MDIS 2019 : interfero tutorial with DiapOTB

P. Durand, V. Poughon, G. Usseglio, M. Azzoni

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

Context

✦HOW to easily download S1 (or S2 data) from PEPS: focus on script PEPS_DOWNLOAD.py

✦ DiapOTB

✦MultiSLC : a script for time series inteferograms or coregistered SLC on a same MASTER image

peps_download.py : options (*script O. Hagolle*)

A script is often more convenient than interactive download, especcialy on clusters

how it works : login PEPS needed

Copy python procedure peps_download.py and peps.txt (login) in the same directory of future SAFE data

module load python

python peps_download.py -a peps.txt -c S1 -p SLC -m SM -d 2016-08-15 -f 2016-10-01 --latmin -21.5 --latmax -21.25 --lonmin 55.5 --lonmax 55.7

unzip files

delete polar vh (usualy not used - save space)
rm ./S1*.SAFE/measurement/s1*-*-*-vh-*-*-*-*-*.tiff

peps_download.py : options (*script O. Hagolle*)

-h, --help *show this help message and exit*
-l LOCATION, --location=LOCATION *town name (pick one which is not too frequent to avoid confusions)*
-a AUTH, --auth=AUTH *Peps account and password file MDIS.txt tempo@mdis.fr Mdis2019!*
-w WRITE_DIR, --write_dir=WRITE_DIR *Path where the products should be downloaded*
-c COLLECTION, --collection=COLLECTION *Collection within theia or PEPS collections (S1 S2 etc)*
-p PRODUCT_TYPE, --product_type=PRODUCT_TYPE *GRD, SLC, OCN (for S1) | S2MSI1C S2MSI2A S2MSI2Ap (for S2)*
-m SENSOR_MODE, --sensor_mode=SENSOR_MODE *EW, IW , SM, WV (for S1) | INS-NOBS, INS-RAW (for S3)*
-n, --no_download *Do not download products, just print curl command*
-d START_DATE, --start_date=START_DATE *start date, fmt('2015-12-22')*
-f END_DATE, --end_date=END_DATE *end date, fmt('2015-12-23')*
--json=SEARCH_JSON_FILE *Output search JSON filename*
--windows *For windows usage*
--M=MASTER IMAGE *set the master image file name (.SAFE)*
-t TILE, --tile=TILE *Sentinel-2 tile number*
--lat=LAT *latitude in decimal degrees*
--lon=LON *longitude in decimal degrees*
--latmin=LATMIN *min latitude in decimal degrees*
--latmax=LATMAX *max latitude in decimal degrees*
--lonmin=LONMIN *min longitude in decimal degrees*
--lonmax=LONMAX *max longitude in decimal degrees*
-o ORBIT, --orbit=ORBIT *Orbit Path number*

peps_download.py : examples

example 1 : python peps_download.py -l 'Toulouse' -a peps.txt -d 2016-12-06 -f 2017-02-01 -c S2ST

example 2 : python peps_download.py --lon 1 --lat 44 -a peps.txt -d 2015-11-01 -f 2015-12-01 -c S2

example 3 : python peps_download.py --lonmin 1 --lonmax 2 --latmin 43 --latmax 44 -a peps.txt -d 2015-11-01 -f 2015-12-01 -c S2

example 4 : python peps_download.py -l 'Toulouse' -a peps.txt -c SpotWorldHeritage -p SPOT4 -d 2005-11-01 -f 2006-12-01

example 5 : python peps_download.py -c S1 -p GRD -l 'Toulouse' -a peps.txt -d 2015-11-01 -f 2015-12-01

Example SM Reunion Island
python peps_download.py -n -a mdis.txt -c S1 -p SLC -m SM -d 2016-08-15 -f 2016-10-01 --latmin -21.5 --latmax -21.25 --lonmin 55.5 --lonmax 55.7

peps_download.py : SM Reunion on disk

python peps_download.py -n -a mdis.txt -c S1 -p SLC -m SM -d 2016-08-15 -f 2016-10-01 --latmin -21.5 --latmax -21.25 --lonmin 55.5 --lonmax 55.7

curl -k -o search.json https://peps.cnes.fr/resto/api/collections/S1/search.json?box=55.5,-21.5,55.7,-21.25\&startDate=2016-08-15\&completionDate=2016-10-01\&maxRecords=500\&productType=SLC\&sensorMode=SM

##############################

6 products to download

##############################

```
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current Dload  Upload   Total   Spent    Left  Speed
100 36401    0 36401    0     0  82979      0 --:--:-- --:--:-- --:--:-- 83107
```

S1B_S4_SLC__1SDV_20160929T014610_20160929T014634_002277_003D71_788E disk

S1B_S6_SLC__1SDV_20160928T145211_20160928T145235_002270_003D30_C8EC disk

S1A_S4_SLC__1SDV_20160923T014652_20160923T014716_013173_014F12_6794 disk

S1A_S6_SLC__1SDV_20160922T145253_20160922T145317_013166_014EDB_653E disk

S1A_S4_SLC__1SDV_20160818T014650_20160818T014715_012648_013DB1_0969 disk

S1A_S6_SLC__1SDV_20160817T145252_20160817T145315_012641_013D72_EE0D disk

peps_download.py : Reunion SM on tape

python peps_download.py -n -a mdis.txt -c S1 -p SLC -m SM -d **2015**-08-15 -f **2015**-10-01 --latmin -21.5 --latmax -21.25 --lonmin 55.5 --lonmax 55.7

curl -k -o search.json https://peps.cnes.fr/resto/api/collections/S1/search.json?box=55.5,-21.5,55.7,-21.25\&startDate=2015-08-15\&completionDate=2015-10-01\&maxRecords=500\&productType=SLC\&sensorMode=SM

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 13227    0 13227    0     0  44104      0 --:--:-- --:--:-- --:--:-- 44385
```

S1A_S6_SLC__1SDV_20150928T145246_20150928T145310_007916_00B0E2_30CD **tape**
S1A_S6_SLC__1SDV_20150823T145245_20150823T145309_007391_00A2AA_7A2B **tape**

##############################
2 products to download
##############################

DiapOTB Context

Technical context : Port a legacy processing chain for SAR interferometry called Diapason as an Orfeo ToolBox remote module (DiapOTB) funded by CNES.

Scientific context : The differential SAR interferometry (DInSAR)

- ✦ Processing of two SAR images of the same portion of the Earth's surface taken at different time.
- ✦ Highlighting differences between SAR images to analyse potential events (earthquake, destruction, ...).

Aims :

- ✦ Launching a complete DInSAR chain with a simple python script or standalone applications.
- ✦ Re-using procesing.

Data : Sentinel-1 with SM and IW mode, Cosmo SM and Spotlight, in the near future TerraSAR-X

DiapOTB Applications

General SAR Applications :

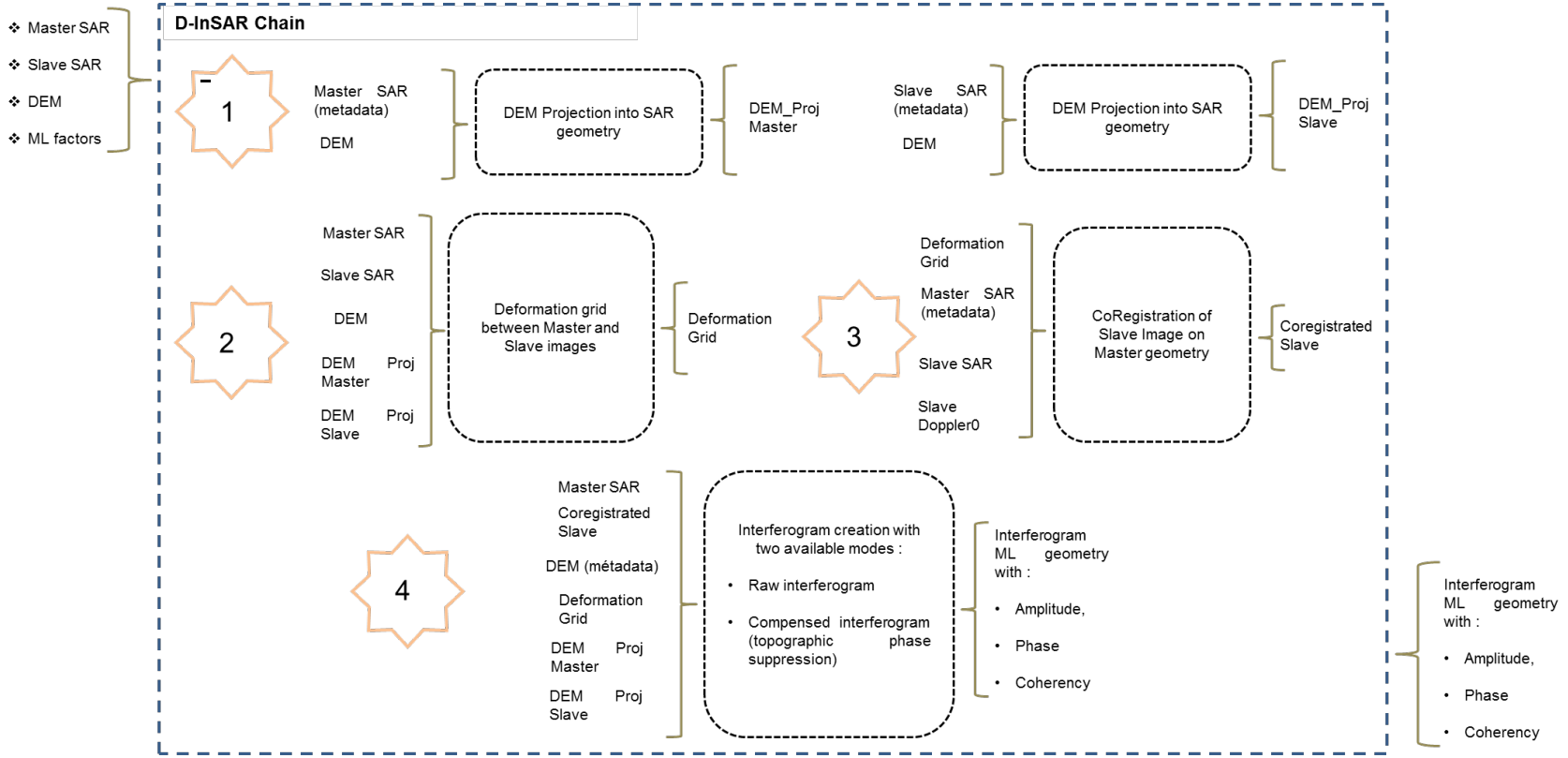
- Doppler0 estimation
- MultiLook
- Simulation of SAR image from Digital Elevation Model
- Projection of Digital Elevation Model into SAR geometry

Specific Interferometry Applications :

- Deformation grids creation
- CoRegistration
- Interferogram

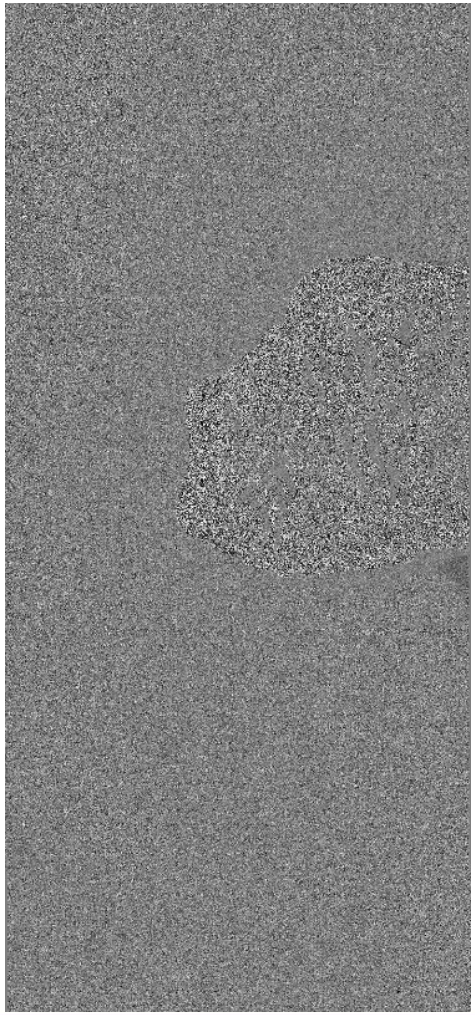
DiapOTB Chain

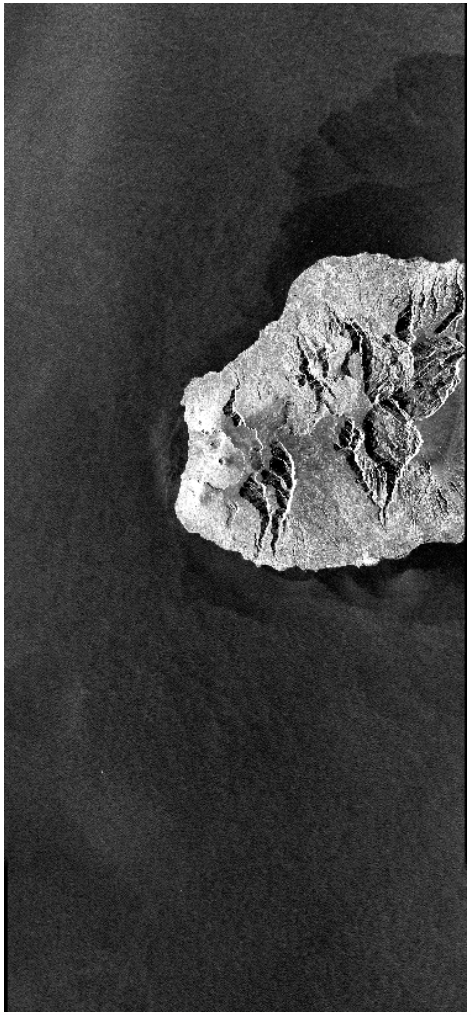
DInSAR Chain in Python : Use previous applications as parts of a larger processing workflow to get a final interferogram from two SLC (single look complex) images and a digital elevation model.



Example of DiapOTB applications : SARMultiLook

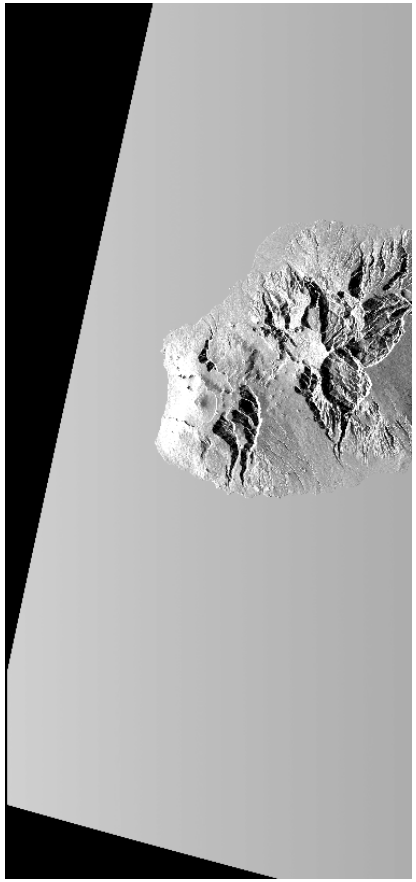
SLC Image



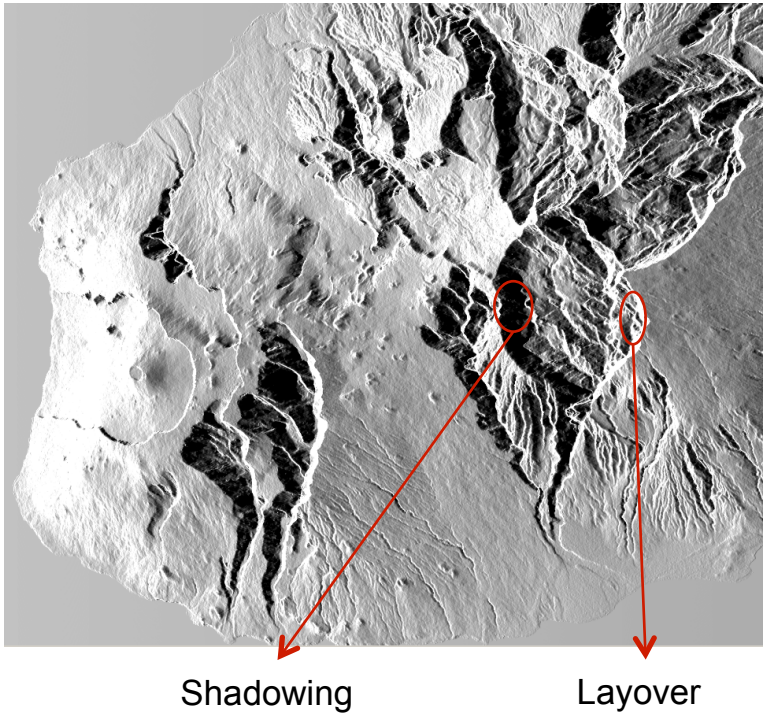


Real image ML (3x3)

Example of DiapOTB application : SARDEMToAmplitude



Get an amplitude simulated image from only a DEM and SAR metadata



A generic filter was developed to estimate amplitude. This filter can be easily extended to different purposes such as calculating Shadowing or Layover masks.

Example of results for DiapOTB chain

Data : Sentinel-1 StripMap mode (Reunion)

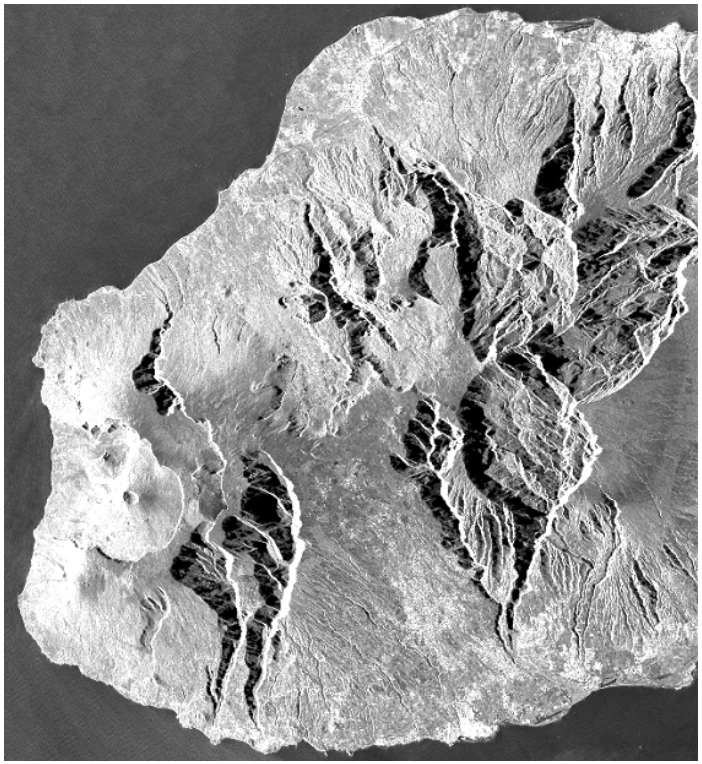
Results : Interferogram with 3x3 factors for averaging and representing by a Vector Image :

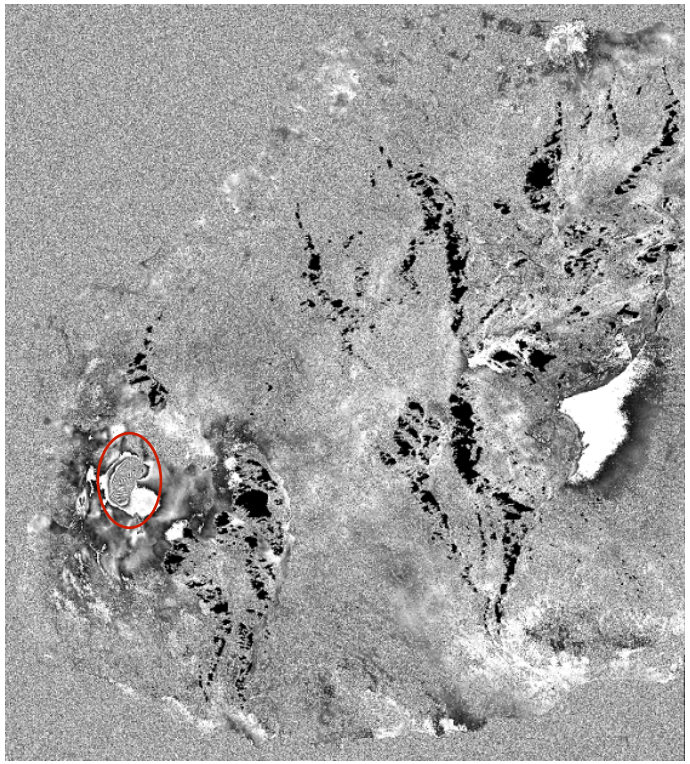
- Band 1 : Amplitude
- Band 2 : Phase
- Band 3 : Coherency
- Band 4 : IsData boolean to indicate if output pixels are consistent

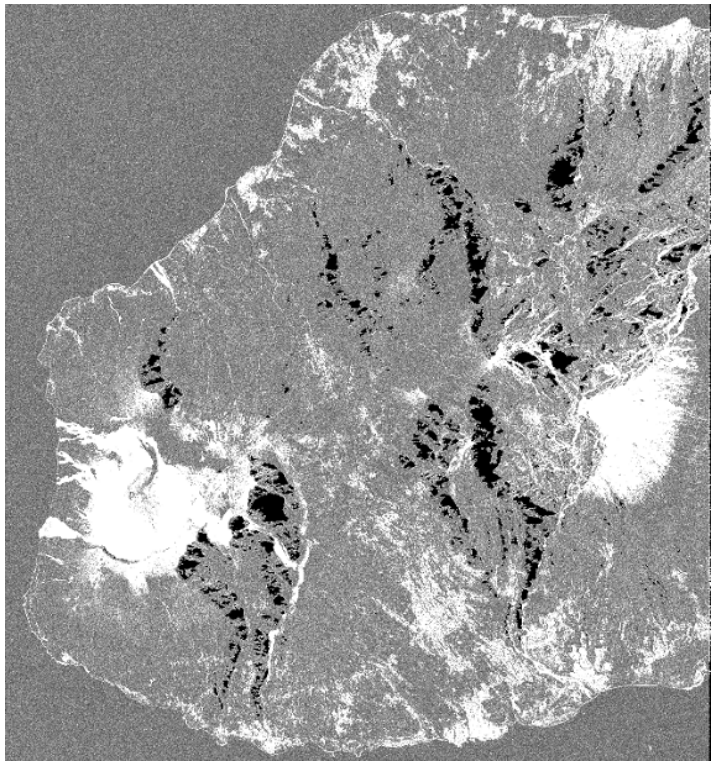
Performances :

- DiapOTB applications are multi-threaded applications
- Runs on CNES HPC cluster (HAL)
- Around 8 minutes for the complete StripMap chain on 1 node with 24 Cpus

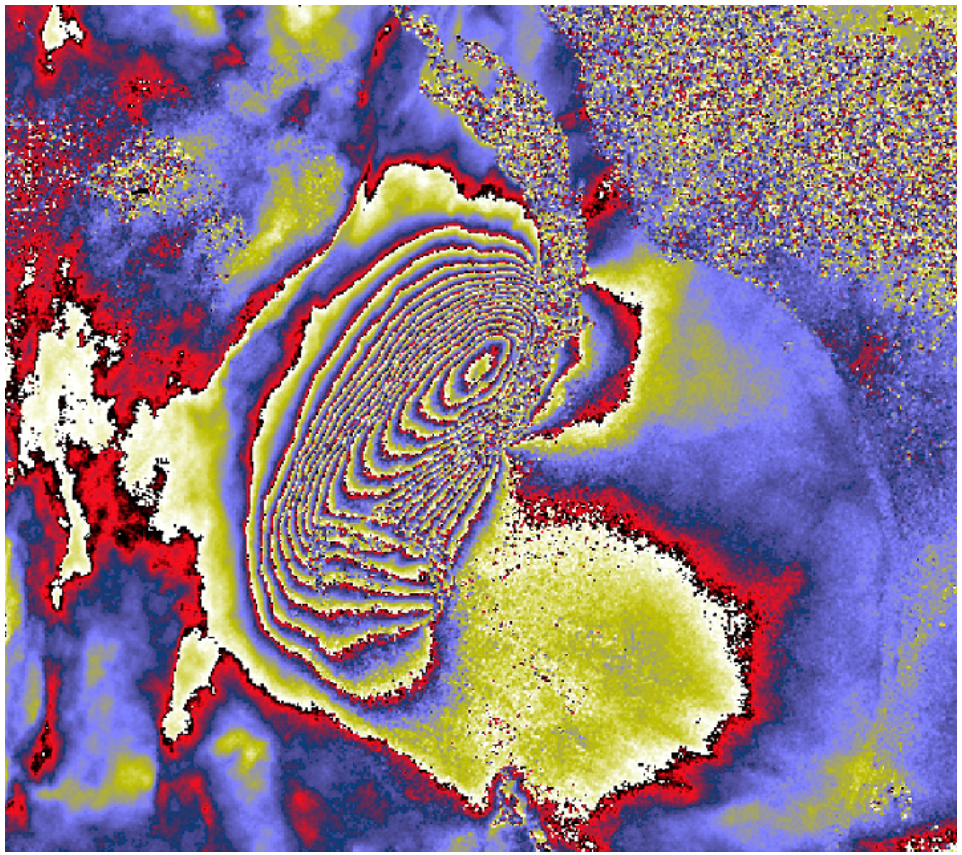
Example of results for DiapOTB chain







Example of results for DiapOTB chain



Displacement near piton de la fournaise between our input images taken on 2016/08/18 and 2016/09/29 with S1-A and S1-B respectivily.

DiapOTB prospects

• Available Now

gitlab.orfeo-toolbox.org/remote_modules/diapotb

Support for S1 SM and S1 IW and Cosmo SM and Spotlight

• Phase unwrapping not available yet : any willing person welcome !

DiapOTB scripts examples

example :

Multi-look

qsub -I -X -l select=1:ncpus=12:mem=60000Mb -l walltime=10:00:00 *to affect RAM and CPUs*

export MODULEPATH=/Modules_DiapOTB/modulefiles:$MODULEPATH

module load diapOTB

export ITK_GLOBAL_DEFAULT_NUMBER_OF_THREADS=12 *for multithreading*

cd /working_directory/

Configure parameters in .json
python $DIAPOTB_HOME/python_src/diapOTB_S1IW.py config.json *for launching*

Chaîne de traitements D-InSar

Boîte à outils

Orfeo Toolbox

_ Outils de traitements raster et vecteur.

GDAL

_ Bibliothèque d'outils pour traitements Raster/Vecteur.
_ API Python

Python

_ Langage d'écriture du script

PEPS

_ Plateforme du CNES destinée au téléchargement des images Sentinel.

Cluster Linux

_ Cluster CNES, Grappe de serveurs CNES, outils de calculs puissant.
_ 24 cpus, 120Go, 24 thread.

Diap OTB

_ Issu de la chaîne Diapason.
_ Outils de traitements interférométrique.

Chaîne de traitements D-InSar

Principe de la chaîne D-InSar_MultiSlc

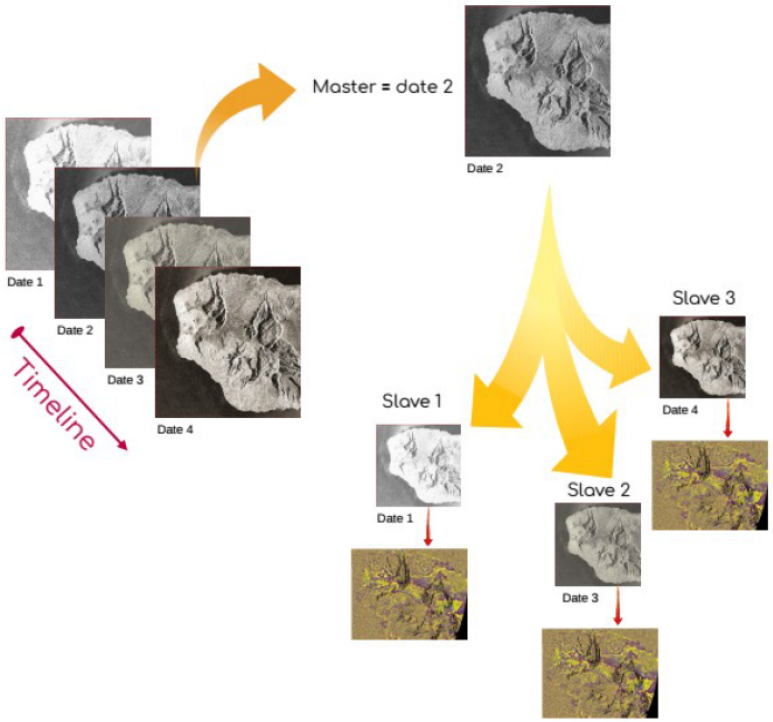
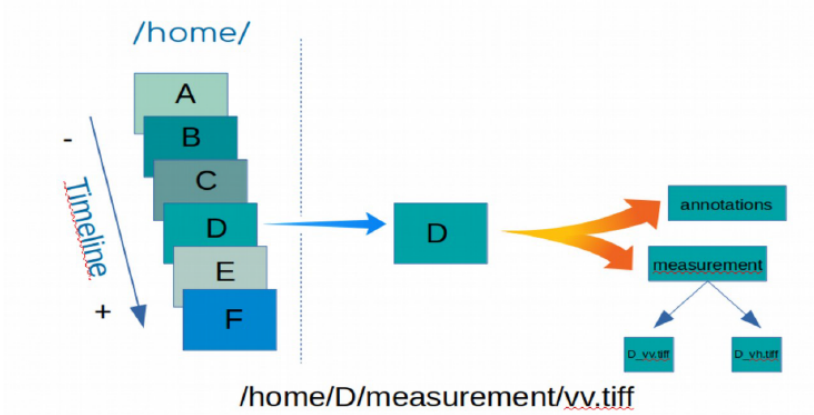


Illustration d'un chemin de fichier pour Sentinel 1



Notre script doit pouvoir retrouver le chemin d'après le nom seul.

Chaîne de traitements D-InSar

Liste des options du script :

-d : date de début

-f : date de fin

-e : dates à exclure

-m : image maîtresse

-b : bursts à traiter IW

-roi : Définir une ROI (ft : ulx uly lrx lry)

-i : interférogramme (activée par défaut)

-o : Orthrectification

-l : Light version (activée par défaut)

-ram : Définir la ram (4Go à par défaut)

-spacingxy : Résolution spatiale pour l'orthorectification (rééchantillonnage 0.0001° par défaut)

-ml : Multilook range x azimuth

Principes de fonctionnement

. Lancement en ligne de commande

. Le script se lance en duo avec un fichier de configuration

Ml Range Azimut

Grid Treshold
Grid Gap
Gain [...]

.Choix date début, date fin, et si besoin des dates à exclure

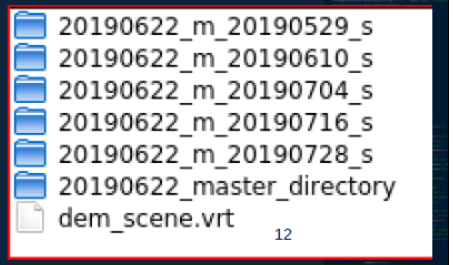
.Choix de la maîtresse

.Sortie interférogramme ou non.

. Orthorectification ou non.

.Light mode ou full content

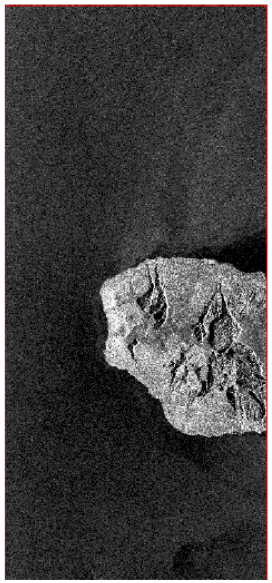
Organisation du dossier de sortie



Chaîne de traitements D-InSar

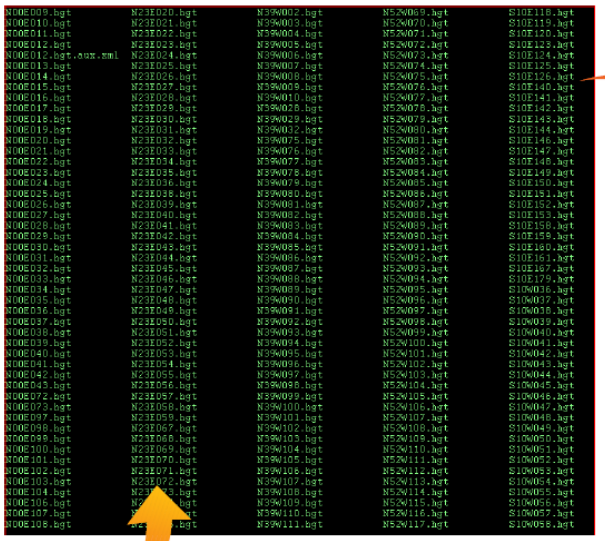
Création automatique du MNT de la zone

Le script intègre une fonction de S1Tilling* pour la création d'un Modèle Numérique de Terrain



Création d'une empreinte shapefile de l'image

Intersection de l'image sur les tuiles SRTM



Récupération des tuiles dans un datalake

Création d'un raster virtuel sous GDAL

* S1Tilling : Script de Thierry Koleck (CNES - CESBIO)

Script SAR_MultiSlc activation

python SAR_MultiSlc.py SAR_MultiSlc.json -d 20160607 -f 20160930 -m s1a-s6-slc-vv-20160817t145252-20160817t145315-012641-013d72-002.tiff -o yes

python SAR_MultiSlc.py SAR_MultiSlc.json -d 20160607 -f 20160930 -m s1a-s4-slc-vv-20160818t014650-20160818t014715-012648-013db1-002.tiff -o yes

python QuickMlView.py -in s1a-s6-slc-vv-20160817t145252-20160817t145315-012641-013d72-002.tiff -ml 3 3

python QuickMlView.py -in s1a-s4-slc-vv-20160818t014650-20160818t014715-012648-013db1-002.tiff -ml 3 3