Automatic Time Series Generation Using HyP3 InSAR Products

ASF Engineering

Installation Instructions **2**
- Installing Necessary Common Packages 2
- Installing ASF Software 2
- Setting Environment Variables 2
- Accessing Necessary Earthdata Applications 3

Operation Instructions **4**
- Operation in hyp mode 4
- Operation in custom mode 4
- Operation in aria mode 5
- Running TRAIN and GIAnT 5
- Creating Output Files 6
- List of Command Line Options 7

Examples **8**

Abstract

This document is a guide to users of the ASF InSAR stack processing software. It includes installation and operation instructions as well as examples of code usage.

Installation Instructions

Installing Necessary Common Packages

In order to operate the ASF InSAR stack processing software, the following prerequisite packages need to be installed: numpy, gdal 2.0, scipy, h5py, matplotlib, netcdf4, configobj, configparser, and GIAnT. On a Ubuntu 16 base installation, these packages can be installed as follows:

```
$  sudo apt-get install python-numpy
$  sudo add-apt-repository -y ppa:ubuntugis/ubuntugis-unstable
$  sudo apt update
$  sudo apt upgrade
$  sudo apt install gdal-bin python-gdal
$  sudo apt-get install python-scipy
$  sudo apt-get install python-h5py
$  sudo apt-get install python-matplotlib
$  sudo apt-get install python-netcdf4
$  sudo apt-get install python-configobj
$  sudo apt-get install python-configparser
```

Note that installation instructions for GIAnT are not included in this document. Rather, please refer to http://earthdef.caltech.edu/projects/giant/wiki for details of installing GIAnT.

Installing ASF Software

Next, get the ASF software from github. Assuming you have a github account and that you have your SSH keys setup appropriately, these commands will install the ASF stack software in /usr/local:

```
$  cd /usr/local
$  sudo git clone https://github.com/asfadmin/hyp3-lib.git
$  sudo git clone https://github.com/asfadmin/hyp3-giant.git
$  sudo git clone https://github.com/asfadmin/hyp3-TRAIN.git
$  sudo pip install asf-hyp3
```

Setting Environment Variables

Set the necessary environment variables by adding these lines to your ~/.bashrc file:

```
# Set HyP3 Environment Variables
export PYTHONPATH=/usr/local/hyp3-lib/src
export PYTHONPATH=$PYTHONPATH:/usr/local/hyp3-TRAIN/src

export PATH=/usr/local/hyp3-TRAIN/src:$PATH
export PATH=/usr/local/hyp3-lib/src:$PATH
export PATH=/usr/local/hyp3-giant/src:$PATH
```

If you have not already done so, you'll also need to set the these environment variables in order to access the GIAnT software:

```
# GIAnT Environment Variables
export PYTHONPATH=$PYTHONPATH:/usr/local/GIAnT
export PATH=$PATH:/usr/local/GIAnT/SCR
```

Accessing Necessary Earthdata Applications

In order to download the MERRA2 weather model data, you must have a NASA Earthdata account. If you do not have an Earthdata Login, you can create one at https://urs.earthdata.nasa.gov/users/new.

Once you have an Earthdata login, you must create a .netrc file in your home directory. The file has this form:

```
machine urs.earthdata.nasa.gov
    login username
    password password
```

Where, *username* and *password* are your Earthdata login information.

Finally, you must approve two applications through Earthdata. The first, "NASA GESDISC DATA ARCHIVE", allows access to the MERRA2 weather data from the Goddard Earth Sciences Data and Information Services Center (GES DISC). The other, "Alaska Satellite Facility HyP3 API" provides access to your HyP3 subscriptions via the ASF HyP3 API:

1) Login to https://urs.earthdata.nasa.gov/home
2) Go to "Applications" → "Authorized Apps"
3) Go to "Approve More Applications" at the bottom of the page
4) Approve the application NASA GESDISC DATA ARCHIVE
5) Approve the application Alaska Satellite Facility HyP3 API

Having gone through these steps, you'll be able to automatically download HyP3 subscriptions using the stack processing software. And, the MERRA2 weather model data will be automatically downloaded when using TRAIN weather model corrections.

Operation Instructions

In order to create time series from GIAnT using ASF HyP3 INSAR products, ASF provides the python command line tool procS1StackGIANT.py.

```
usage: procS1StackGIANT.py [-h] [-a APIKEY] [-d DESC] [-f FILTER] [-g]
                           [-i INPUT] [-l] [-m MIN MAX] [-n] [-p PATH]
                           [-r X Y] [-s HEADING] [-t] [-u UTC] [-v NVALID]
                           [-z] [-e | -w]
                           {hyp,custom,aria} output
```

The tool operates on a HyP3 subscription name, a set of Hyp3 INSAR zip files, a directory of directories of unzipped HyP3 INSAR zip files, a user supplied custom list of INSAR phase files, or ARIA Beta products as found in ASF's VERTEX data search engine.

Operation in *hyp* mode

If the type of input files is specified as *hyp*, then the inputs are assumed to be HyP3 INSAR directories. If the --zip option is also selected, inputs are assumed to be the original HyP3 zip files. If an input subscription name is also supplied (*--input*) then these zip files will be downloaded through the HyP3 API. If no input subscription name is specified with the *hyp* option, then it is assumed that the zip files are stored in the directory pointed to by *--path*. If no path is specified, the code will use the current working directory. Whenever *--input* is used, the user must be authenticated. The default is to get authentication credentials from the file ~/.netrc. However, this behavior is overridden when the *--apikey* option is invoked - instead, the authentication credentials will be read as username and api-key from the file specified. Finally, when the *--group* option is selected with the type specified as *hyp*, the files will be sorted into directories based upon time groupings and processed by group. The alternative would be to process the entire stack as if it overlapped.

All other required input values can be read from the ASF supplied metadata and geotiff files. However, the user may override these values by using the optional command line switches *--utc* and *--heading*. A descriptor file will be automatically generated and used to drive the rest of the run.

Operation in *custom* mode

If on the other hand, a user wants to operate on a their own set of INSAR phase files, then the *custom* command line argument should be selected. In this case, the *--desc* (descriptor file), the *--utc* (utctime), and the *--heading* (spacecraft heading) switches all become required in order to garner sufficient metadata to process the phase files through GIAnT.

The descriptor file describing a total of N INSAR pairs is of the form:

<mdate1> <sdate1> <pFile1> <cFile1> <baseline1>
<mdate2> <sdate2> <pFile2> <cFile2> <baseline2>
<mdate3> <sdate3> <pFile3> <cFile3> <baseline3>
...
<mdateN> <sdateN> <pFileN> <cFileN> <baselineN>

Where,

mdate is master date (length 8 - YYYYMMDD or 15 YYYYMMDDTHHMMSS),
sdate is the slave date (8 or 15 characters),
pFile is the unwrapped phase geocoded geotiff,
cFile is the coherence geocoded geotiff, and,
baseline is the perpendicular baseline for this pair

Operation in *aria* mode

A third mode of operation for the command line tool is *aria*. In this mode, it is assumed that the input files are the unwrapped phase and coherence files created by ARIA as archived in the ASF Vertex system. As such, the individual files have been created by the ISCE software and are named as follows:

```
filt_topophase.unw.geo
filt_topophase.unw.geo.hdr
filt_topophase.unw.geo.vrt
filt_topophase.unw.geo.xml
phsig.cor.geo
phsig.cor.geo.hdr
phsig.cor.geo.vrt
phsig.cor.geo.xml
```

Moreover, these files are assumed to be stored in the sub-directory named merged under the directory named after the zip file, i.e. there must exist a directory named after the zip file and a sub-directory named *merged* that contains the above mentioned filt_topophase.unw and phsig.cor files. Note that some of the older zip files unzip their contents into the current working directory and not into appropriately named directories and *merged* subdirectories. In this case, the code will automatically create the proper directories for the procedure to run as long as the inputs given are the original zip files downloaded from ASF. Once the input files are identified by the software, processing proceeds as with other modes.

Running TRAIN and GIAnT

Once the descriptor file is parsed, images are re-projected to the geographic coordinate system (EPSG:4326) if necessary. The stack is then cut to a common overlap. If the --train option is selected, the TRAIN atmospheric correction model will next be run on the phase data. In the final preprocessing step, the files are all converted to raw format suitable for input into GIAnT.

Following image pre-processing and atmospheric corrections, preparations for running GIAnT are undertaken. The interferogram list, ifg.list, is created along with the required example.rsc file. The prepbasxml.py, prepdataxml.py, and userfn.py files are also all created automatically. Note that several command line parameters can be used to manipulate the GIAnT run. The *--rxy* switch will specify a non-deforming point in your image stack to be used as the center of the reference region. By default, the code allows GIAnTto select this point. However, better results may be obtained by specifying this point in a known region of non-deformation. The *--nsbas* option selects between running SBAS and NSBAS inversions, with the default being SBAS. The *--nvalid* switch sets the fraction of images from the stack

that must be coherent in order for a pixel to be included in the final output when running an NSBAS inversion. FInally, *--filter* sets the filter length in years.

After all pre-processing is completed and preparations are made, the GIAnT software is called. Programs prepdataxml.py, PrepIgramStack.py, and prepbasxml.py are run to set the stage for time series inversion. Normally, either SBASInvert.py or NSBASInvert.py are called to perform the actual time series calculations - in this case, the *--raw* switch will create a raw time series set of geotiffs and an animated gif of the same. If, however, the *--error* switch is used, then either SBASxval.py or NSBASxval.py will be called and a time series and animated gif of the error estimate will be created instead.

Creating Output Files

Results from the GIAnT run are next converted into an annotated animated GIF file. The color scale on the animations is by default a min/max scaling. This can be changed using the *-minmax* switch. The GIF along with the actual h5 file output from GIAnT are transferred to the output directory "PRODUCT_{output}". In addition, the h5 file is converted into individual geocoded geotiff files which are similarly placed in this product directory. If the *--raw* switch is used, then two sets of geotiffs and two animations will be created - one for the raw time series and one for the GIAnT processed time series. Similarly, If the *--error* switch is used, then two sets of geotiffs and two animations will also be created - one for the error estimate time series and one for the GIAnT processed time series. The other output from the stack processing code is the input descriptor file for the run. If a HyP3 subscription name was passed to the code there will be a directory named "hyp3-products" containing the zip files downloaded from the HyP3 API and another directory named "hyp3-products-unzipped" containing the contents of the zip files.

Upon exiting, procS1StackGiant.py will delete all intermediate files created during the run unless the *--leave* switch is specified on the command line.

List of Command Line Options

Option		Type	Description
--help	-h	boolean	Gives long list of help
--apikey	-a	string	Name of api-key configuration file to use for logging into the HyP3 API. Default is to login using .netrc credentials.
--desc	-d	string	Name of interferogram stack descriptor file
--filter	-f	float	GIAnT time filter length in years
--group	-g	boolean	Turns on group sorting by time. WIth this option, the input files will first be sorted into time classes - all images within 8 seconds of each other will be grouped into the same class.
--input	-i	string	Name of the HyP3 subscription to use for the input files. If this option is selected, then all files in the HyP3 subscription named will be downloaded through the HyP3 API.
--leave	-l	boolean	Leave all intermediate files in place. The default action is to delete them upon program completion
--minmax	-m	float x 2	Set the minimum and maximum color scale extent. The default is to use a min/max scaling.
--nsbas	-n	boolean	Switches between SBAS (default) and NSBAS inversion programs
--path	-p	string	Path to the input files, either absolute or relative
--rxy	-r	integer x 2	Set the center of the reference region used by GIAnT. The default is to let GIAnT select its own reference region
--heading	-s	float	Spacecraft heading at time of acquisition
--train	-t	boolean	Turns on the use of TRAIN atmospheric correction model
--utc	-u	float	UTC time of the image stack
--nvalid	-v	float	Fraction of images that must be coherent for a pixel to be included in the NSBAS inversion. The default is 0.8, or 80%
--error	-e	boolean	Flag to create the error estimate times series geotiffs and gif files. Mutually exclusive with *--raw* switch.
--raw	-w	boolean	Flag to create the raw data time series geotiffs and gif files. Mutually exclusive with *--error* switch.
--zip	-z	boolean	SIgnifies that input files are HyP3 zip files instead of already unzipped directories

Examples

```
$ procS1StackGIANT.py -g --input "InSAR Gamma Cedar City" -m -100 100 --raw hyp gamma_cc_auto
```

This command will download the HyP3 subscription named “InSAR Gamma Cedar City”. It will unzip all of the files, group them by time, and process them using GIAnT. Animations and geotiffs of both the GIAnT time series and the raw time series will be created. The color scale on the animations is set from -100 mm to +100 mm. The output directories and files will contain the string *gamma_cc_auto*

```
$ procS1StackGIANT.py -g -i "InSAR Gamma Cedar City" -m -100 100 -w -t hyp gamma_cc_auto_train
```

This is the same command as above, except it will run the TRAIN atmosphere model on the data. Note that it will first check with the HyP3 API to see if any new files exist for this subscription. If so, they will be downloaded and unzipped. Otherwise, no downloading or unzipping will take place.

```
$ procS1StackGIANT.py -g -i "InSAR Gamma Cedar City" -m -100 100 -e -t hyp gamma_cc_train_err
```

This is nearly the same command as above, except it will run the GIAnT script SBASxval.py on the time series to create the inverted time series as well as an error estimate. Note that this option (-e) cannot be used with the raw option (-w).

```
$ procS1StackGIANT.py -f 0.0028 -p hyp3-products-unzipped --train -g --raw hyp iran_auto_train
```

Run the stack creation program on the HyP3 unzipped directories stored in *hyp3-products-unzipped*. Set the GIAnT time filter length to 0.0028 years (1 day). Run the TRAIN atmospheric correction model and create both the GIAnT processed time series and the raw time series. The output name is set to *iran_auto_train*.