# Copernicus

# Tutorial: Working with Sentinel-2 products

## RUS & SNAP

Copernicus
Europe's eyes on Earth

Space

Copernicus EU
Copernicus EU
Copernicus EU
www.copernicus.eu

# AGENDA

- 1. Introduction to RUS
- 2. Introduction to SNAP
- 3. Downloading and opening a Sentinel-2 image
- 4. Processing a Sentinel-2 product: vegetation monitoring
  - Preprocessing
  - NDVI and LAI
- 5. Automate processing using GPT


# COPERNICUS RESEARCH & USER SUPPORT (RUS)

## AN ESA/EU FUNDED SERVICE

# RUS Service - What for?

- **Foster the handling and processing of data from Copernicus missions by the Academic, scientific, R&D community, SMEs**
- **Mitigate the “digital divide” affecting Copernicus data access and exploitation**
- **Enhance and support the initiatives aiming at supporting Copernicus uptake**

# RUS Service - How?

- **Offering unique access to free data and ICT resources to scale up R&D and early prototyping activities over large amounts of Sentinel products**
- **Providing a specialized user helpdesk accompanying the service users in their activities with technical advice from a team of skilled experts**
- **Offering open hands-on training sessions in Europe**
  - **Customized technical training**
  - **Training programme dedicated to future “trainers”**

# RUS Service – For whom?

- **Research users** Scaling up of algorithms on large amounts of core products
- **Expert users** Process large amount of core products using Free and Open-Source Software (FOSS) or Commercial Off The Shelf (COTS) tools
- **University classes** Use Sentinel core products with either own algorithms, FOSS or COTS
- **Specific research/user communities** (e.g. EU Member States, Commission Services, Third Countries, SMEs, H2020 projects…) Request dedicated support to facilitate uptake of Sentinel core products

# RUS Service - Tasks

## Provide and manage resources for the service users

- Sentinel and Copernicus Contributing Missions data
  - open also to other data sources (e.g. Landsat, ALOS)
- Toolboxes and software needed for data processing and prototyping activities
  - possibility to install also own tools
- Virtualised and scalable computing resources (VM or VM clusters) provided by dedicated cloud providers

## Support ownership of Copernicus space component by end-users

- Capacity building
- Generic user training and training of Copernicus trainers

# RUS Service - Levels

## The RUS Service has been designed to accommodate as much as possible user needs and provide the appropriate support and ICT resources

- User requests are analysed in order to estimate this support and these resources
- User requests are answered following a 3-level classification
  - Basic support service (Level A), 1-3 month duration
  - Development support service (Level B), 1-6 month duration
  - Processing support service (Level C), 1-12 month duration

# RUS Service – Typical VM Content

## Processing environment

- Sentinel-1/2/3 Toolboxes
- SNAP4Cloud (built-on Apache Ignite)
- Support tools
  - Sen2Cor
  - Sen2Three
  - SMOS Toolbox
  - Orfeo Toolbox
  - BRAT Toolbox
  - Rugged library
  - OpenCV
  - GDAL library
  - NCO
  - NETCDF
  - OpenJPG
  - Image Magick
  - QGis

## Development environment

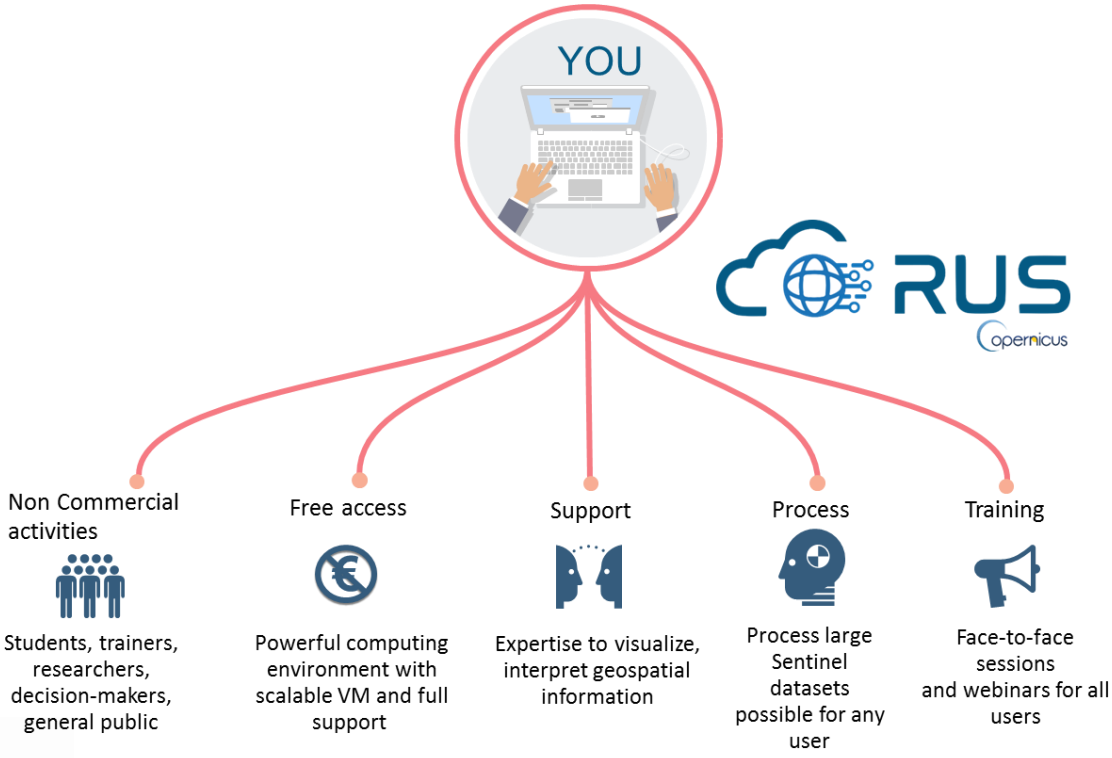
- Eclipse Mars
- GCC
- Cmake
- Maven
- Git
- R
- Eclipse Mars plugins
  - Pydev
  - CDT (C/C++ IDE)
  - StatEt
  - Cmaked
  - EGit

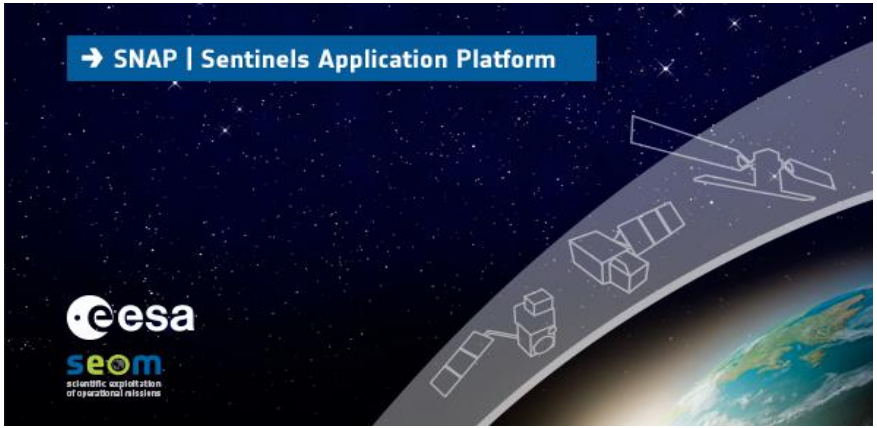
# RUS Service – Training activities

## Proposed training plan for RUS service

- Through different event types
  - Invited (turnkey events)
  - Co-hosted (contribution to existing events)
  - Fully organised events
- Through existing networks (Copernicus Info sessions - NEREUS, REC, UNOSAT, FAO, UNEP, EARSC, EARSEL…)
- Organisation
  - Duration : from 1 day to a full 5-days course
  - Focus on Europe
  - 27 face-to-face events to be planned over 3 years

# RUS Service and You

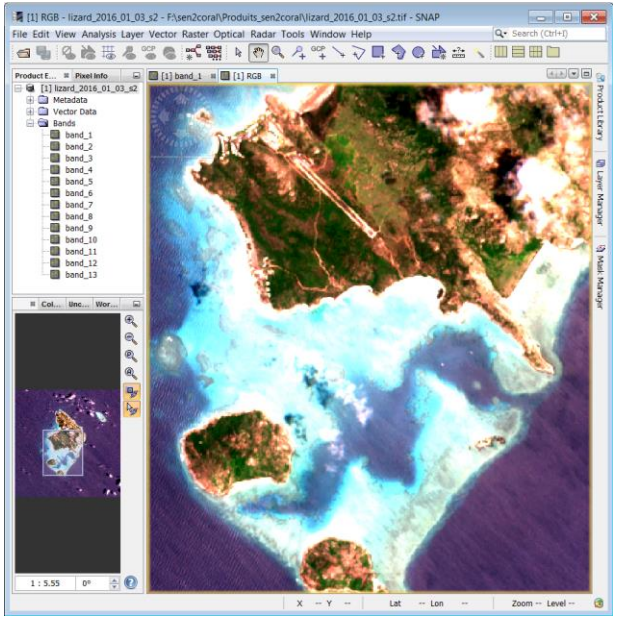




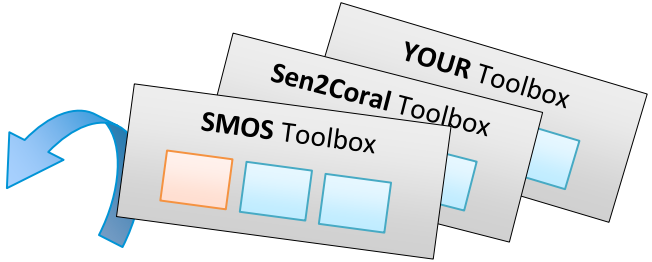

# SENTINEL APPLICATION PLATFORM

# SNAP – What is?

- An open-source, flexible and re-usable scientific Toolbox
- Based on the strong software heritage of BEAM & NEST
- With specific Sentinel data exploitation tools
- Multi-mission :
  - Sentinel-1/2/3
  - Envisat, MODIS, SeaWIFS, …
  - AVHRR, SMOS, Chris-PROBA, SPOT VGT, …
  - LANDSAT, RapidEye, SPOT
  - Generic formats : NetCDF, GeoTIFF, HDF…
- Modular and extensible



# SNAP – Architecture


**Sentinel-1** Toolbox (S1TBX)

**Sentinel-2** Toolbox (S2TBX)

**Sentinel-3** Toolbox (S3TBX)

**SNAP** Desktop

**SNAP** Engine

**NetBeans** RCP

**GeoTools**

**JAI**

**NetCDF**

**...**

**Java** SE 8 Platform

**Python**

# SNAP – Core features

- Common architecture and **data model** for all Toolboxes
- Very fast **image display and navigation** even of giga-pixel images
- Graph Processing Framework (**GPF**)
- Advanced **layer management** allows adding and manipulation of new overlays such as images of other bands, images from WMS servers or ESRI shapefiles
- Flexible **band arithmetic** using arbitrary mathematical expressions
- Accurate **reprojection** and **ortho-rectification** to common map projections
- **Geo-coding** and **rectification** using ground control points
- Automatic SRTM **DEM** download and tile selection
- **Multithreading** and **Multi-core** processor support

# SNAP – Extensibility

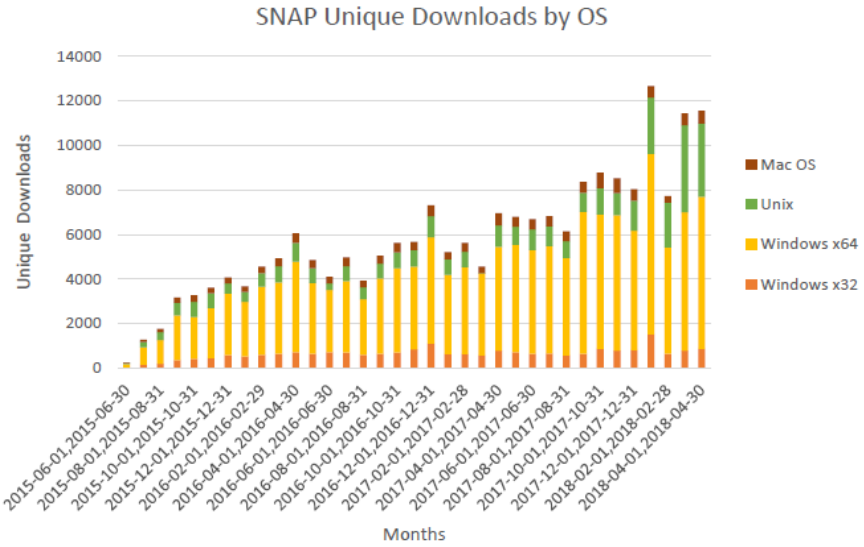
- **Java** as the native language. A Java API to
  - Develop your own application (cli or gui)
  - Extend the existing application by adding modules

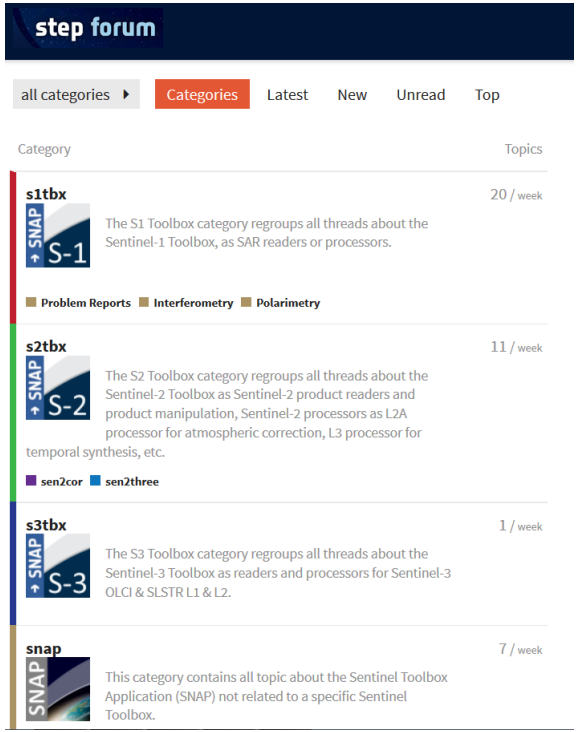

- **Python** as a main extension language.
  - SNAPPY: The Python API mimics the Java API
  - Write your own Python scripts leveraging on SNAP
  - Extend SNAP by developing modules in Python (numpy, scipy, …). Useful for bridging your C/C++ library


- Standalone **Tools Adapter** module
  - Use your own cli tools as if they were a native SNAP module
  - GDAL, Orfeo Toolbox, Sen2Cor, Sen2Three, ...

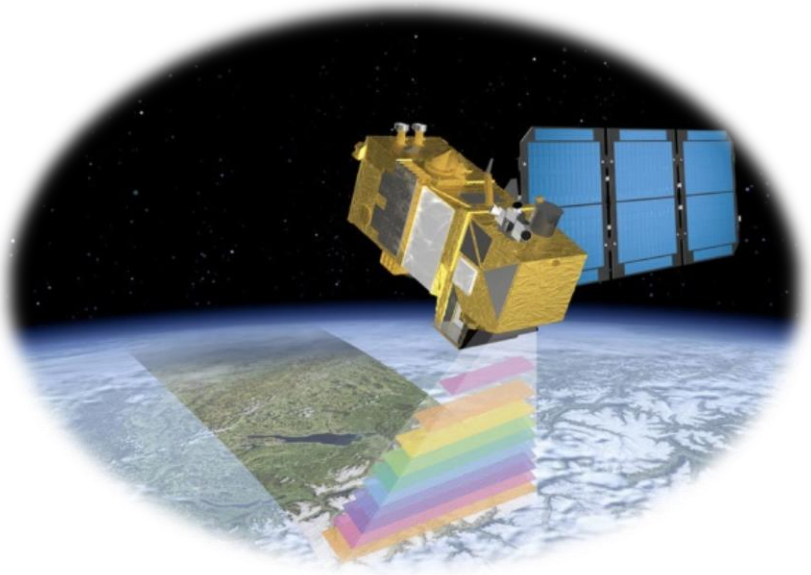
# SNAP – Community

- Our website : **step.esa.int**
  - Get it : **http://step.esa.int/main/download/**
  - Ask questions, discuss anything SNAP or Sentinel data related : **http://forum.step.esa.int**
  - Tutorials, News, Links, related software, ...

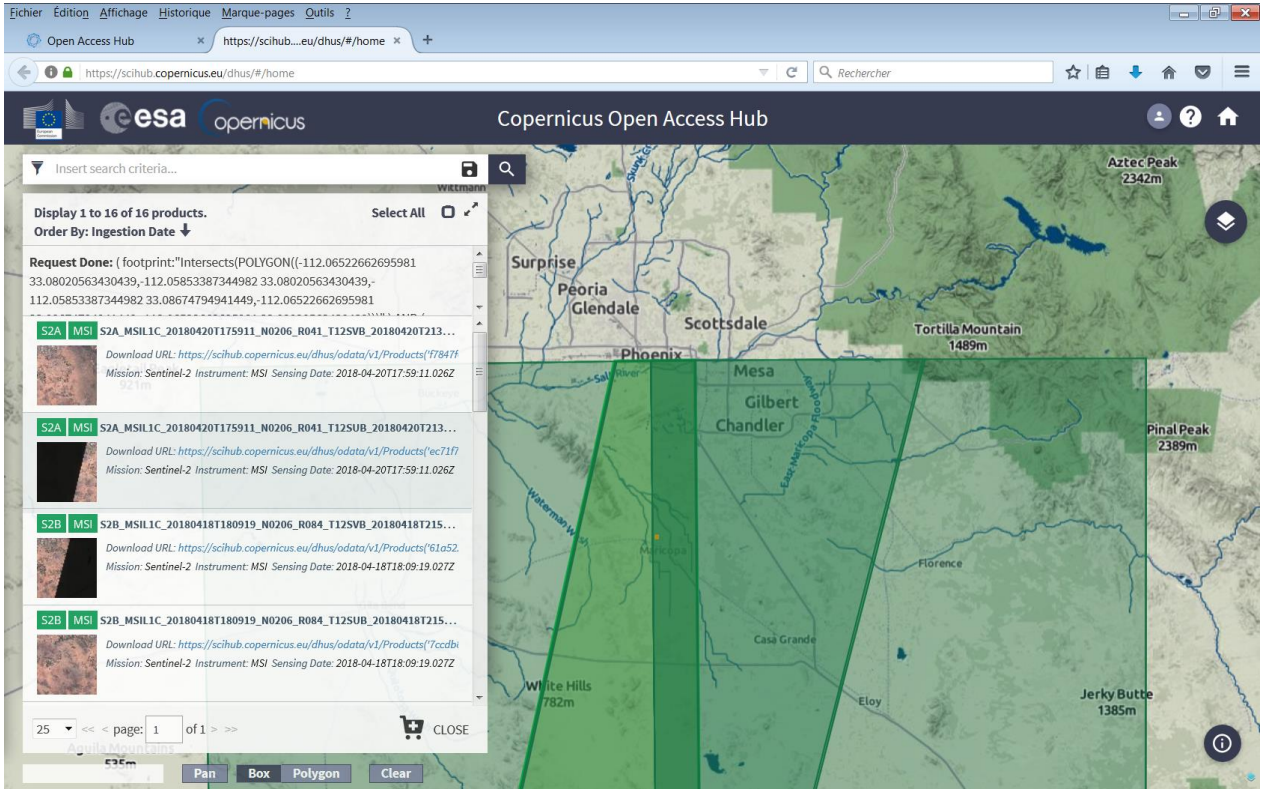


# 3. Downloading and opening a S2 image



# 3. Downloading a S2 product



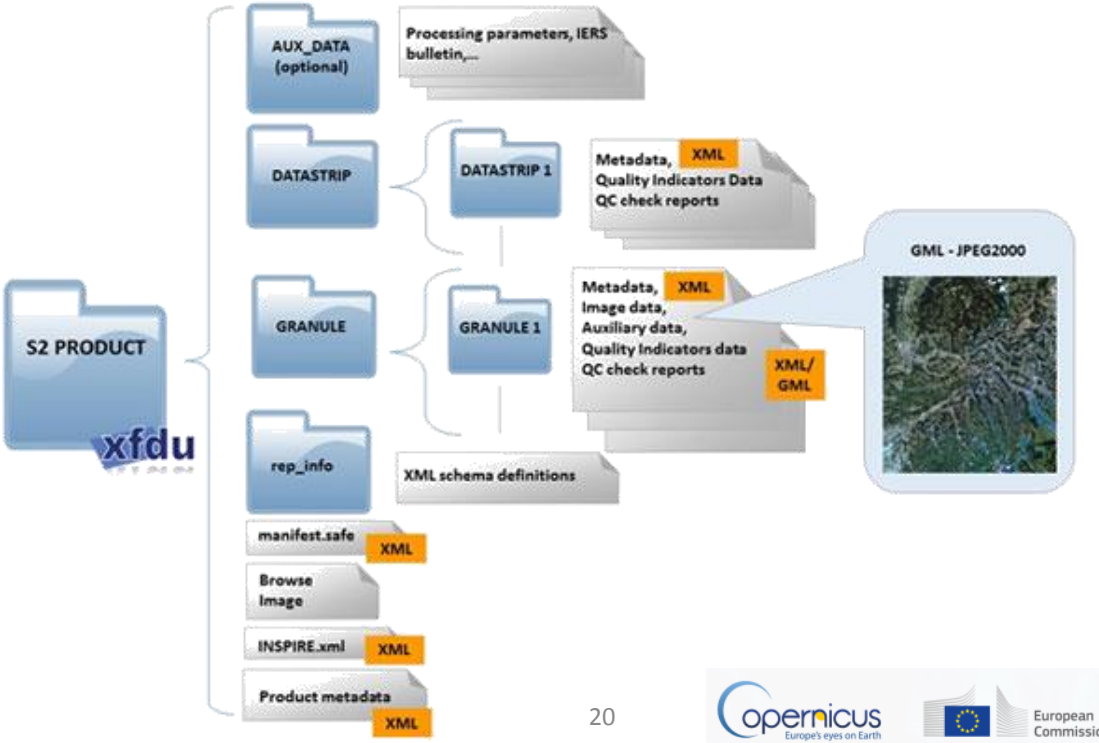
# Opening a S2 product

## Open a Sentinel 2 product:

**S2A_OPER_PRD_MSIL1C_PDMC_20160424T021342_R084_V20160423T182433_20160423T182433.SAFE**

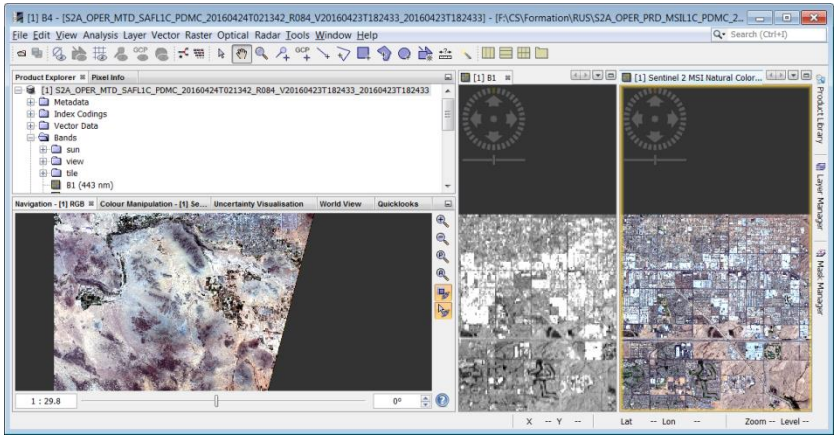
## See product content

- Metadata
- VectorData
- Bands
- Masks

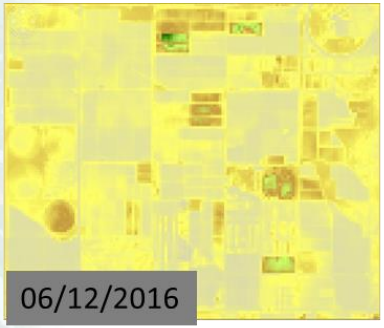


# Analyzing the product

- Open B1
- Display an RGB composite: right click on product, choose profile
- Multi-size
- Tile window
- Metadata
- Use the navigation view to spatially link both window
- View level: pyramids
- Add layers



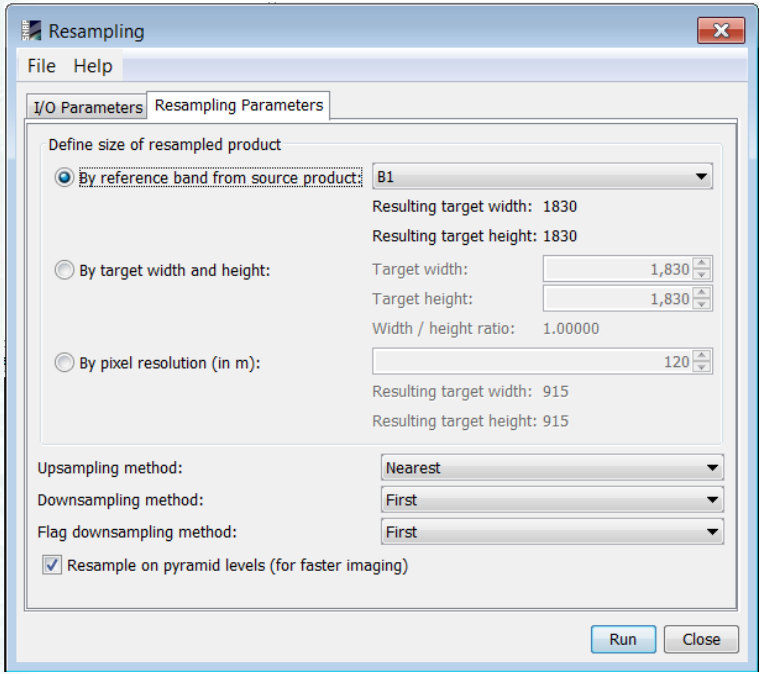
# 4. Processing a S2 product: vegetation

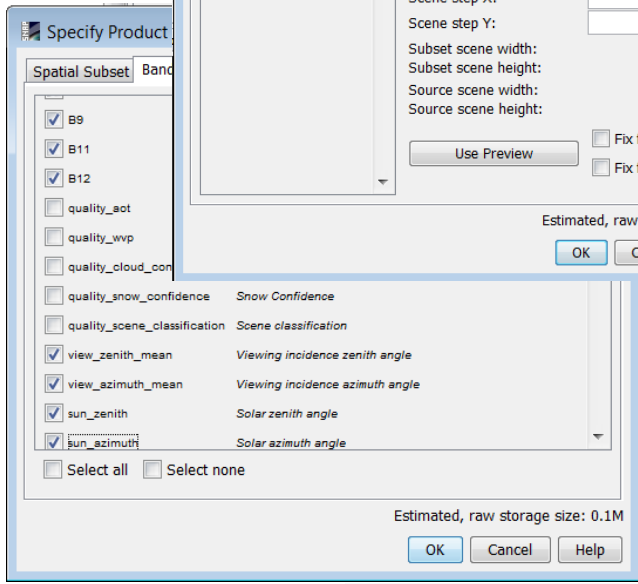


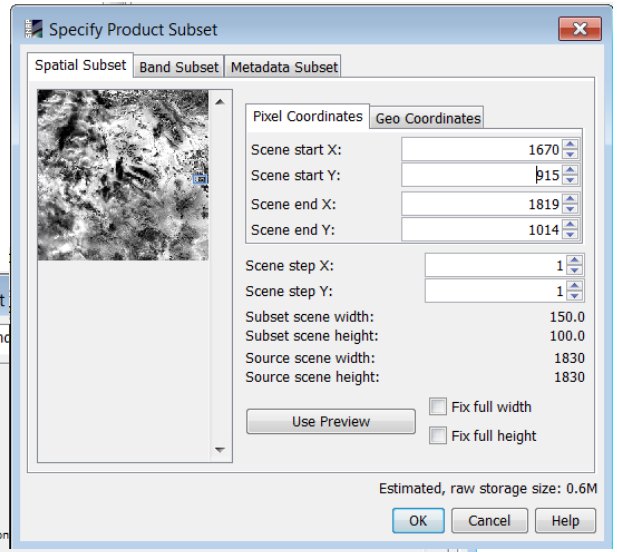



# Pre-processing

- **Atmospheric corrections?**
- **Resample**
- **Subset**



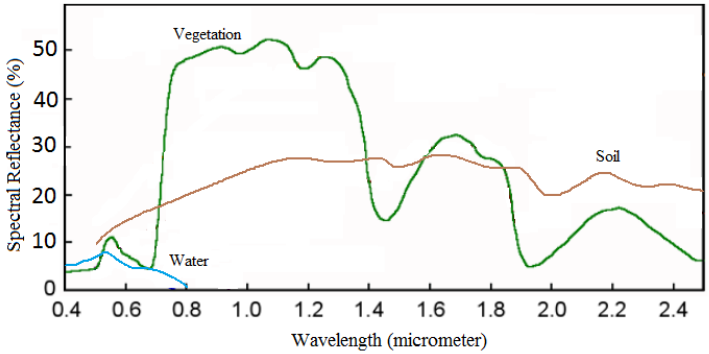
# Previous concepts

## Spectral signature

- Reflectance/emittance as a function of wavelength. The spectral signature of an object is a function of the incidental EM wavelength and material interaction with that section of the electromagnetic spectrum.
- Each material has a unique signature, therefore it can be used for material classification

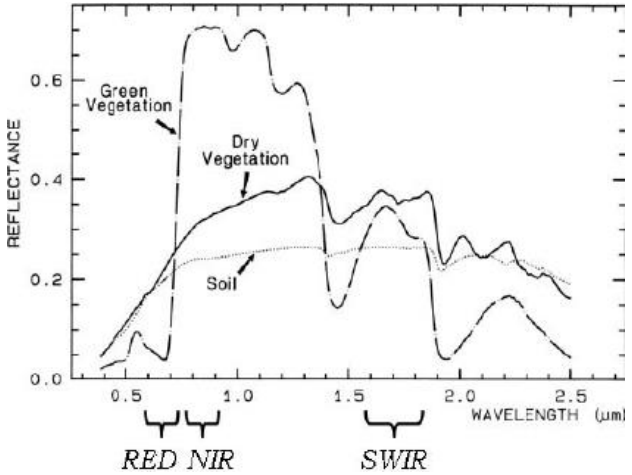
## Radiometric Indices

- Convenient way to resume information
- Exploit the particular spectral properties of vegetation
- Depend on spectral response of the sensors
- Potentially unlimited number of combinations



# NDVI (Normalized Diff.Vegetation index)

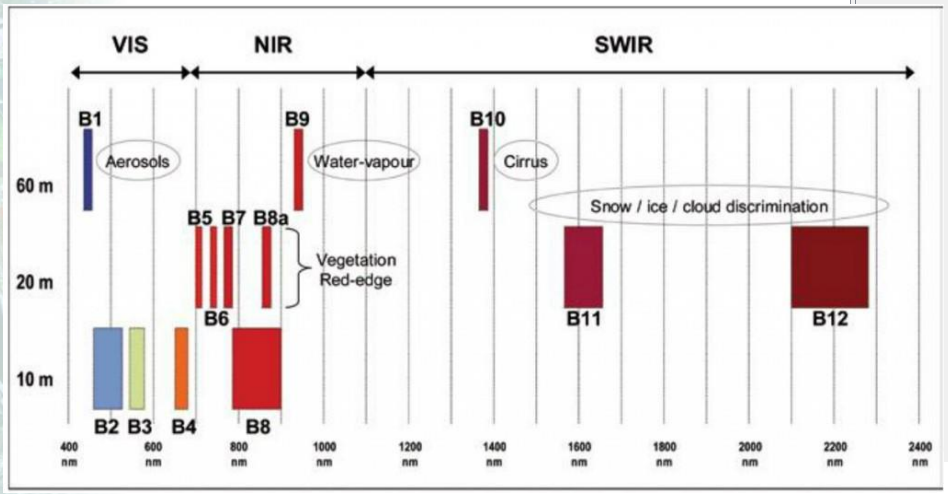
- The spectral signature of vegetation shows a abrupt rise of the reflection level at 0,7 µm. So much more active the chlorophyll of the plants, so much bigger is the rise
- The NDVI composes a measurement for the photosynthetic activity and is strongly in correlation with density and vitality of the vegetation.
- The normalizing reduces topographic and atmospheric effects and enables the simultaneous examination of a wide area.

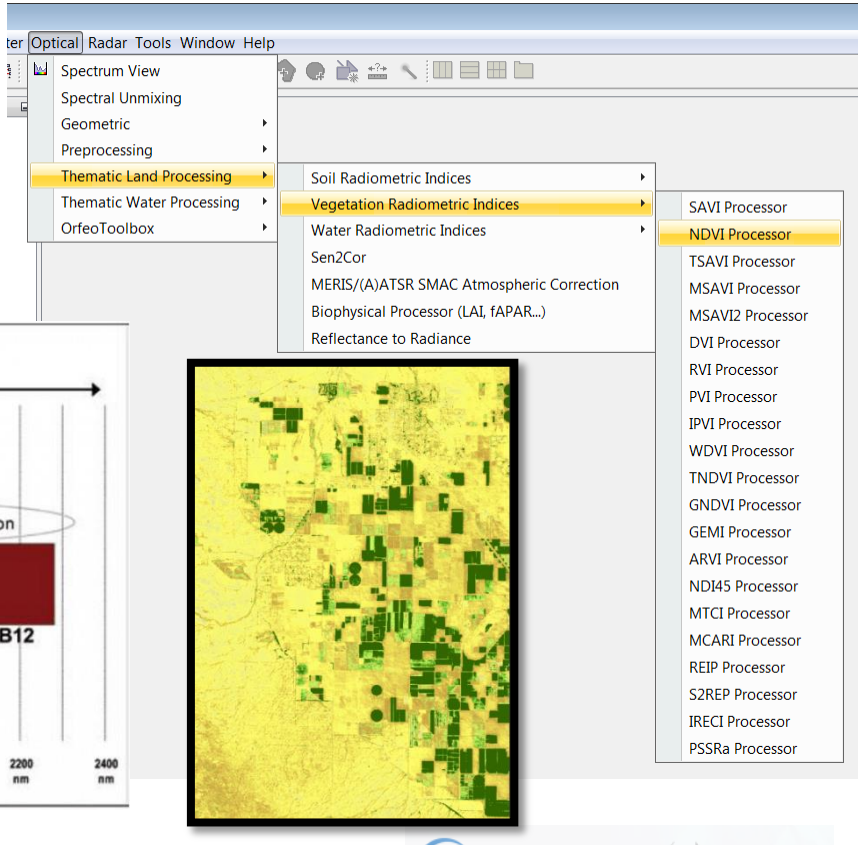



$$NDVI = \frac{(NIR - Red)}{(NIR + Red)}$$

# Vegetation index in SNAP

- Via **BandMaths**
- Via **Vegetation Radiometric Indices**:



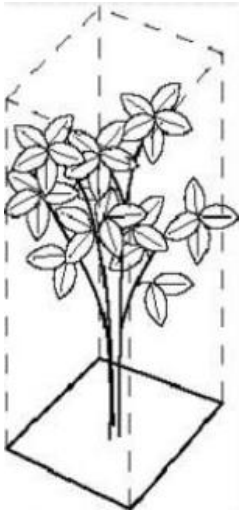



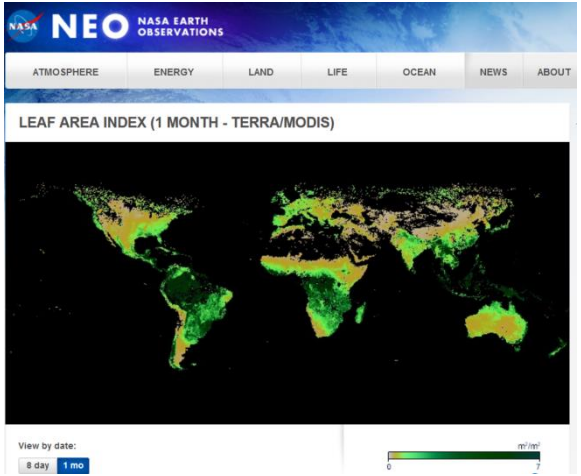
# Biophysical parameters

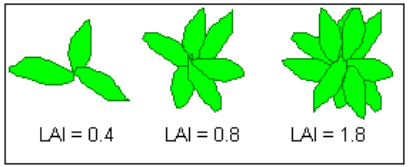
- **Absolute quantification** of the biomass and the nitrogen nutrition, via calculation of different indicators:
  - fCover - green cover fraction
  - FAPAR - fraction of absorbed photosynthetically active radiation
  - LAI - leaf area index
  - Chlorophyll content
- **Robust and independent** from observation conditions
- Complex processing algorithms

# Leaf Area Index

- The half the total developed area of green leaves per unit of ground horizontal surface area [units: m2 m-2]
- Determines the size of the interface for exchange of energy
- Indication of canopy density, biomass
- Can be used to predict productivity and crop growth

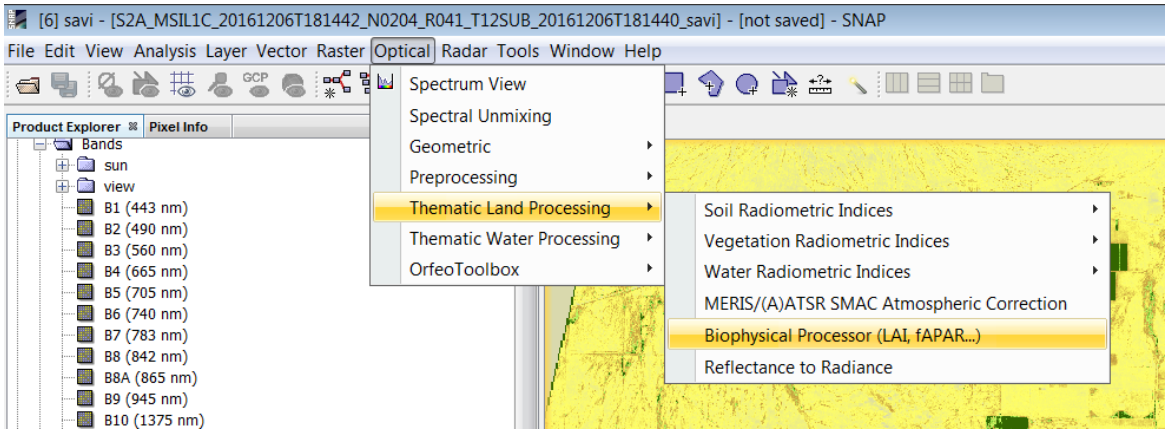




# Compute LAI in SNAP

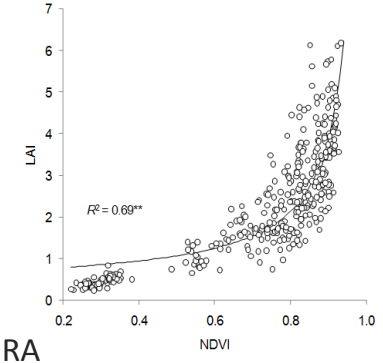
- **Biophysical processor**



# Biophysical Parameters - Computation

## Empirical methods

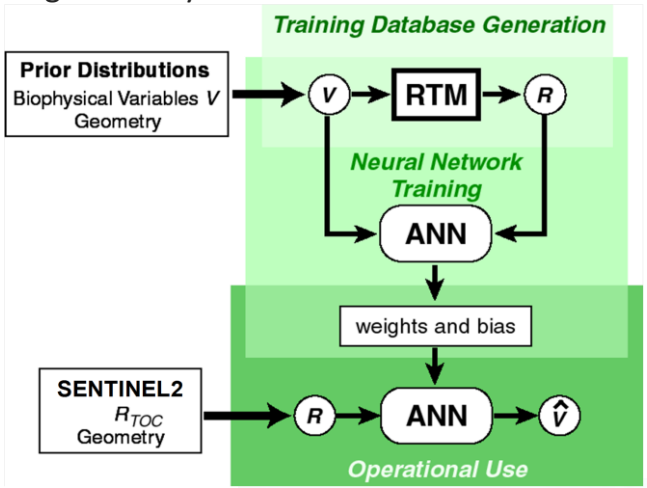
- Establishment of a statistical relationship between a vegetation index or the reflectance and a biophysical variables




## Physical methods

- By solving the radiative transfer equations
- Use of radiative transfer model:
  - PROSPECT: leaf model
  - SAIL: canopy model
  - PROSAIL: combination, simple, the most used

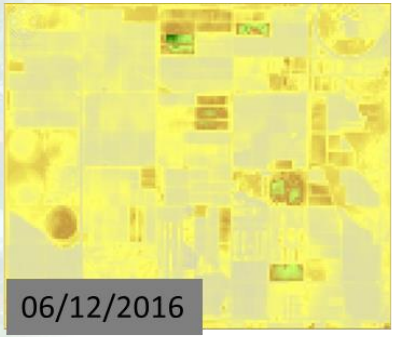
Algorithm by INRA




## Hybrid methods

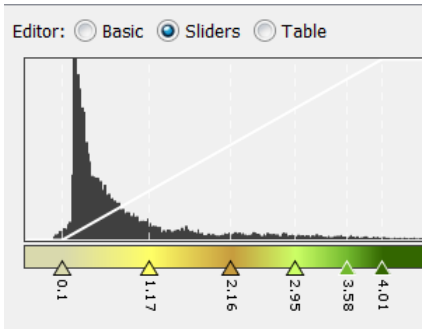
# Biophysical Parameters - Computation

- Repeat processes to obtain the LAI with the other L2A available images.
- Is there an evolution in the crops?



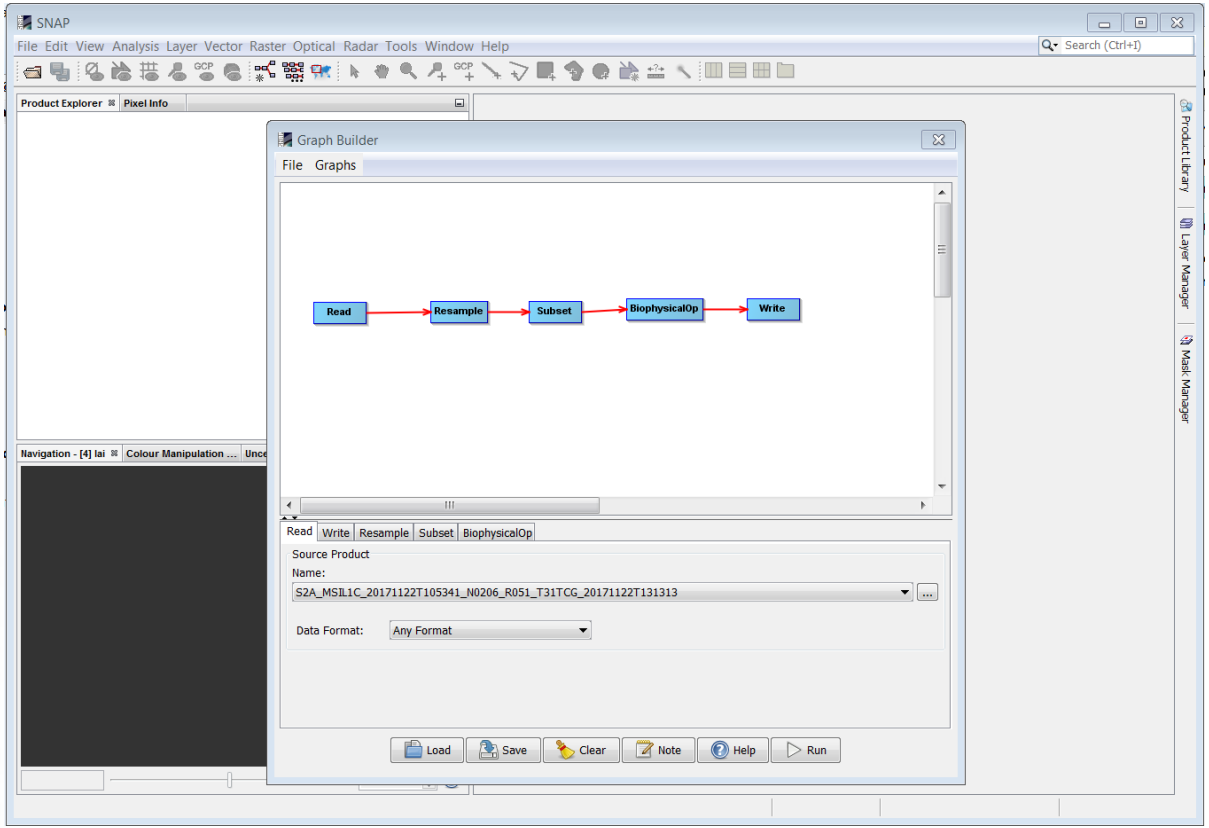






# 5. Automate Processing using GPT

# Graph Builder



# Building a graph

- Export to xml file
- Analyze the graph file
- Is it possible to reconfigure it and use it for automate processing?

## Graph Processing Tool (GPT)

- **Execute operators via command-line**
- **Execute graphs (stored as XML files)**

```
Usage:
  gpt <op>|<graph-file> [options] [<source-file-1> <source-file-2> ...]

Description:
  This tool is used to execute SNAP raster data operators in batch-mode. The
  operators can be used stand-alone or combined as a directed acyclic graph
  (DAG). Processing graphs are represented using XML. More info about
  processing graphs, the operator API, and the graph XML format can be found
  in the SNAP documentation.

Arguments:
  <op>                Name of an operator. See below for the list of <op>s.
  <graph-file>        Operator graph file (XML format).
  <source-file-i>     The <i>th source product file. The actual number of source
                      file arguments is specified by <op>. May be optional for
                      operators which use the -S option.
```

# G P T

## Main option

- **-h :**Displays command usage. If <op> is given, the specific operator usage is displayed.
- **-t <file> :** The target file. Default value is 'target.dim'.
- **-S<source>=<file> :** Defines a source product. <source> is specified by the operator or the graph. In an XML graph, all occurrences of ${<source>} will be replaced with references to a source product located at <file>.
- **-P<name>=<value> :** Defines a processing parameter, <name> is specific for the used operator or graph

## First steps

**Run: gpt NdviOp –h**

**Run NdviOp with the Sentinel-2 L2 product at 60 m**

# GPT-Demo

**Modify the ‘biophysical’ graph file for accepting input parameters**

**Create an script for launching the script with different products**

**Integrate the downloading into the script**