



BRING THE USERS CLOSER TO THE DATA

VIRTUAL RESEARCH ENVIRONMENT

PROBA-V's Mission Exploitation Platform (MEP) complements the PROBA-V user segment by offering an operational Exploitation Platform on the data and derived products to bring the users closer to the data. The MEP offers multiple online applications, such as:

- ➔ **Geo Viewer** where you can view PROBA-V data in full-resolution
- ➔ **Time Series Viewer** to explore the PROBA-V time series for specific regions or single pixels
- ➔ **Notebooks** to create and share documents that contain live code, equations, visualizations, etc.
- ➔ **N-daily compositor** to generate composites for a chosen number of days (N), selected bands, etc.
- ➔ **Virtual Research Environment** with access to the PROBA-V and SPOT-VEGETATION data archive, and a powerful set of tools (e.g. SNAP toolbox, GRASS GIS, QGIS) or to develop and test applications (R, Python, Java and C++)

POWERFUL SET OF TOOLS ON THE CLOUD

Researchers and developers can request a Virtual Research Environment (VRE) with access to the complete PROBA-V and SPOT-VEGETATION data archive, and derived vegetation indices.



DATA

Direct access to the full PROBA-V and SPOT-VEGETATION data archive.



PROCESSING

A large processing cluster for parallel processing to reduce processing time.



TOOLS

A powerful set of tools to work with the data, e.g. SNAP Toolbox, GRASS, GIS, QGIS.



DEVELOP

Support for Python, R, Java, C++ to develop and test your applications.



ENVIRONMENT

Possibility to install your own tools and download additional data.

[PROBA-V-MEP.ESA.INT/VRE](https://proba-v-mep.esa.int/vre)



SAVE TIME ON YOUR PROCESSING

A nice example of how you can save time on your processing using the PROBA-V MEP. Within the Copernicus Global Land Service we process all available PROBA-V 100 m data to create land cover maps for Africa (in development).

The metrics extraction needs +5000 CPU-hours for whole Africa and was processed on the MEP with 517 executors in 22 hours. That's 99.6% time saving!

False color composite showing the biomass density, derived from PROBA-V 100 m. The darker the more biomass.

BRING THE USERS CLOSER TO THE DATA

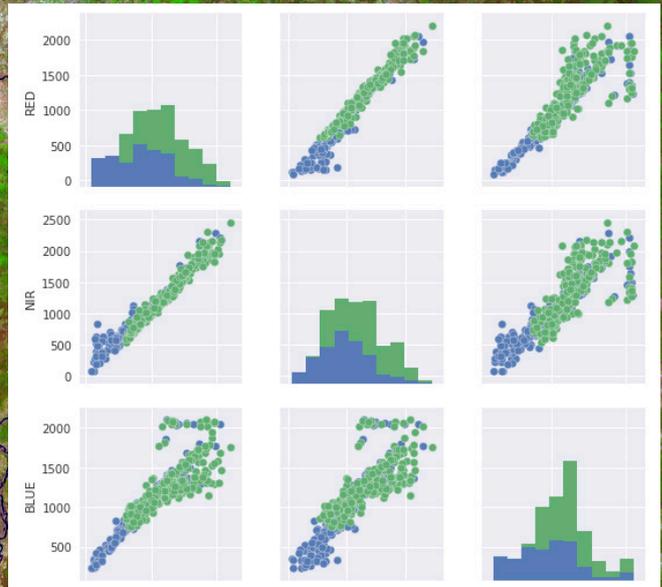
NOTEBOOKS

PROBA-V's Mission Exploitation Platform (MEP) complements the PROBA-V user segment by offering an operational Exploitation Platform on the data and derived products to bring the users closer to the data. The MEP offers multiple online applications, such as:

- ➔ **Geo Viewer** where you can view PROBA-V data in full-resolution
- ➔ **Time Series Viewer** to explore the PROBA-V time series for specific regions or single pixels
- ➔ **Notebooks** to create and share documents that contain live code, equations, visualizations, etc.
- ➔ **N-daily compositor** to generate composites for a chosen number of days (N), selected bands, etc.
- ➔ **Virtual Research Environment** with access to the PROBA-V and SPOT-VEGETATION data archive, and a powerful set of tools (e.g. SNAP toolbox, GRASS GIS, QGIS) or to develop and test applications (R, Python, Java and C++)

PROBA-V MEP NOTEBOOKS

Have you ever read a paper showing nice graphs, without giving a clear explanation on how they were produced? Or do you want to share your code and results with your colleagues easily? Then you should use the PROBA-V MEP Notebooks web application. It offers a solution for all these problems, because it combines a browser based interactive programming environment with the power of our Hadoop processing cluster, connected to the full PROBA-V data archive.

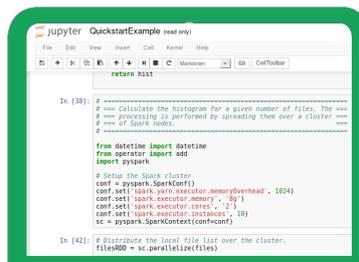


WRITE YOUR CODE

Write your own code and share it with your colleagues, whereby explaining what you do

SHARE EXPERIENCES

Work in a team on one Notebook and build on each other's input



```

return hist

In [38]: #
# == Calculate the histogram for a given number of files. The ==
# == processing is performed by spreading them over a cluster ==
# == of Spark nodes. ==
#

from datetime import datetime
from operator import add
import pyspark

# Setup the Spark cluster
conf = pyspark.SparkConf()
conf.set('spark.park.executor.memoryOverhead', 1024)
conf.set('spark.executor.memory', '5g')
conf.set('spark.executor.cores', '2')
conf.set('spark.executor.instances', 10)
sc = pyspark.SparkContext(conf=conf)

In [42]: # distribute the local file list over the cluster.
filesRDD = sc.parallelize(files)
    
```

FAST PROCESSING

Process fast on our cluster and see the results immediately

DATA ARCHIVE

Use the complete PROBA-V, SPOT-VEGETATION archive and derived vegetation indices, representing a time series of 18 years of data.