y_{obs}} - I(\vec{x}))^T \vec{C_{obs}}^{-1}(\vec{y_{obs}} - I(\vec{x})) \tag{2}$$

In order to provide smooth continuous solution we use a first order differential operator matrix which permits calculation between consecutive time steps.

$$\vec{\Delta} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -1 \end{pmatrix} \tag{3}$$

Now we can define a "zero-order" model which provides balance between measurements and expectation of smoothness:

$$J_{model} = \frac{1}{2}(\vec{\Delta}\vec{x})^T \vec{C_m}^{-1}(\vec{\Delta}\vec{x}) \tag{4}$$

where $\vec{C_m}$ is covariance matrix of the model.

However, usually estimation of $\vec{C_m}$ is not possible and we assume that a model error is a scalar constant value γ:

$$J_{model} = \frac{\gamma^2}{2}\vec{x}^T(\vec{\Delta^T}\vec{\Delta})\vec{x} \tag{5}$$

The difficulty here is estimation of the γ parameter. Usually it estimated by generalized cross validation.

In addition we can define a prior term:

$$J_{prior} = \frac{1}{2}(\vec{x} - \vec{x}_p^T)\vec{C_p}^{-1}(\vec{x} - \vec{x}_p) \tag{6}$$

where $\vec{C_p}$ is a prior covariance matrix. J_{prior} is our assumption about prior (background) value of a parameter. In case of a gap in time series solution tends to return to prior value x_p.

The final mismatch (cost) function is the "trade-off" of the three terms:

$$J = J_{obs} + J_{prior} + J_{model}, \tag{7}$$

The above expression is the core of the DA system in EO-LDAS but it is used in this work in its simplified form since we don't use radiative transfer modelling.

Bidirectional reflectance can be described as a sum of linear terms (kernels) which refer to different modes of scattering. Here we use expression given in [8] which is a sum of isometric, volumetric and geometric terms:

$$\rho(\theta, \vartheta, \phi, \lambda) = f_{iso}(\lambda) + f_{vol}(\lambda)K_{vol}(\theta, \vartheta, \phi) + f_{geo}(\lambda)K_{geo}(\theta, \vartheta, \phi) \tag{8}$$

where θ - solar zenith angle, ϑ - view zenith angle, ϕ - relative azimuth, f_{iso}, f_{vol} and f_{geo} are the kernel weights, K_{vol} and K_{geo} are kernels.

or in matrix form:

$$\vec{\rho} = \vec{K}\vec{f} \tag{9}$$

where $\vec{\rho}$ is a vector of reflectance values, $\vec{f}$ is a vector of kernel weights and $\vec{K}$ is a matrix of kernel values.

Kernels depend only on view and azimuth angels and can be easily calculated. So unknowns are the kernel weights which can be found by using the model (8) as $H(x)$ in equation (1) but it is possible to get the least square solution with imposing of an temporal constraint [7]:

$$\vec{f^*} = (\vec{K^{*T}C^{-1}}\vec{K^*} + \gamma^2 B^{\vec{T}}B)^{-1}K^{*T}\vec{C}^{-1}\rho \tag{10}$$

where $\vec{C}$ - covariance matrix and $\vec{B}$ - matrix which specifies temporal constraint.

Thus MODIS reflectance is normalised, smoothed and interpolated by applying regularised version of the Kernel BRDF models [6, 7]. This is a (semi) physically consistent and fast method which allows normalisation of reflectance in optical domain and explicit characterisation of the output uncertainties.

Uncertainties of SAR backscatter are standard deviation within 1x1km pixel. Uncertainties of thermal MODIS data are given in the product. MODIS per band uncertainties were taken from XXX

Uncertainties for MODIS reflectance are set to [0.004, 0.015, 0.003, 0.004, 0.013, 0.010, 0.006] for bands 1-7 respectively ([9]).

All processing have done by exploiting multiprocessing possibilities of the CEMS cluster by dividing each site by blocks, processing each block individually and then merging them to a whole image. Submitting calculation for each block to the "short-serial" queue of the high performance computing facility (LOTUS) allows processing up to 2000 jobs simultaneously.

Due to requirements of the project we have to provide continuous time series of Land Surface Temperature (LST). However the emissivity signal does support the smoothness assumption. Therefore we use difference between air temperature and LST because in this case it is possible to expect some smooth transition between days. Air temperature was obtained from the NCEP/NCAR reanalysis database [5]. It has daily frequency and spatial coverage 2.5° x 2.5° global grids.

Code availability

3.4Tb of data have been produced. Among them 284 GB – fast track sites, 3.1 T regional sites, where 1.8Tb are optical, 790 Gb thermal and 59 Gb of microwave data. More than 20000 lines of code were produced. The code is available on the Jasmin cluster in /group_workspaces/ jasmin2/ baci/ sigil/ baci_wp2_code/ The documentation is available on the basecamp and on Jasmin in /group_workspaces/ jasmin2/ baci/ sigil/ baci_wp2_doc/ In addition, a

tutorial was compiled, which gives some examples of how to operate the data using python.

Data Records

Study sites

Whole surface of Earth is a subject of constant change. However we can find some regions which can have particular interest due to faster climate change, political induced change, wars, etc. We can define such regions as hot-spots sites.

Due to different reasons of change we focus on six regional areas (Table 1). Spatial extent of the sites is determined by size of MODIS tiles (fig. 2).

Regional site	Main reason of changing
Northern Europe	Increasing commercial use of forest resources
South Africa	Fires and human induced forest degradation
Horn of Africa	Droughts and wars
West Africa	Extremes in climate variability
Ukraine	Droughts and political instability
Southern Europe	Droughts and fires

Table 1: Regional sites

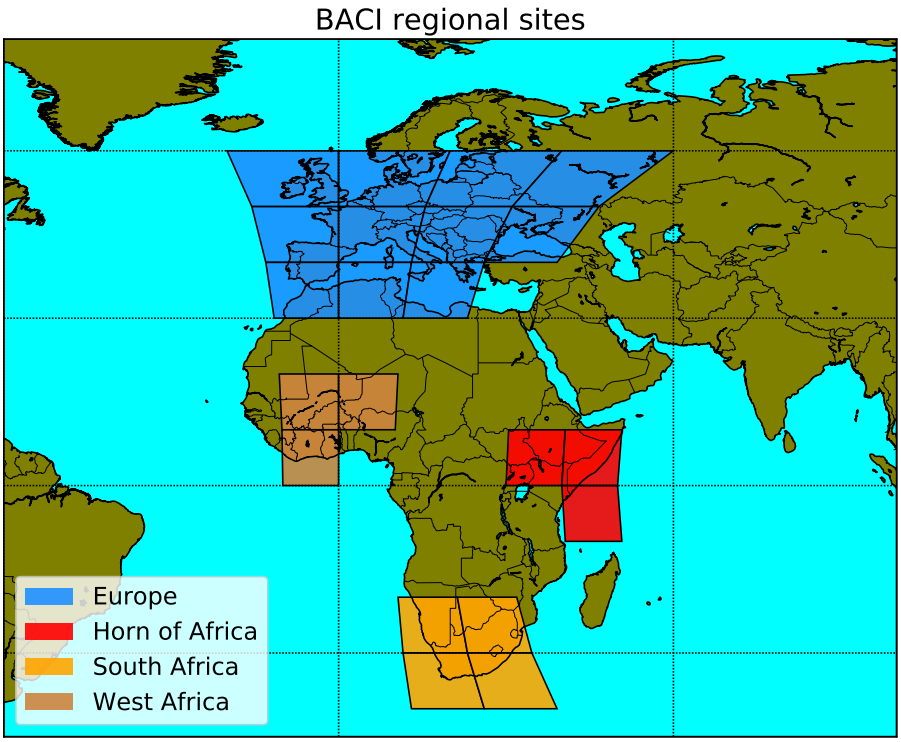


Figure 2: BACI regional sites on the world map

EO data

An important part of the project is exploiting resources of the facility for the Climate and Environmental Monitoring from Space (CEMS). We use CEMS computational power and archive of MODIS daily reflectance product (MOD09GA) from 2000 to 2015 ([3], ???reference to MODIS-CEMS archive???). MODIS daily averaged temperature MOD11A1 was downloaded from the NASA LP-DAAC ([10]). ESA ENVISAT/ASAR and Sentinel-1 HH/HV backscatter was downloaded from (???ref. to source of ASAR and S1 data???).

Sensors	years	
MODIS MOD09	2001-2016	Top Of Canopy (TOC) reflectance
MODIS MOD11	2001-2016	LST
ENVISAT ASAR	2002-2012	VV backscatter for ascending descending orbits
Sentinel-1	2014-2016	VV/HH backscatter for ascending descending orbits

Table 2: EO sensors and products

Technical Validation

State Surface Vector

The result of the processing is the Surface State Vector (SSV) which consists of Reflectance, LST and Backscatter. Figure 3 shows an example of SSV for one pixel. We can see what MODIS reflectance and LST go through all 15 years meanwhile ASAR backscatter stops in 2012. The Backscatter signal is continued by Sentinel-1 in 2014, so we have the gap in two years. We can notice that uncertainties of backscatter are relatively large. Backsactter is highly variable from pixel to pixel therefore when we aggregate it to lower resolution uncertainties are increasing.

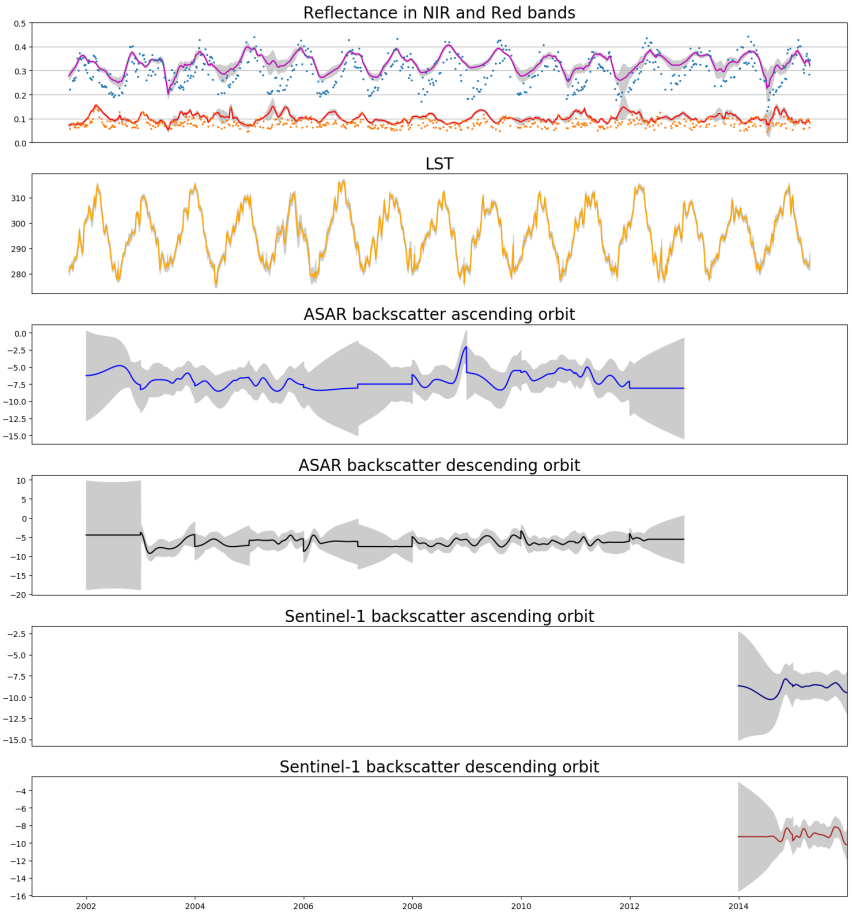


Figure 3: Time-series of reflectance, LST and backscatter

In the next sections each of the components of SSV is described in more details.

Optical data

Figure 4 shows retrieval of reflectance with one day, seven days and 30 days temporal steps. Note that increasing of temporal step means that we treat all measurements within that step as one observation and thus better constrain inversion. When we increase number of steps to seven we have more observations per step which provides better constraining and can decrease uncertainty. However, where are lost of some details. When step is increased to 30 days we have losing even more details and have increased uncertainties. The reason of increased uncertainty is that there are more variations in reflectance values within 30-days temporal window. The other thing to note is that at the beginning of year solar zenith angle has much higher values. Therefore normalised reflectance deviates from observations at the beginning of year and go through observations at the end and middle. So the data are retrieved with weekly time step because despite of reducing variability it maintains sensitivity to process. In addition it provides smaller file size and faster computational speed.

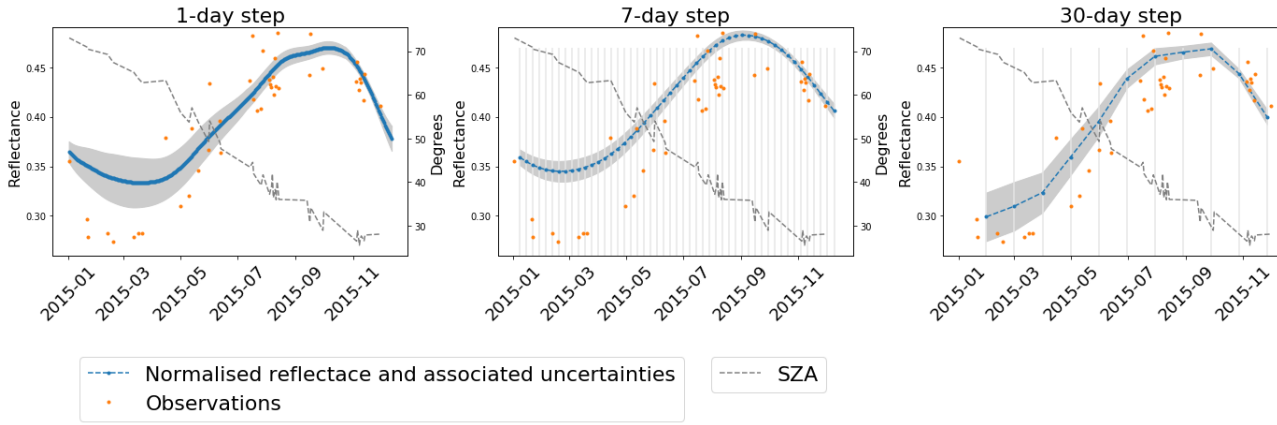


Figure 4: Different temporal steps in retrieval of normalised reflectance. Gray vertical lines represent bounds of temporal step

Figure 5 displays normalised reflectance for the BACI region sites (figure 2) with seven days time step. We see that there are some gaps especially over areas with frequent snow and clouds. This means that for these pixels we don't have enough observations to constrain the retrieved reflectance.

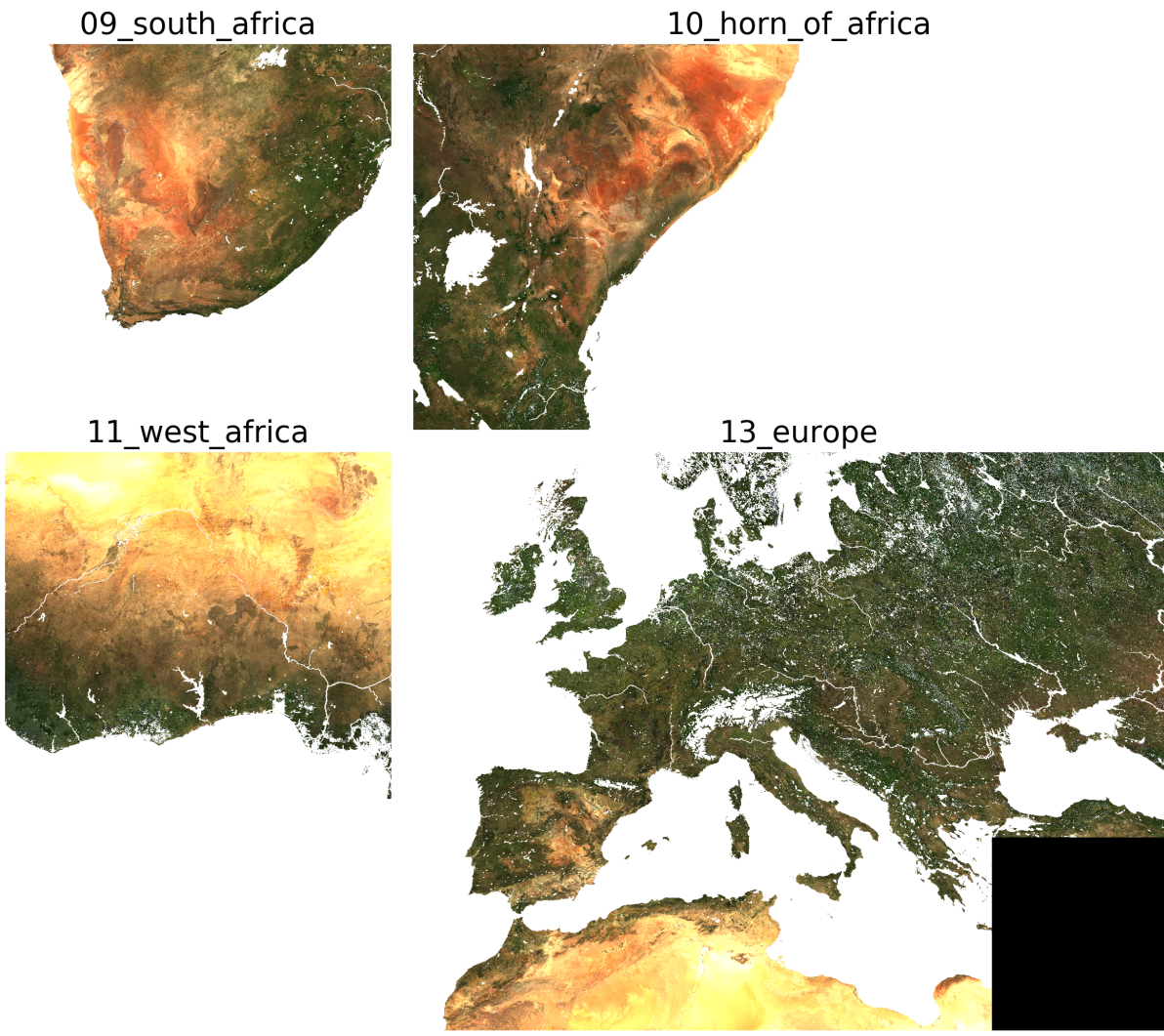


Figure 5: True colour composite of daily normalised surface reflectance over Europe. 7 days step.

Figure 6 shows difference between one day reflectance (MODD09) and normalised reflectance for the Southern Europe. We can see that most pixels of original MODIS image (left panel of fig 6) did not pass quality control due to atmospheric conditions. However, on the right panel most of the gaps are filled (fig 6). The way how the method works along time axis is shown on figure 7. The curve of normalised reflectance goes through actual observations of reflectance. I.e. information is restored with minimum lost. Here we should note that the fit

of normalised data to original data can not look optimal because original data are strongly influenced by the BRDF effect.

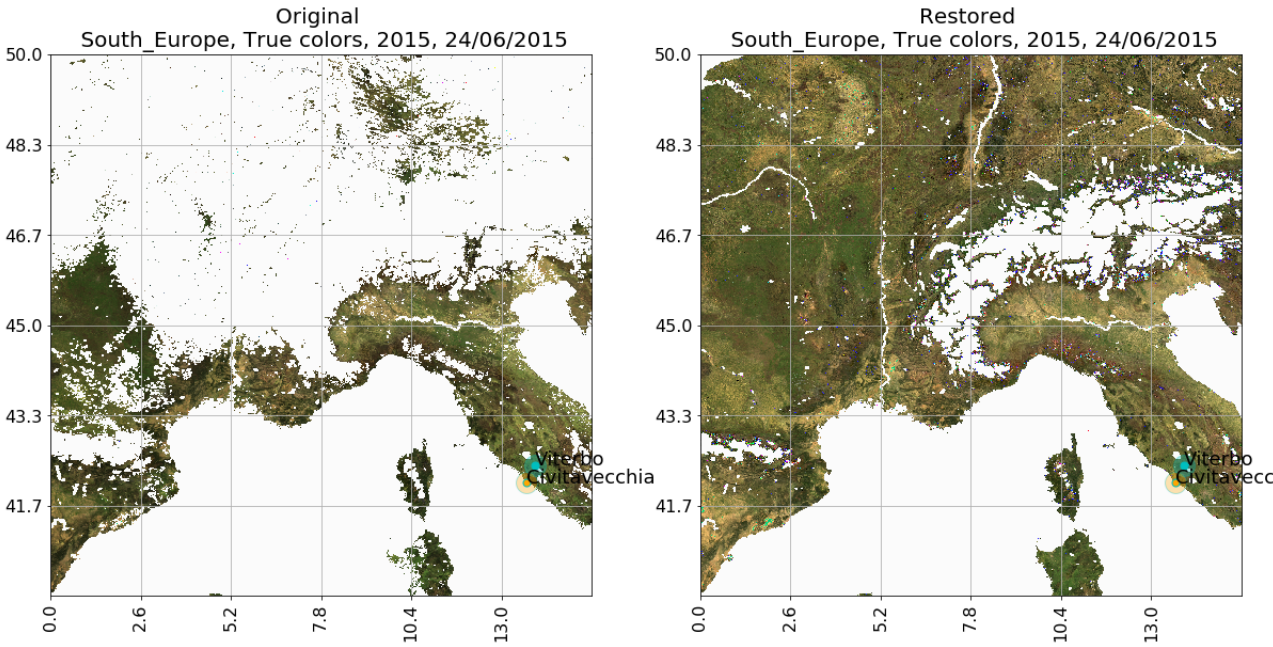


Figure 6: True colour composite of daily normalised surface reflectance over South Europe. Left panel: original MODIS data for year 2015 and Day of Year (DOY) 200. Right panel: restored normalized reflectance.

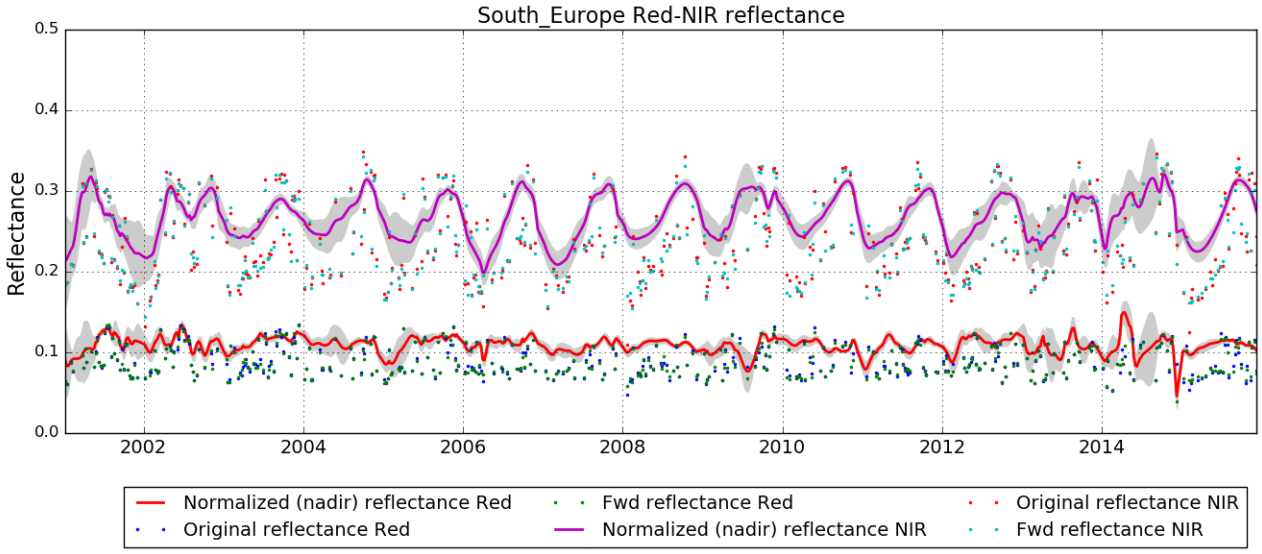


Figure 7: Time series of reflectance and associated uncertainties with seven days time step in MODIS bands one and two for years 2001 - 2015, single pixel over region of Viterbo (Italy). Dots depict actual observations, solid lines represent restored normalised reflectance.

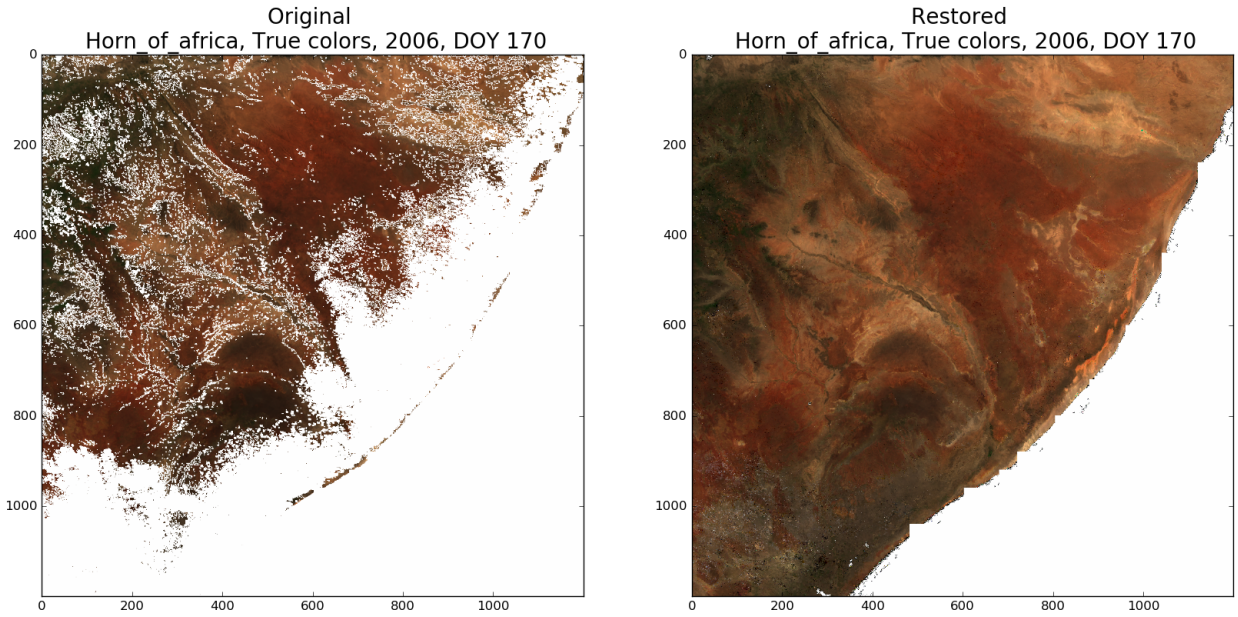


Figure 8: True colour composite of daily normalised surface reflectance over part of east Africa. Left panel: original MODIS data for year 2006 and Day of Year (DOY) 170. Right panel: restored normalised reflectance.

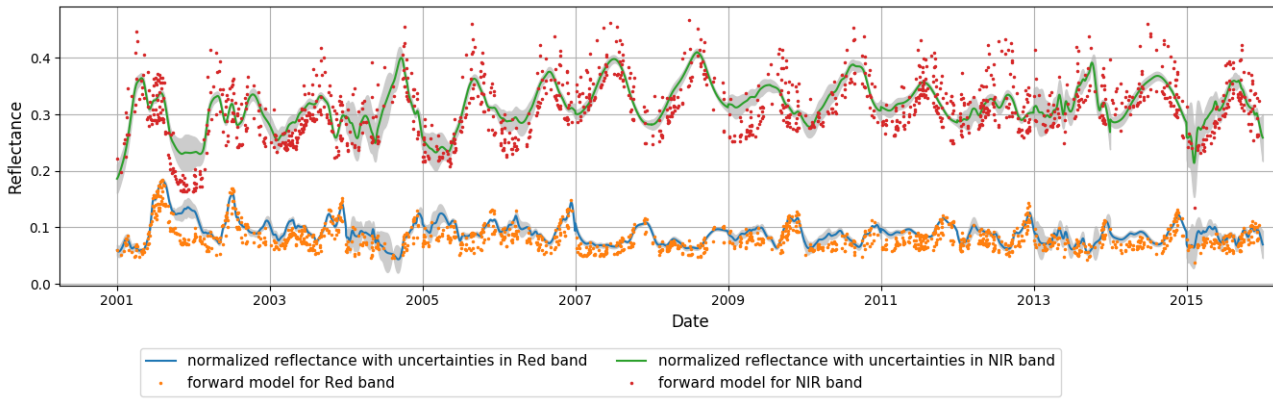


Figure 9: Time series of reflectance and associated uncertainties with seven days time step in MODIS bands one and two for years 2001 - 2015, east Africa. Dots depict actual observations, solid lines represent restored normalised reflectance.

Land surface temperature

Due to nature of LST we can not impose assumption of smoothness. However, we due to requirements to data we have to provide continuous time series. We can expect some degree of smoothness for difference between air temperature and LST. On the figure 11 we can see that despite of large difference in resolutions between air temperature and LST their difference does not exhibit borders of pixels unless this is a no-data region.

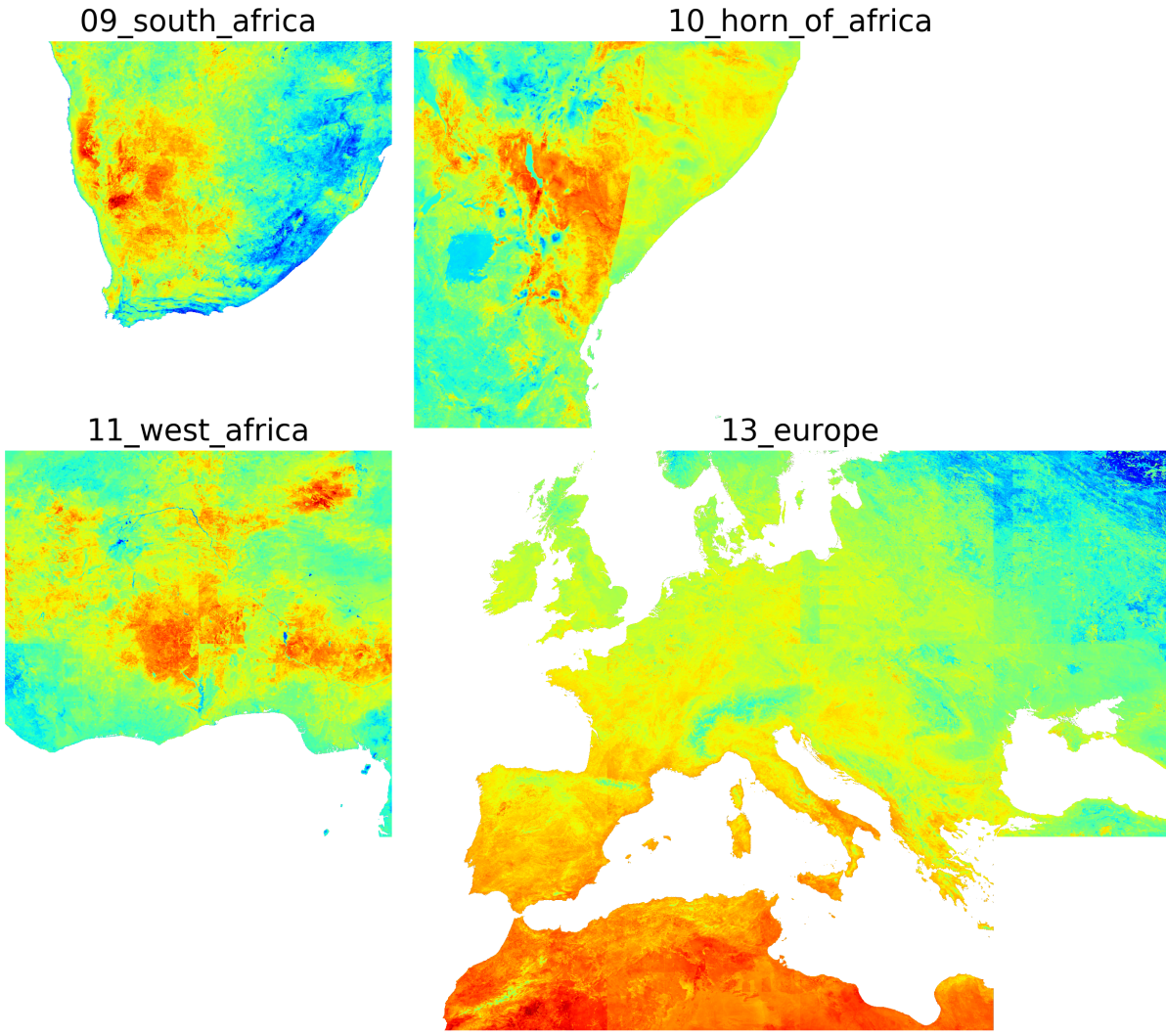


Figure 10: LST for the BACI region sites

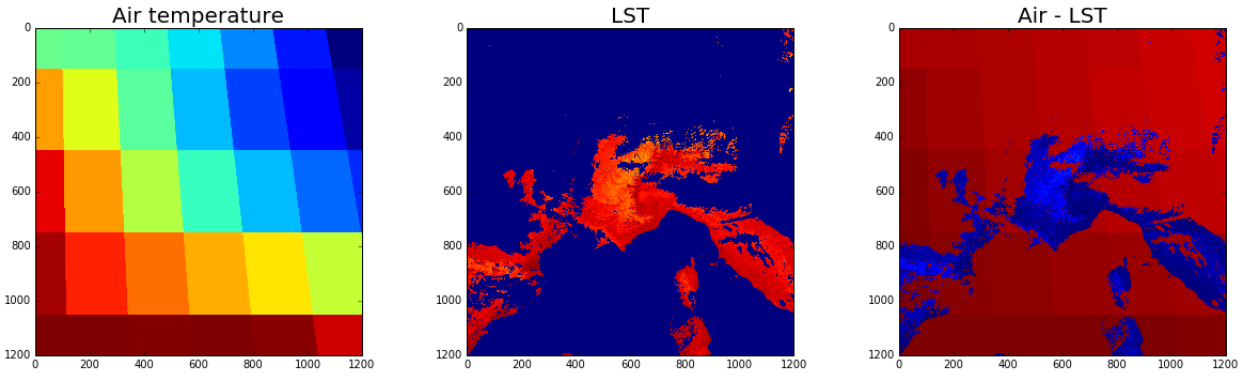


Figure 11: Difference between Land Surface Temperature and Air temperature

Figure 12 shows dynamics of Land Surface Temperature (LST) over the Viterbo fast track site for year 2014. We can notice relatively low uncertainties due to large amount of observations.

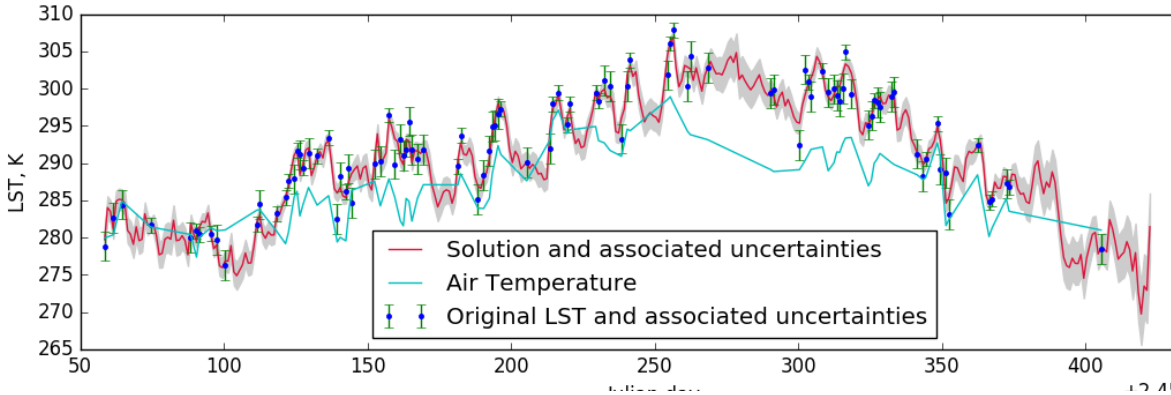


Figure 12: Smoothing LST by difference between Air Temperature and LST

Backscatter

Figures 14 and 15 show images of backscatter over the Viterbo site for year 2015 and DOY 365. We can see that Sentinel-1 observations are quite sparse comparing to MODIS and there are no observations after September. When we do not have observations the solution returns to the prior value and uncertainties are increasing. We can see that it is not possible to use this information in change detection or any other analysis without taking uncertainty into account.

Figure 13: Backscatter for the BACI region sites

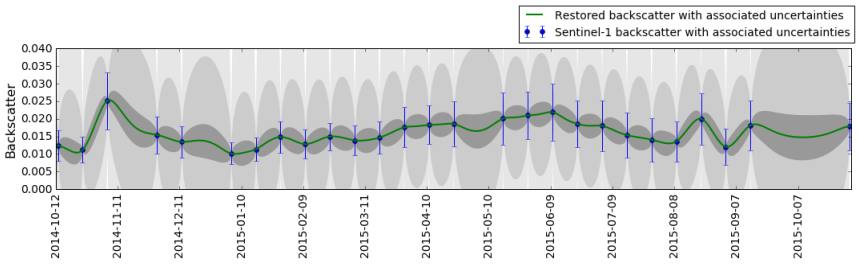


Figure 14: Sentinel-1 VH backscatter (linear) for the Viterbo test site, Italy (BACI FT), characterised as agricultural test site, for the period 10/2014 to 10/2015.

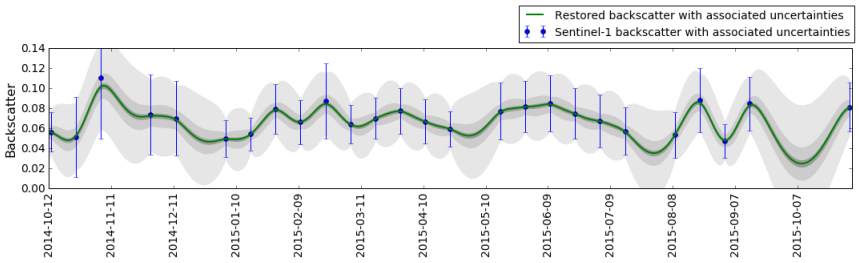


Figure 15: Sentinel-1 VV backscatter (linear) for the Viterbo test site, Italy (BACI FT), characterised as agricultural test site, for the period 10/2014 to 10/2015.

In order to characterise each site we calculated mean and standard deviation over whole areas. Figure 16 shows mean values for the test over Europe. We can see that reflectance and LST show relatively stable dynamic over all 15 years. The only exclusion is year 2002 in NIR which may be related to prolonged photosynthetic activity in this year. However, this feature does not reflected in dynamics of LST. Uncertainties of SAR data are quite high but this is expected because of very high range of values of the SAR signal. Dynamics of the ASAR ascending is noticeable because of very high values in 2002. However, this increase is in the range of uncertainties.

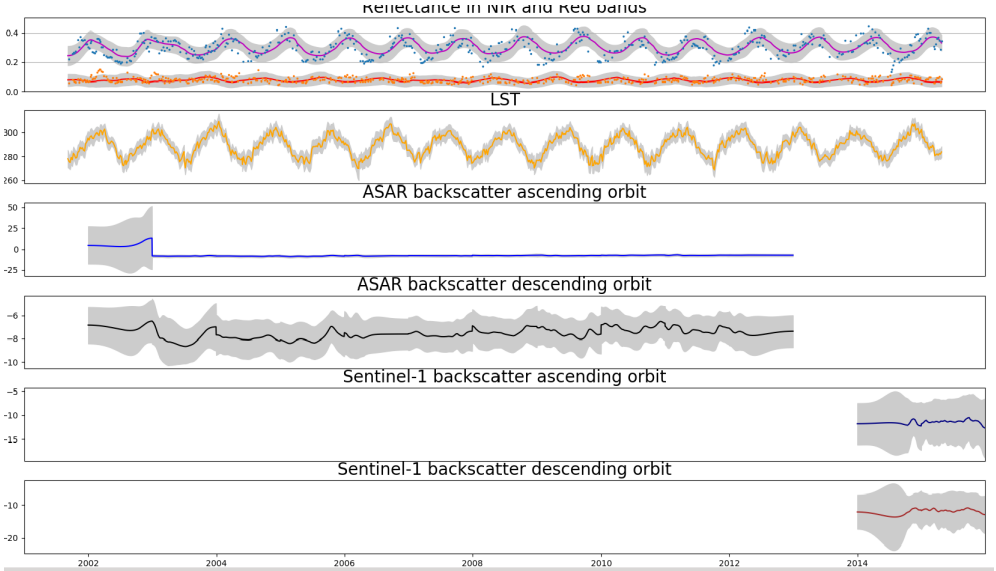


Figure 16: Mean and standard deviation over the European regional test site

Usage Notes

In order to give an example of usability of the data set we perform clustering over a small area. Firstly,

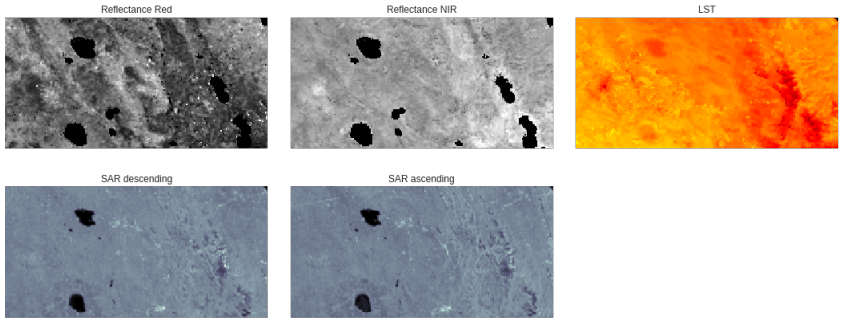
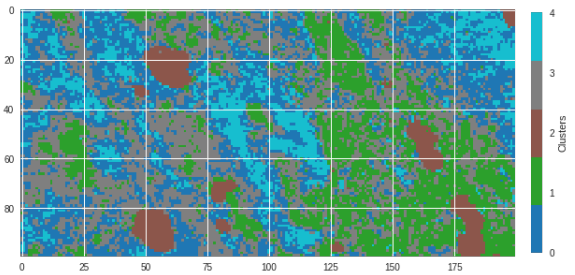
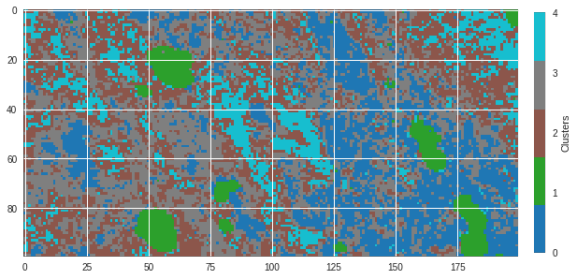


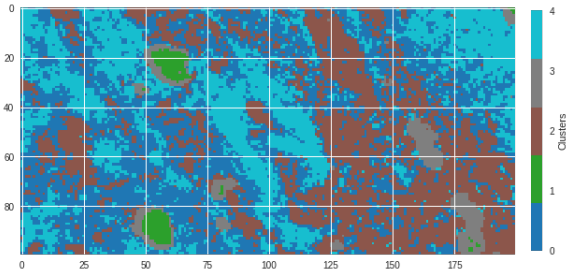
Figure 17: Clustering input



(a) Clustering red nir



(b) Clustering red nir lst



(c) Clustering red nir lst sar

Figure 18: Example of clustering

Conclusions

In this work, we have presented the SSV generation framework. The aim of the framework is to generate the SSV from all observations available in the MODIS albedo (visible), LST (passive/thermal microwave) and ENVISAT ASAR / Sentinel-1 (active microwave), to provide a unique spatially and temporally consistent (as far as the observations allow) series of observations of the land surface, across optical and microwave domains. The innovation of the presented work is in providing a SSV in a common space/time framework, containing information from multiple, independent data streams. This is vital for sensitivity to change as the microwave data may show changes that are not apparent in the optical, and vice versa. The SSV approach has been to provide data which are temporally regularised to avoid large day-to-day variations, and to account for this via the use of a specified uncertainty on the resulting observation trajectory. This uncertainty is provided as part of the SSV. However, this involves observations with very different space and time-varying characteristics, which presents challenges, described in this work. The conception of the SSV has been to account for variations in observation characteristics (time/space gaps) by increasing uncertainty, and not smoothing/interpolating observations.

Acknowledgements

We acknowledge financial support of the EU H20:20 BACI project (Project No. 640176).

Author contributions

Each author's contribution to the work should be described briefly, on a separate line, in the Author Contributions section.

Competing interests

A competing interests statement is required for all papers accepted by and published in *Scientific Data*. If there is no conflict of interest, a statement declaring this must still be included in the manuscript.

References

[1] *BACI: TOWARDS A BIOSPHERE ATMOSPHERE CHANGE INDEX. Available online:* `http://baci-h2020.eu/index.php/`. June last access at June 2017.

[2] Z. W. C. Schaaf. *MCD43A1 MODIS/Terra+Aqua BRDF/Albedo Model Parameters Daily L3 Global - 500m V006. NASA EOSDIS Land Processes DAAC.* https://doi.org/10.5067/modis/mcd43a1.006. 2015.

[3] R. W. E. Vermote. *MOD09GA MODIS/Terra Surface Reflectance Daily L2G Global 1kmand 500m SIN Grid V006. NASA EOSDIS Land Processes DAAC.* https://doi.org/10.5067/modis/mod09ga.006. 2015.

[4] *European Space Agency (ESA) GlobAlbedo Project. Available online: http://www.globalbedo.org (accessed on 8 January 2016).*

[5] E. Kalnay et al. "The NCEP/NCAR 40-year reanalysis project". In: *Bull. Amer. Meteor. Soc.* 77 (1996), pp. 437–470.

[6] P. Lewis et al. "An Earth Observation Land Data Assimilation System (EO-LDAS)". In: *Remote Sensing of Environment* 120 (May 2012), pp. 219–235. ISSN: 00344257. DOI: 10.1016/j.rse.2011.12.027. URL: http://linkinghub.elsevier.com/retrieve/pii/S0034425712000788.

[7] Tristan Quaife and Philip Lewis. "Temporal Constraints on Linear BRDF Model Parameters". In: *IEEE Transactions on Geoscience and Remote Sensing* 48.5 (May 2010), pp. 2445–2450. ISSN: 0196-2892. DOI: 10.1109/TGRS.2009.2038901. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5409680.

[8] Jean-Louis Roujean, Marc Leroy, and Pierre-Yves Deschamps. "A Bidirectional Reflectance Model of the Earth's Surface for the Correction of Remote Sensing Data". In: *Journal of Geophysical Research* 97.D18 (1992), pp. 20, 455–20, 468.

[9] E F Vermote and A Vermeulen. *Atmospheric correction algorithm: spectral reflectances (MOD09). Algorithm Technical Background Document.* 1999.

[10] S. H. Z. Wan. *MOD11A1 MODIS/Terra Land Surface Temperature/Emissivity Daily L3 Global 1km SIN Grid V006. NASA EOSDIS Land Processes DAAC.* https://doi.org/10.5067/modis/mod11a1.006. 2015.

Jupyter notebook allowing opening and exploitation of BACI SSV

BACI SSV demonstration

March 25, 2019

1 Analysing multiple sources of time-series data for change detection

Maxim Chernetskiy, Mathias Disney 2019 This exercise aims to explore a new, unique dataset generated as part of the Biosphere-Atmosphere Change Index (BACI) project (see http://baci-h2020.eu/index.php/). BACI is an ambitious 4 year EU-funded project (2015-2019) aiming to develop new ways to use EO and modelling to develop new indicators of change that might impact biodiversity. These can be obvious things such as fires, drought, deforestation etc. but also more subtle combinations of climate and anthropogenic impacts. This project relates to vaious efforts to monitor biodiversity. See for example the concept of so-called Essential Biodiversity Indicators (EBVs):

GEOBON: https://geobon.org/ebvs/what-are-ebvs/ Pereira et al. (2013) http://science.sciencemag.org/content/339/6117/277 Schmeller et al. (2017) https://link.springer.com/article/10.1007/s10531-017-1386-9 Kissling et al. (2018) https://www.nature.com/articles/s41559-018-0667-3

This is a good summary of some of the issues relating to EO by Pereira https://earthobservations.org/documents/meetings/201212_geobon_allhands/5%20EBVs%203Dec2012%20He

1.1 BACI Work package 2: Consistent EO datasets across space, time and wavelength

As part of the BACI project, here at UCL Geography we have been developing the underlying observation basis of the BACI index, what we call the System State Vector (SSV). This dataset consists of observations from different sensors across the EM spectrum, including optical (reflectance, albedo), thermal infrared (land surface temperature, or LST), and microwave (RADAR backscatter). We have developed a framework to allow datasets from different sensors such as NASA MODIS (optical, thermal) and RADAR data from ESA's ASAR (prior to 2014) and Sentinel-1 (2014 on) to be merged into a single stream of observations. The unique comnbintation of data in the SSV is more sensitive to changes in the surface state than any of the data sources individually. This exercise is aimed at allowing you to explore the properties of the BACI SSV in various ways. The SSV data are freely available over regions including Europe, Horn of Africa and S. Africa. There are likely to be opportunities for MSc RSEM students to apply the BACI data in dissertation projects.

1.2 Outline

This notebook provides worked examples of opening and manipulation of the BACI Surface State Vector (SSV) files in netCDF format. SSV consists of surface reflectance, albedo, land surface temperature (LST) and synthetic apperture radar (SAR) backscatter. All datasets have the same spatial

resolution, geographical projection and temporal step. Presented examples are based on the following python 3 libraries: netCDF4 - working with python netCDF; GDAL - Geospatial Data Abstraction Library; seaborn - enhanced data visualization and scikit-learn - machine learning library.

We have provided example Original datasets can be found in /home/ucfamc3/max_python/data/baci_wp2_files/. All regional sites which are in the geographical Europe are in the same folder /home/ucfamc3/max_python/data/baci_wp2_files/13_europe/

NOTE: you should **NOT** copy these files (they are large). Either use the links already setup from the directory where this notebook is located, or use the full path name for the files e.g.

Datasets are in sinusoidal projection and divided by MODIS tiles https://modis-land.gsfc.nasa.gov/MODLAND_grid.html where the h18v04 denotes horizontal tile 18, vertical tile 04. This means that processed regions are larger than BACI regional sites.

Table of Contents: 1. Opening and reading 1.1 Optical data 1.2 Land Surface Temperature (LST) 1.3 Synthetic Apperture Radar (SAR) backscatter 2. Reprojection 3. Principal Componenet Analysis (PCA) 4. Clustering 4.1 Red and NIR 4.2 Red, NIR and LST 4.3 Red, NIR, LST and micowave

```
In [ ]: import numpy as np
        import netCDF4 as nc
        import seaborn as sns
        from scipy import stats
        import pandas as pd
        import os
        import matplotlib.pyplot as plt
        %matplotlib inline
        import warnings
        warnings.filterwarnings('ignore')
        warnings.simplefilter('ignore')
```

```
In [ ]: sns.set()
```

1.3 1. Opening and reading

1.3.1 1.1 Optical data

First, open a netcdf file with the netCDF4 library function Dataset() which opens file returns a netcdf dataset. You have to specify your own path to the data!

```
In [ ]: ds = nc.Dataset('rho_h17v04_2015_7day.nc')
```

Now let's see what is inside the dataset

```
In [ ]: print(ds)
```

We can see some variables and two groups: reflectance and albedo. The meaning of groups is combining some variables together in order to make netcdf more structured. Now let's examine the variables. Dates are represented as julian dates which are days since January 1, 4713 BC. If we print the variable:

```
In [ ]: print(ds['julday'][:])
```

So we see that almost two and half million days are passed since starting date This form of representation is usefull but not super convinient for humans. So we can convert julian days to "normal" date format

```
In [ ]: import matplotlib.dates as dates
        date_arr = []
        for jd in ds['julday'][:]:
            date_arr.append(dates.num2date(dates.julian2num(jd)).date())
        date_arr = np.array(date_arr)
        print([d.isoformat() for d in date_arr])
```

The next couple of variables are latitude and longitude

```
In [ ]: print(ds['lat'])
        print(ds['lon'])
```

We can see that geographical coordinates are saved as two dimensional arrays with 1200X1200 pixels in each. If it's 2D array we can show an image of it

```
In [ ]: plt.subplot(121)
        plt.title('Lat')
        # This the main function in showing an image. Imshow - image show. Here it simply shows
        plt.imshow(ds['lat'], cmap = plt.set_cmap('Blues'))
        plt.colorbar(fraction=0.046, pad=0.03)

        plt.subplot(122)
        plt.title('Lon')
        plt.imshow(ds['lon'])
        plt.colorbar(fraction=0.046, pad=0.03)

        plt.tight_layout()
```

The last variable with out group is CRS which is a representation of geographical projection.

```
In [ ]: print(ds['crs'])
```

The groups 'reflectance' and 'albedo' are the combination of variables which relate to reflectance and albedo. This is the main thing in the datasets.

```
In [ ]: print(ds['reflectance'])
        print(ds['albedo'])
```

We see that varialbes in these groups have three dimentions i.e. date, x and y. The dimentionality of the variables:

```
In [ ]: print(ds['reflectance/refl_b2'])
```

In this case the temporal step is seven days + we can have some missing data. So first dimention is equal to 49. If we look at variable 'julday':

```
In [ ]: print(ds['julday'])
```

Plot NIR band
Use interactive mode (i.e. uncomment the first line %matplotlib notebook) to find coordinates

```
In [ ]: %matplotlib notebook
        plt.imshow(ds['reflectance/refl_b2'][0,:,:])
        plt.show()
```

Switch from interactive mode to normal show a point which was just found

```
In [ ]: %matplotlib inline
        # ****************
        x = 1033; y = 893
        # ****************
        plt.imshow(ds['reflectance/refl_b2'][0,:,:])
        plt.plot(x, y, marker='o', color='r')
```

Show temporal prifile of the point. Note that in python arrays coordinate 'y' is the first dimension and 'x' the seond!

So the same size as the first dimention of reflectance. Therefore we can use it to plot temporal evelution of the data along with associated uncertainties:

```
In [ ]: rho = ds['reflectance/refl_b2'][:, y, x]
        rho_sd = ds['reflectance/refl_b2_sd'][:, y, x]

        plt.fill_between(ds['julday'], rho + rho_sd, rho - rho_sd, color='0.8')
        plt.plot(ds['julday'], rho)
```

Let's show seven reflectance bands for the first date

```
In [ ]: def plot_refl(var, vmin=0, vmax=1):
            plt.figure(figsize=(15, 5))
            for b in range(7):
                plt.subplot(2, 4, b + 1)
                plt.title('band %d' % (b+1))
                plt.imshow(ds[var % (b+1)][0,:,:], vmin=vmin, vmax=vmax, cmap=plt.set_cmap('gray
                plt.colorbar(fraction=0.046, pad=0.03)
                plt.axis('off')
            plt.tight_layout()

In [ ]: plot_refl('reflectance/refl_b%d', vmin=0, vmax=0.4)
```

In the same way we can show uncertainties in each band

```
In [ ]: plot_refl('reflectance/refl_b%d_sd', vmin=0, vmax=0.05)
```

The group in the netcdf reflectance file is albedo.

```
In [ ]: def plot_albedo(var, vmin=0, vmax=1):
            plt.figure(figsize=(15, 5))
            bands = ['vis', 'nir', 'swir']
            for b in range(3):
                plt.subplot(2, 4, b + 1)
                plt.title('%s' % bands[b])
                plt.imshow(ds[var % bands[b]][0,:,:], vmin=vmin, vmax=vmax, cmap=plt.set_cmap('g
                plt.colorbar(fraction=0.046, pad=0.03)
                plt.axis('off')
            plt.tight_layout()

In [ ]: plot_albedo('albedo/albedo_%s', vmin=0, vmax=0.4)
        plot_albedo('albedo/albedo_%s_sd', vmin=0, vmax=0.05)
```

We can show a RGB image as three dimentional array where each dimention is a color componenet. For example true color composite.

```
In [ ]: img_rgb = np.zeros((1200, 1200, 3))
        img_rgb[:, :, 0] = ds['reflectance/refl_b1'][0,:,:] * 5
        img_rgb[:, :, 1] = ds['reflectance/refl_b4'][0,:,:] * 5
        img_rgb[:, :, 2] = ds['reflectance/refl_b3'][0,:,:] * 5
        plt.imshow(img_rgb)
        # plt.colorbar(fraction=0.046, pad=0.03)
        plt.axis('off')
```

1.3.2 1.2 Land Surface Temperature (LST)

Open and show the LST dataset.

```
In [ ]: # ds_lst = nc.Dataset('../data/baci_wp2_files/lst_h18v04_2015_7day.nc')
        ds_lst = nc.Dataset('lst_h17v04_2015_7day.nc')
        print(ds_lst)

In [ ]: lst = ds_lst['lst'][:, y, x]
        lst_sd = ds_lst['lst_sd'][:, y, x]

        plt.fill_between(ds_lst['time'], lst + lst_sd, lst - lst_sd, color='0.8')
        plt.plot(ds_lst['time'], lst)

In [ ]: plt.figure(figsize=(10, 5))

        plt.subplot(121)
        plt.title('LST')
        plt.imshow(ds_lst['lst'][0,:,:], cmap=plt.set_cmap('hot'))
        plt.colorbar(fraction=0.046, pad=0.03, label='Temperature, K')
        plt.axis('off')

        plt.subplot(122)
        plt.title('LST uncertainty')
```

```
plt.imshow(ds_lst['lst_sd'][0,:,:], vmin=0, vmax=5)
plt.colorbar(fraction=0.046, pad=0.03, label='Temperature, K')
plt.axis('off')

plt.tight_layout()
```

1.3.3 1.3 Synthetic Apperture Radar (SAR) backscatter

In [ ]:
```
ds_sar = nc.Dataset('sentinel-1_descending_h17v04_2015_7day_vv.nc')
print(ds_sar)
```

In [ ]:
```
# x = 1000
# y = 800

sar = ds_sar['bs'][:, y, x]

sar_sd = ds_sar['bs_sd'][:, y, x]

plt.fill_between(ds_sar['julday'], sar + sar_sd, sar - sar_sd, color='0.8')
plt.plot(ds_sar['julday'], sar)
```

In [ ]:
```
plt.figure(figsize=(10, 5))

plt.subplot(121)
plt.title('Backscatter')
plt.imshow(ds_sar['bs'][0,:,:], cmap=plt.set_cmap('bone'))
plt.colorbar(fraction=0.046, pad=0.03, label='Db')
plt.plot(x, y, 'o')
# plt.axis('off')

plt.subplot(122)
plt.title('Backscatter uncertainty')
plt.imshow(ds_sar['bs_sd'][0,:,:], vmin=0, vmax=5)
plt.colorbar(fraction=0.046, pad=0.03, label='Db')
plt.axis('off')

plt.tight_layout()
```

1.4 2. Reprojection

BACI netCDF files are in the MODIS sinusoidal projection (SR-ORG:6842). Below we can see an example with grid lines.

In [ ]:
```
from grid_lines import *

def draw_geogrid(img, lat_max, lat_min, lon_max, lon_min, proj, geo):
    """
    Draw grig lines over an image
    """
```

```
            if os.path.isfile('grid_lines.py'):
                # Draw lat-lon grid lines
                for lon in range(lon_min, lon_max, 2):
                    x1, y1 = get_pixels(geo, lat_max, lon)
                    x2, y2 = get_pixels(geo, lat_min, lon)
                    x1, x2, y1, y2 = find_inter(x1, x2, y1, y2, img.shape[0], img.shape[1])
                    plt.plot([x1, x2], [y1, y2], c='k')
                for lat in range(lat_min, lat_max, 2):
                    x1, y1 = get_pixels(geo, lat, lon_max)
                    x2, y2 = get_pixels(geo, lat, lon_min)
                    x1, x2, y1, y2 = find_inter(x1, x2, y1, y2, img.shape[0], img.shape[1])
                    plt.plot([x1, x2], [y1, y2], c='k')
```

```
In [ ]: lat_max = np.max(ds['lat'])
        lat_min = np.min(ds['lat'])
        lon_min = np.min(ds['lon'])
        lon_max = np.max(ds['lon'])

        # Get projection and geo-transformation
        crs_proj = ds['crs'].spatial_ref
        crs_geo = ds['crs'].GeoTransform

        # Get first date from the NIR reflectance band
        # img = ds['reflectance/refl_b2'][0,:,:]

        fig = plt.figure(figsize=(7, 7))
        plt.imshow(ds['reflectance/refl_b2'][0,:,:], vmin=0, vmax=0.6, cmap=plt.set_cmap('gray')

        draw_geogrid(ds['reflectance/refl_b2'][0,:,:],
                     np.round(lat_max).astype(int),
                     np.round(lat_min).astype(int),
                     np.round(lon_max).astype(int),
                     np.round(lon_min).astype(int),
                     crs_proj, crs_geo)
```

So we see that in sinusoidal projection geometry can't be described by 1D latitude and longitude vectors. Besides not all GIS software can recognize sinusoidal projection. An example is OpenLayers. In addition not all software can work well (or can't work at all) with netCDF format. Therefore sometimes it's nesssary to reproject data to WGS 84 (lat-lon, EPSG:4326) and save result in some well known format such as GeoTiff. For reprojection and geotiff we need library GDAL - Geospatial Data Abstraction Library

```
In [ ]: # import gdal
        import gdal, osr


        # Create empty output file
        drv_out = gdal.GetDriverByName('GTiff')
```

```
ds_out = drv_out.Create('baci_wgs84.tif', 1200, 1250, 1, gdal.GDT_Float32)

# make output projection
wgs84 = osr.SpatialReference()
wgs84.ImportFromEPSG(4326)
proj_out = wgs84.ExportToWkt()

# make geotransformation
lon_size = (lon_max - lon_min) / 1200.
lat_size = (lat_max - lat_min) / 1200.
# 0.01296 0.008
geo_out = (np.float(ds['lon'][0,0]), lon_size, 0, np.float(ds['lat'][0,0]), 0, -lat_size

# Setup input geotrasnforamation and projection
ds_out.SetGeoTransform(geo_out)
ds_out.SetProjection(proj_out)

# Create input gdal dataset in memory
drv_in = gdal.GetDriverByName('MEM')
ds_in = drv_in.Create('', 1200, 1200, 1, gdal.GDT_Float32)

# Setup input geotrasnforamation and projection
ds_in.SetGeoTransform(crs_geo)
ds_in.SetProjection(str(crs_proj))

# Write an image from netCDF to input gdal dataset
ds_in.GetRasterBand(1).WriteArray(ds['reflectance/refl_b2'][0,:,:])

# Do reprojection
res = gdal.ReprojectImage(ds_in, ds_out, str(crs_proj), proj_out, gdal.GRA_Average)
ds_in = None
ds_out = None
```

In [ ]:
```
# Open and show rreprojected geotiff

fig = plt.figure(figsize=(7, 7))
ds_out = gdal.Open('baci_wgs84.tif')
img_out = ds_out.GetRasterBand(1).ReadAsArray()
img_out = np.ma.array(img_out, mask = np.logical_or(img_out>1, img_out==0))
plt.imshow(img_out, vmin=0, vmax=0.6)

draw_geogrid(img_out,
             np.round(lat_max).astype(int),
             np.round(lat_min).astype(int),
             np.round(lon_max).astype(int),
             np.round(lon_min).astype(int),
             proj_out, geo_out)
```

Now we see an image transformed to WGS84. In this case we can use 1D arrays of latitude and longitude

1.5 3. Principal Component Analysis (PCA)

PCA is a way to decompose a set of observations (eg a time series in this case) to a sequence of underlying orthogoinal components (basis functions). This allows us to quantify where the variation in our signal comes from. A nice visual explanation of PCA is

http://setosa.io/ev/principal-component-analysis/

and a good intro paper

https://www.nature.com/articles/nmeth.4346

PCA is a way of simplifying high-dimensional data while retaining the main patterns and features - filtering the signal from the noise as it were. As a result PCA has been widely-applied in Big Data, machine learning to find patterns in complex, large and high-dimensional data sets.

This section shows a simple example of using PCA to reveal what is common in the three types of datasets

```
In [ ]: # ds_lst['time'][1:-3]
```

For PCA, clustering and classification we can use python machine learning library sklearn

```
In [ ]: import sklearn.decomposition as decomp
        import sklearn.preprocessing as prep
```

In this example we use only one pixel for one year. sklearn PCA uses as input a variable X which has shape (n_samples X n_features). In our example we have time series with 49 dates for three microwave domains: optical, temperature and microwave. So we have 49 samples and 3 features.

```
In [ ]: X = np.zeros((49, 3))
```

```
In [ ]: # create input variable X
        X = np.zeros((49, 3))

        # We have data with quite different scales so doing normalization is essential
        #X[:,0] = prep.normalize(rho)
        #X[:,1] = prep.normalize(lst[1:-3])
        #X[:,2] = prep.normalize(sar[1:-3])

        X[:,0] = np.reshape(prep.normalize(rho.reshape(-1,1),axis=0),49)
        X[:,1] = np.reshape(prep.normalize(lst[1:-3].reshape(-1,1),axis=0),49)
        X[:,2] = np.reshape(prep.normalize(sar[1:-3].reshape(-1,1),axis=0),49)

        # create an instance of object PCA
        pca = decomp.PCA()

        # fit PCA
        pca.fit(X)
```

```
# transform the data
comp = pca.transform(X)
print(comp.shape)

plt.figure(figsize=(15,3))
plt.subplot(141)

# plt.subplot(142)
# plt.plot(ds_lst['time'], lst)
# plt.subplot(143)
# plt.plot(ds['julday'], rho)
# plt.subplot(144)
# plt.plot(ds_lst['time'], sar)
tit = ['optical', 'LST', 'SAR']
for i in range(3):
    plt.subplot(1,4,i+2)
    plt.title(tit[i])
    plt.plot(X[:, i])


plt.figure(figsize=(15,3))
plt.subplot(141)
plt.title('Ratio of explained variance')
plt.plot(pca.explained_variance_ratio_, marker='o', ls='--')
for i in range(3):
    plt.subplot(1,4,i+2)
    plt.title('PCA %d' % (i+1))
    plt.plot(comp[:, i])
```

Define correlation function for mapping accross the axes

```
In [ ]: def corrfunc(x, y, **kws):
            """
            Correlation function

            Parameters
            ----------
            x: x-variable for regression
            y: y-variable
            """
            s, i, r, p, sd = stats.linregress(x, y)
            ax = plt.gca()
            ax.annotate("$r^2$ = {:.2f}\n y={:.2f}x+{:.2f}".format(r, s, i),
                        xy=(.05, .75), xycoords=ax.transAxes, fontsize=14)
```

Do PCA and plot correlation matrix (CM). This is a standard way for interpretation of PCs. We correlate each PC comp[:,k] with each input vector X[:,n]. Where k=0,..,number_of_PCs-1, n=0,..,number_of_input_parameters-1. The aim is to find relationships between reflectance, land

temperature and microwave backscatter. So, if a PC has higher correlattion with more than one input parameters we can guess some degree of relationship between these parameters.

On the previous figure (ratio of explaned variance) we can see that more than 90% variability explaned by the first three principal componenets. Therefore we can use only these three PCs.

```
In [ ]: d = {'rho': X[:,0], 'lst': X[:,1], 'sar': X[:, 2],\
             'PC1': comp[:,0], 'PC2': comp[:,1], 'PC3': comp[:,2]}
        # make pandas data frame which is special tabular representation of data
        df = pd.DataFrame(data=d)
        # use seaborn library to plot pairwise relationships in data
        g = sns.pairplot(df, kind='reg')
        # map correlation coeeficients. I.e. print r2 in corresponding plots
        g.map_lower(corrfunc)
        # sns.plt.show()
```

Examine how much of the observed variance is explained by the first three principal components. We can see that first PC is very well explained by SAR time series and second PC has high level of negative correlation with LST. Third PC partly explained by LST (r^2=-0.64). Here we don't see obvious relationship between parameters becouse all PCs are dominated by just one parameter. However, it doesn't mean that such relationship doesn't exist for other regions.

Sklearn also has other types of PCA: Incremental PCA, Kernel PCA, etc.

1.6 4. Clustering

Let's use the sklearn library for clustering of our data

```
In [ ]: from sklearn.cluster import KMeans

        # Let's open Sentinel-1 ascending orbit backscatter
        ds_sar_asc = nc.Dataset('sentinel-1_ascending_h17v04_2015_7day_vv.nc')
```

In order to increse speed we make a subset from the images

```
In [ ]: img_red = ds['reflectance/refl_b1'][0, 800:900, 1000:1200].data
        img_nir = ds['reflectance/refl_b2'][0, 800:900, 1000:1200].data
        img_lst = ds_lst['lst'][0, 800:900, 1000:1200].data
        img_sar = ds_sar['bs'][0, 800:900, 1000:1200]
        img_sar_asc = ds_sar_asc['bs'][0, 800:900, 1000:1200]

In [ ]: # Assign zero to no-data pixels
        img_red[img_red > 1] = 0
        img_nir[img_nir > 1] = 0
        img_lst[img_lst > 1000] = 0

        print(np.unique(img_sar))

        fig = plt.figure(figsize=(15, 6))

        ax=plt.subplot(231)
```

```
plt.title('Reflectance Red')
ax.imshow(img_red, vmin=0, vmax=0.2, cmap=plt.set_cmap('gray'))
plt.axis('off')

ax=plt.subplot(232)
plt.title('Reflectance NIR')
ax.imshow(img_nir, vmin=0, vmax=0.5, cmap=plt.set_cmap('gray'))
plt.axis('off')

ax=plt.subplot(233)
plt.title('LST')
ax.imshow(img_lst, vmin=260, vmax=300, cmap=plt.set_cmap('hot'))
plt.axis('off')

ax = plt.subplot(234)
plt.title('SAR descending')
ax.imshow(img_sar, vmin=-23, vmax=2, cmap=plt.set_cmap('bone'))
plt.axis('off')

ax = plt.subplot(235)
plt.title('SAR ascending')
ax.imshow(img_sar_asc, vmin=-23, vmax=2, cmap=plt.set_cmap('bone'))
plt.axis('off')

fig.tight_layout()
```

Here we use popular clustering method K-means

```
In [ ]: def do_clust(Ximg):
            """
            Do clustering and plot the results

            Parameters
            ----------
            Ximg: array
                an array of input values (n_samples X n_features)
            """

            # Initialize the KMeans object and fit the data
            kmeans = KMeans(n_clusters=5, random_state=0).fit(Ximg)
            # Predict clusters
            km = kmeans.predict(Ximg)
            # Reshape array of clusters to originl size of image
            img_c = np.reshape(km, (img_red.shape[0], img_red.shape[1]))
            # Show results
            im = plt.imshow(img_c, cmap=plt.get_cmap('tab10', 5))
            plt.colorbar(ticks=range(5), label='Clusters', fraction=0.023, pad=0.03)
```

1.6.1 4.1 Red and NIR

Firstly we are going to use only optical data from red and NIR bands

```
In [ ]: # Array of input values with shape n_samples x n_features
        Ximg = np.zeros((img_red.shape[0] * img_red.shape[1], 2))

        # We have data with quite different scales so doing normalization is essential
        Ximg[:, 0] = prep.normalize(img_red.reshape(-1,1),axis=0).flatten()
        Ximg[:, 1] = prep.normalize(img_nir.reshape(-1,1),axis=0).flatten()


        plt.figure(figsize=(10, 5))
        do_clust(Ximg)
```

1.6.2 4.2 Red, NIR and LST

On next step let's add temperature for the same date as a new feature

```
In [ ]: Ximg = np.zeros((img_red.shape[0] * img_red.shape[1], 3))

        # We have data with quite different scales so doing normalization is essential
        Ximg[:, 0] = prep.normalize(img_red.reshape(-1,1),axis=0).flatten()
        Ximg[:, 1] = prep.normalize(img_nir.reshape(-1,1),axis=0).flatten()
        Ximg[:, 2] = prep.normalize(img_lst.reshape(-1,1),axis=0).flatten()

        plt.figure(figsize=(10, 5))
        do_clust(Ximg)
```

In the results of clusterisation with data of reflectance and LST we can clearly see some lakes.

1.6.3 4.3 Red, NIR, LST and micowave

Add SAR data

```
In [ ]: Ximg = np.zeros((img_red.shape[0] * img_red.shape[1], 5))

        # We have data with quite different scales so doing normalization is essential
        Ximg[:, 0] = prep.normalize(img_red.reshape(-1,1),axis=0).flatten()
        Ximg[:, 1] = prep.normalize(img_nir.reshape(-1,1),axis=0).flatten()
        Ximg[:, 2] = prep.normalize(img_lst.reshape(-1,1),axis=0).flatten()
        Ximg[:, 3] = prep.normalize(img_sar.reshape(-1,1),axis=0).flatten()
        Ximg[:, 4] = prep.normalize(img_sar_asc.reshape(-1,1),axis=0).flatten()

        plt.figure(figsize=(10, 5))
        do_clust(Ximg)
```

The new information that have been added to results is separation of lakes with shallow waters (right hand side of the image) and more deep waters (left hand side). This kind of information can't be detected by optical data. At the same time we can see some patterns related vegetation which can't be clearly seen by a microwave sensor.

1.7 Outcomes, other regions, additional reading,

This exerecise should give you an insight into the contents of the BACI SSV data, but more generally, how you can open, manipulate and analyse time series data of this sort. Following this exercise you should:

Have some knowledge of the concept of EBVs

Understand the contents of the BACI SSV data

Be able to open, explore and plot netCDF files from different sources

Understand how to apply PCA to explore information content of time series data

1.7.1 Other regions, years, tiles

You will note from the data location that there are other BACI focus regions in East and Southern Africa, with other years and tiles. The Horn of Africa region has data from 2002 - 2015. Try the same analysis with a different site, and/or different years and see if you pick up different temporal and spatial patterns. Given which year and tile you chose, see if you can explain the differences.

1.7.2 Reading

Do read some of the background material on EBVs. Note that the concept of EBVs ias based on that the well-established concept of Essential Climate Variables (ECVs), which are now routinely produced. See the links below for example:

http://cci.esa.int/content/what-ecv

https://public.wmo.int/en/programmes/global-climate-observing-system/essential-climate-variables

https://www.ncdc.noaa.gov/gosic/gcos-essential-climate-variable-ecv-data-access-matrix

The BACI SSV was generated using the approach of the EO-LDAS (EO Land Data Assimlation System) system developed here at UCL. Have a look at https://earth.esa.int/web/guest/software-tools/-/article/eoldas and http://www.eoldas.info/ and the underlying paper on which EO-LDAS is based, Lewis et al. (2012): https://www.sciencedirect.com/science/article/abs/pii/S0034425712000788