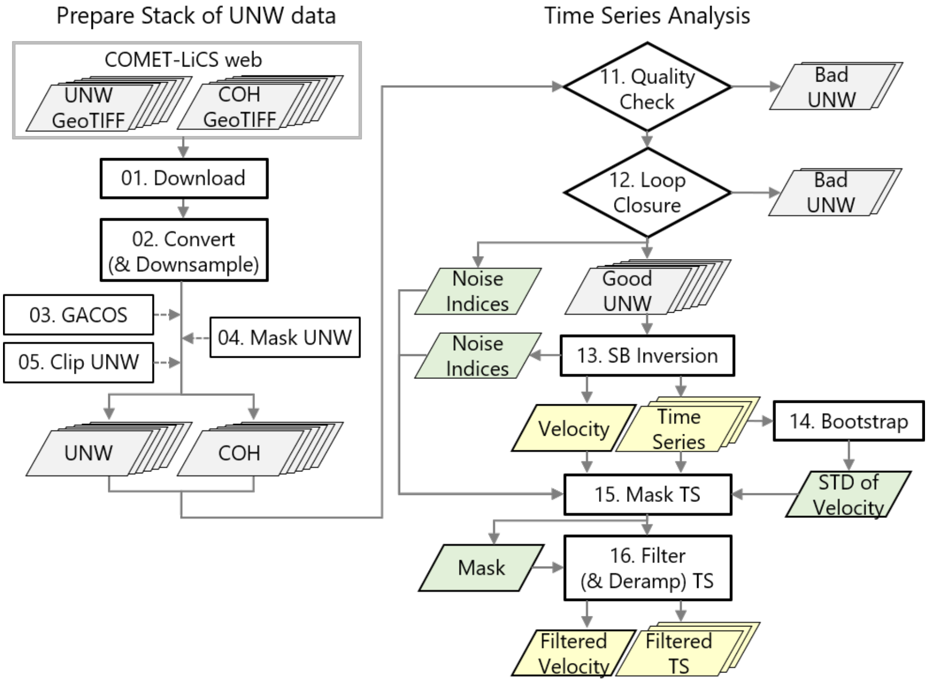
LiCSBAS Tutorial

September 2019 (updated March 2020)
Yu Morishita, University of Leeds, Geospatial Information Authority of Japan

This document contains the InSAR time series analysis tutorial using the LiCSBAS package. The tutorial will take you from a stack of unwrapped and geocoded interferograms downloaded from the COMET-LiCS web to the time series and velocity of the displacement at Campi Flegrei, Italy, from September 2016 to May 2018.

The LiCSBAS package is available on GitHub: https://github.com/yumorishita/LiCSBAS.



LiCSBAS workflow

1. Preparation

LiCSBAS needs the Python3 environment and several modules. To set up the environment, please see https://github.com/yumorishita/LiCSBAS/wiki/1_Installation.

2. Quick Start (Batch Processing)

You can run all the steps from 01-16 or only selected steps with specified parameters at once using a batch script. As the quick start, let's use the prepared batch script for Campi Flegrei from September 2016 to May 2018. Download the batch script:

```
wget https://raw.githubusercontent.com/wiki/yumorishita/LiCSBAS/documents/batch_LiCSBAS_sample_CF.sh
```

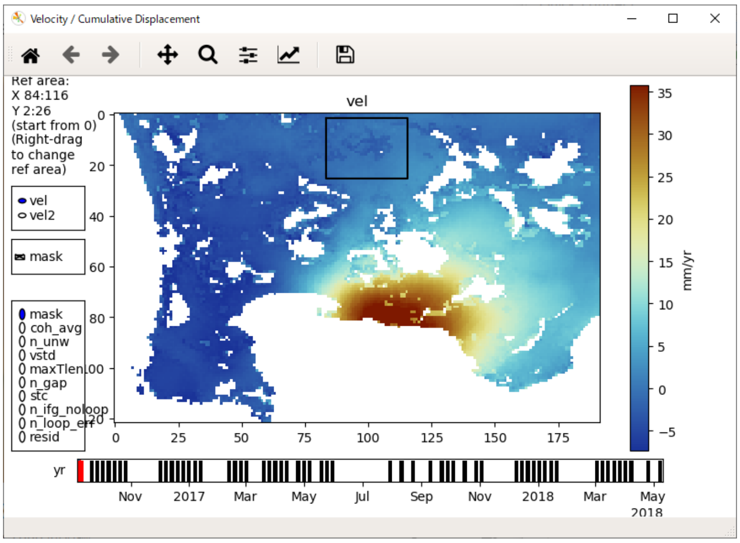
and run it:

```
chmod 764 batch_LiCSBAS_sample_CF.sh
./batch_LiCSBAS_sample_CF.sh
```

then, all the steps will run, and `GEOC`, `GEOCml1`, `GEOCml1clip`, `TS_GEOCml1clip` and `log` directories (~16 GB in total) will be created.

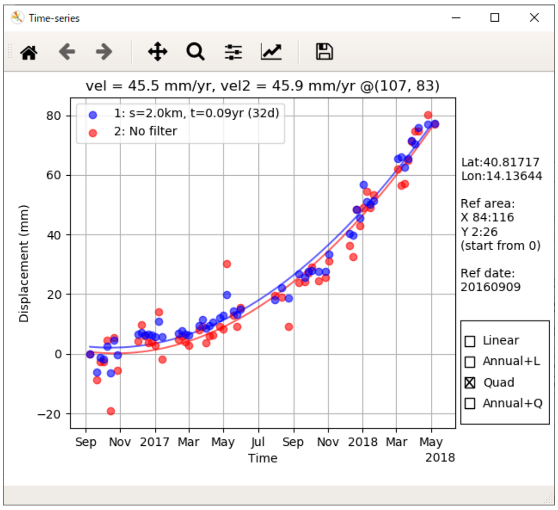
You can display the derived time series and velocity:

```
LiCSBAS_plot_ts.py -i TS_GEOCml1clip/cum_filt.h5 &
```



Velocity / Cumulative Displacement

Ref area:
X 84:116
Y 2:26
(start from 0)
(Right-drag
to change
ref area)



Time-series

vel = 45.5 mm/yr, vel2 = 45.9 mm/yr @(107, 83)

● 1: s=2.0km, t=0.09yr (32d)
● 2: No filter

Lat:40.81717
Lon:14.13644

Ref area:
X 84:116
Y 2:26
(start from 0)

Ref date:
20160909

☐ Linear
☐ Annual+L
☒ Quad
☐ Annual+Q

The sample of `TS_GEOCml1clip` containing the derived time series and velocity can be downloaded from https://raw.githubusercontent.com/wiki/yumorishita/LiCSBAS/sample/LiCSBAS_sample_CF.tar.gz (40MB). Please note that this sample might not be completely the same as what you have created because the available LiCSAR products could be updated.

3. LiCSBAS Step-by-Step

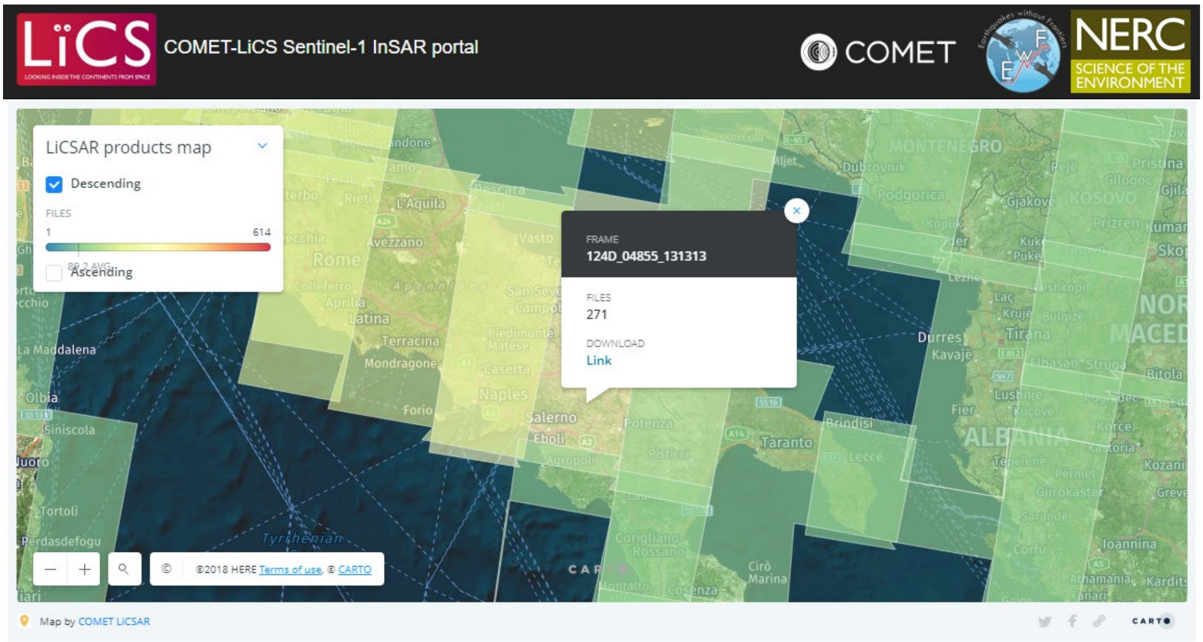
LiCSBAS consists of two major step groups: preparation of stack of unwrapped data (Step 01-05) and time series analysis (Step 11-16).

To see the usage of commands in LiCSBAS, type with -h option, e.g.,

```
LiCSBAS11_check_unw.py -h
```

3.1. Identify frame ID

First, you need to identify the frame to process which includes your area of interest. LiCSAR products are provided for each predefined LiCS-frame. You can see the area of frames, frame IDs, and the number of available interferograms on COMET-LiCS web (https://comet.nerc.ac.uk/COMET-LiCS-portal/).


LiCS
COMET-LiCS Sentinel-1 InSAR portal
COMET
NERC SCIENCE OF THE ENVIRONMENT
LiCSAR products map
☑ Descending
FILES
1
614
□ Ascending
Map by COMET LiCSAR
©2018 HERE Terms of use, © CARTO

FRAME
124D_04855_131313

FILES
271

DOWNLOAD
Link

The target area in this tutorial is Campi Flegrei, Italy, covered by the frame ID of 124D_04854_171313. The list of the available data can be found at http://gws-access.ceda.ac.uk/public/nceo_geohazards/LiCSAR_products/124/124D_04854_171313/.

Once you identified the frame ID to process and confirmed that the sufficient interferograms are available on the frame, it is recommended making a processing directory and moving to the directory, e.g.,

```
mkdir 124D_04854_171313
cd 124D_04854_171313
```

3.2. Step 01: Download GeoTIFF from COMET-LiCS Web

This step downloads GeoTIFF files of unw (unwrapped interferogram) and cc (coherence) for the specified frame ID from COMET-LiCS web.

To download the GeoTIFF files for the frame ID of 124D_04854_171313 from 9 September 2016 to 8 May 2018, run the Step 01 by:

```
LiCSBAS01_get_geotiff.py -f 124D_04854_171313 -s 20160909 -e 20180508
```

The option of -f is not necessary if the name of your working directory includes the frame ID because it can be automatically determined from the name of the directory.

3.3. Step 02: Convert (and Downsample)

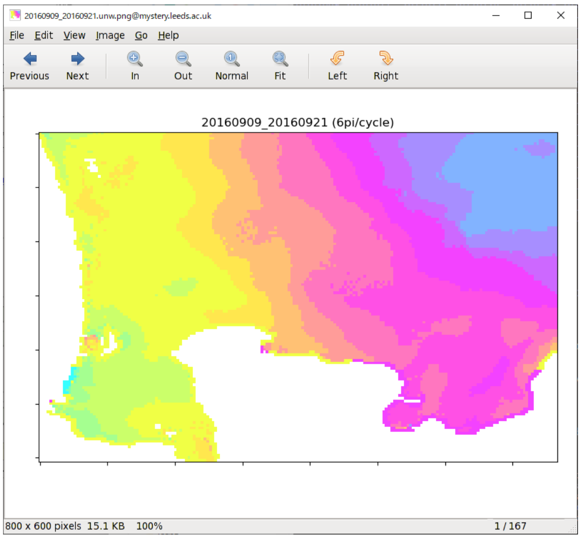
This script converts GeoTIFF files to single-precision floating-point format without header for further time series analysis, and also downsamples (multilooks) data if specified by -n option.

Run the Step 02 by:

```
LiCSBAS02_ml_prep.py -i GEOC -n 1
```

You can display the png images of the unwrapped phases by:

```
eog GEOCml1clip/*/*.png &
```


20160909_20160921.unw.png@mystery.leeds.ac.uk
File
Edit
View
Image
Go
Help
Previous
Next
In
Out
Normal
Fit
Left
Right
20160909_20160921 (6pi/cycle)
800 x 600 pixels 15.1 KB 100%
1 / 167

3.4. Step 03: GACOS Correction (Optional)

This step applies a tropospheric correction to unwrapped data using external GACOS data. GACOS data must be prepared beforehand by requesting on the GACOS web (http://ceg-research.ncl.ac.uk/v2/gacos/). This step is optional, but highly recommended.

3.5. Step 04: Mask Unwrapped Data (Optional)

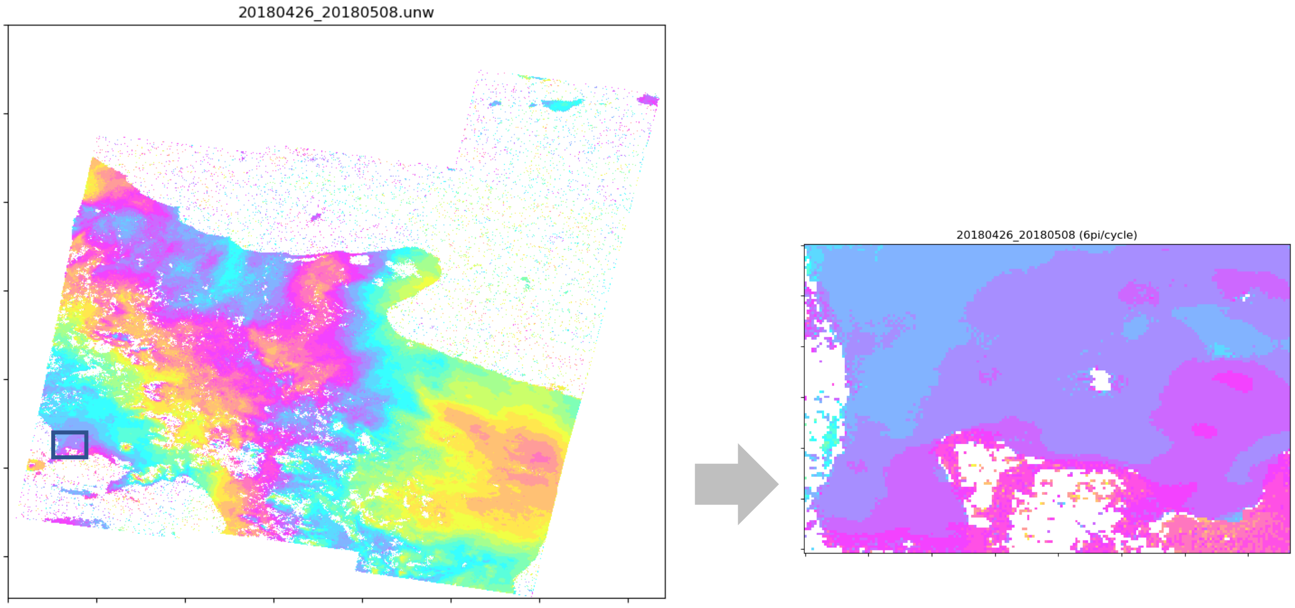
This step masks some parts of the unwrapped data. The masking is effective when the unwrapped data include areas which contains many unwrapping errors and are not of interest. This step is optional.

3.6. Step 05: Clip Unwrapped Data (Optional)

This step clips a specified rectangular area of interest from the unw and cc data. The clipping can make the data size smaller and processing faster, and improve the result of Step 12 (loop closure).

To clip the area around Campi Flegrei (14.03°E-14.22°E, 40.78°N-40.90°N), run:

```
LiCSBAS05op_clip_unw.py -i GEOCml1 -o GEOCml1clip -g 14.03/14.22/40.78/40.90
```



3.7. Step 11: Quality Check

This step checks the quality of each unwrapped data and coherence and identifies bad data to be removed from further processing.

Run the Step 11 by:

```
LiCSBAS11_check_unw.py -d GEOCml1clip
```

You can display the bad (decorrelated) unwrapped data to be removed:

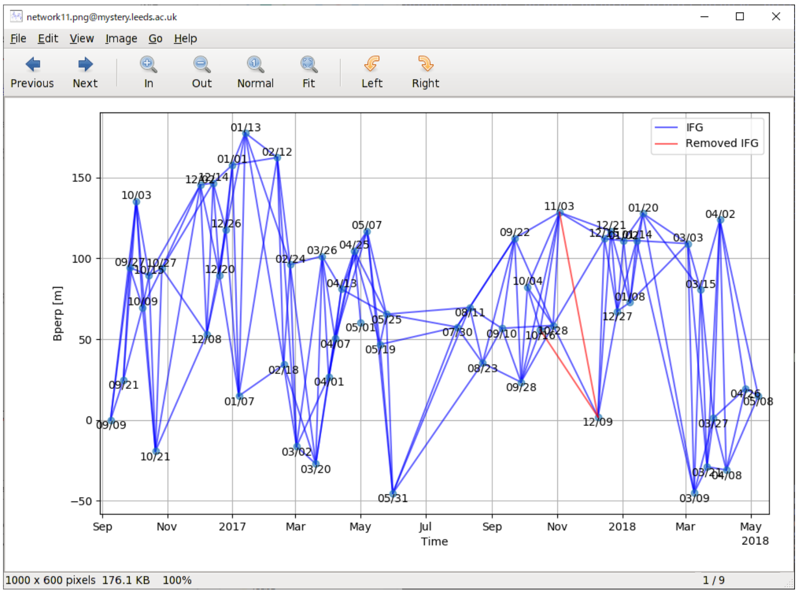
```
eog TS_GEOCml1clip/11bad_ifg_ras/*.png &
```

the good data to be used in the next step:

```
eog TS_GEOCml1clip/11ifg_ras/*.png &
```

and the small baseline network:

```
eog TS_GEOCml1clip/network/network11.png &
```



Coverage rates for each unwrapped data and average of coherence are listed in `TS_GEOCml1clip/info/11ifg_stats.txt` and can be shown by a text editor (e.g., vi, emacs, gedit):

```
gedit TS_GEOCml1clip/info/11ifg_stats.txt &
```

If you are not satisfied with the result, run `LiCSBAS11_check_unw.py` again with different parameters of -u and/or -c options.

3.8. Step 12: Loop Closure

This step computes loop phases from sets of three interferograms and identifies bad data with significant unwrapping errors to be removed from further processing.

Suppose that we have three images and three unwrapped phases, a loop phase (closure phase, phase triplet) can be calculated by

$$\Phi_{123} = \phi_{12} + \phi_{23} - \phi_{13}$$

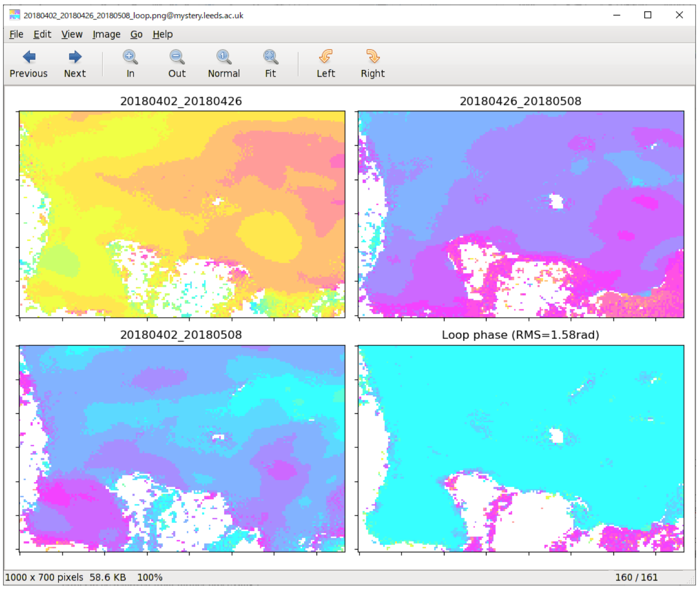
If there are no unwrapping errors in all three interferograms, the loop phase should be almost zero (not exact zero because of the multilooking effect). On the other hand, if one (or more) of the interferograms contains unwrapping errors, the loop phase becomes nearly ±2π×*n*.

Run the Step 12 by:

```
LiCSBAS12_loop_closure.py -d GEOCml1clip
```

Check the loop phases:

```
eog TS_GEOCml1clip/12loop/loop_png/*.png &
eog TS_GEOCml1clip/12loop/bad_loop_cand_png/*.png &
```



Non-zero (i.e., not light blue) areas in the right bottom image mean that one (or more) of the three interferograms contains unwrapping errors there. Luckily in this case no interferograms have significant unwrapping errors and are removed at this step.

You can display the small baseline network to be used in the next step:

```
eog TS_GEOCml1clip/network/network12.png &
```

If you are not satisfied with the result, run `LiCSBAS12_loop_closure.py` again with a different parameter of -l option, get back to the Step 11 and re-run with different parameters, or remove unwanted interferograms from GEOCml1clip and re-run from the Step 11.

3.9. Step 13: Small Baseline Inversion

This step inverts the small baseline interferograms to get the time series of the displacement and velocity using the NSBAS (López-Quiroz et al., 2009; Doin et al., 2011) approach which can deal with gaps in the network with temporal constraint under an assumption of linear displacement.

Run the Step 13 by:

```
LiCSBAS13_sb_inv.py -d GEOCml1clip
```

Now you can display the time series of the displacement and velocity by the following command, but note that this is not the final result.

```
LiCSBAS_plot_ts.py -i TS_GEOCml1clip/cum.h5 &
```

3.10. Step 14: Calculate STD of Velocity by Bootstrap

This step calculates the standard deviation (STD) of the velocity by the bootstrapping method.

```
LiCSBAS14_vel_std.py -t TS_GEOCml1clip
```

3.11. Step 15: Mask Time Series

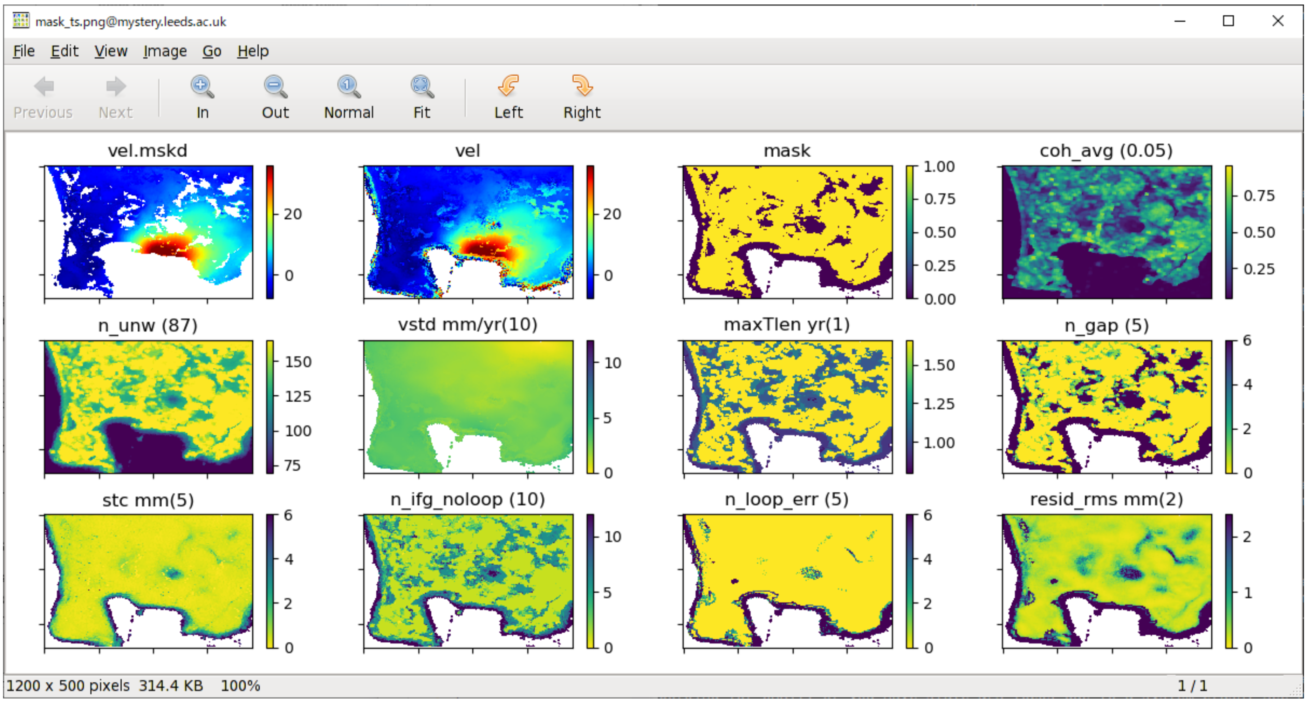
This step creates a mask for the time series and velocity of the displacement using several noise indices derived at previous steps. If any of the values of the noise indices for a pixel is larger/smaller than a specified threshold, the pixel is masked.

Run the Step 15 by:

```
LiCSBAS15_mask_ts.py -t TS_GEOCml1clip
```

Check the result of masking:

```
eog TS_GEOCml1clip/mask_ts.png &
```



Noise index	Option	Meaning
coh avg	-c	Average of coherence
n unw	-u	Number of used unwrapped data
vstd	-v	Standard deviation of velocity (mm/yr)
maxTlen	-T	Max time length of the connected network (year)
n gap	-g	Number of gaps in the network
stc	-s	Spatio-temporal consistency (mm) (Hanssen et al., 2008)
n ifg noloop	-i	Number of interferograms with no loops
n loop err	-l	Number of unclosed loops
resid	-r	RMS of residuals in SB inversion (mm)

The masked/unmasked velocity and the mask are shown at the top raw, and the other images are the noise indices. Numbers shown in the parentheses next to the titles of each noise index are the used threshold. To change the thresholds, e.g., run the following command and check the difference.

```
mv TS_GEOCml1clip/mask_ts.png TS_GEOCml1clip/mask_ts_bak.png
LiCSBAS15_mask_ts.py -t TS_GEOCml1clip -g 5
eog TS_GEOCml1clip/mask_ts*.png &
```

You can play with the masking with different thresholds to get the preferable mask.

3.12. Step 16: Filter (& Deramp) Time Series

This step applies the spatial-temporal filter (i.e., high-pass in time and low-pass in space with the gaussian kernel, same as StaMPS) to the time series of the displacement. Deramping can also be applied using -r option.

Run the Step 16 by:

```
LiCSBAS16_filt_ts.py -t TS_GEOCml1clip
```

Check the effect of the filtering:

```
eog TS_GEOCml1clip/16filt_cum/*.png &
```

To change the length of the spatial/temporal filter from default (2 km/average interval×3 (this case 0.087 yr)) to, e.g., 0.5 km/0.5 yr, run the following commands and compare the results. The smaller/larger value for the spatial/temporal filter has a stronger effect on the filtering (but higher risk to remove the true displacement as well).

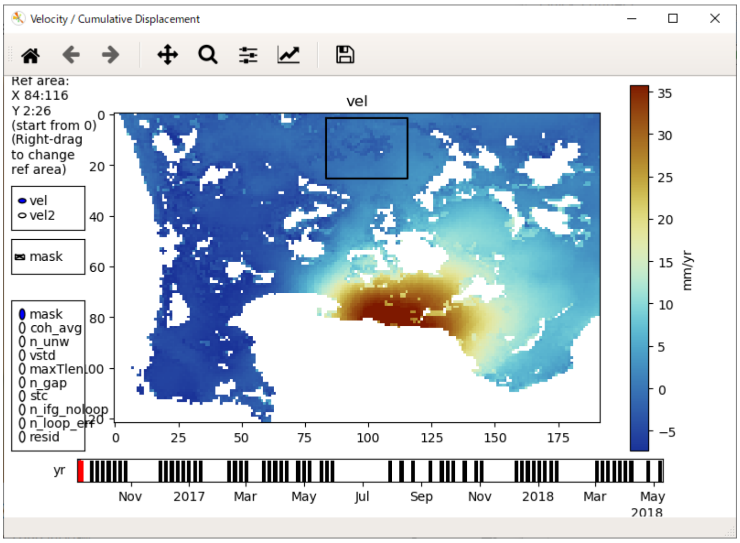
```
mv TS_GEOCml1clip/16filt_cum TS_GEOCml1clip/16filt_cum_bak
LiCSBAS16_filt_ts.py -t TS_GEOCml1clip -s 0.5 -y 0.5
eog TS_GEOCml1clip/16filt_cum_bak/*.png &
eog TS_GEOCml1clip/16filt_cum/*.png &
```

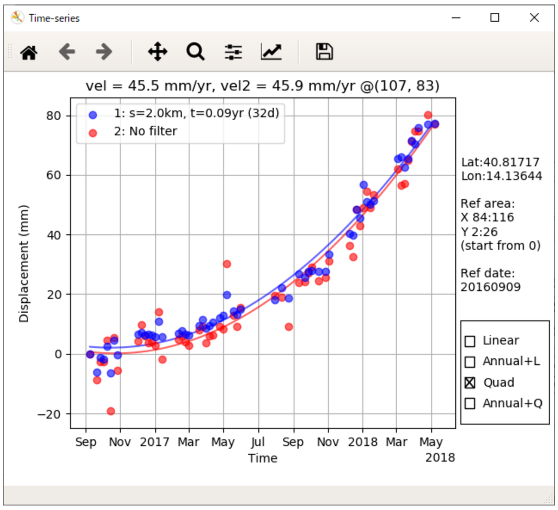
You can play with the filtering with different filter length to get the preferable results.

4. Display results

The derived time series and velocity can be interactively viewed by:

```
LiCSBAS_plot_ts.py -i TS_GEOCml1clip/cum_filt.h5 &
```





☐ Linear
☐ Annual+L
☒ Quad
☐ Annual+Q

5. Export results

GeoTIFF files can be created by `LiCSBAS_flt2geotiff.py`, e.g., for the masked velocity:

```
LiCSBAS_flt2geotiff.py        -i        TS_GEOCml1clip/results/vel.mskd        -p
    TS_GEOCml1clip/info/EQA.dem_par
```

`TS_GEOCml1clip/results/vel.mskd.geotiff` is created and can be easily displayed on a GIS software (e.g., QGIS).

The time series of the displacement at a specified point can be exported as a text file, e.g.,

```
LiCSBAS_cum2tstxt.py -p 107/83 -i TS_GEOCml1clip/cum_filt.h5
```

creates `ts_107_83.txt` and it can be opened by a text editor (e.g., vi, emacs, gedit).

For other functions and tools, see https://github.com/yumorishita/LiCSBAS/wiki/4_other_tools.

References:

Doin, M.-P., Lodge, F., Guillaso, S., Jolivet, R., Lasserre, C., Ducret, G., et al. (2011). Presentation of the small baseline NSBAS processing chain on a case example: the Etna deformation monitoring from 2003 to 2010 using Envisat data. In *Proceedings of "Fringe 2011 Workshop."* Frascati, Italy.

Hanssen, R. F., van Leijen, F. J., van Zwieten, G. J., Bremmer, C., Dortland, S., & Kleuskens, M. (2008). Validation of existing processing chains in TerraFirma stage 2. Product validation : Validation in the Amsterdam and Alkmaar area Draft version 3. (https://raw.githubusercontent.com/wiki/yumorishita/LiCSBAS/documents/Hanssen_2008.pdf)

López-Quiroz, P., Doin, M. P., Tupin, F., Briole, P., & Nicolas, J. M. (2009). Time series analysis of Mexico City subsidence constrained by radar interferometry. *Journal of Applied Geophysics*, *69*(1), 1–15. https://doi.org/10.1016/j.jappgeo.2009.02.006

Morishita, Y., Lazecky, M., Wright, T.J., Weiss, J.R., Elliott, J.R., & Hooper, A.(2020). LiCSBAS: An Open-Source InSAR Time Series Analysis Package Integrated with the LiCSAR Automated Sentinel-1 InSAR Processor. *Remote Sens.* 12, 424, https://doi.org/10.3390/RS12030424.