
pyroSAR Documentation

Release 0.8

John Truckenbrodt, Felix Cremer, Ismail Baris, Rhys Kidd

Apr 09, 2019

Contents

1 Drivers	1
2 SNAP Processing	15
3 GAMMA Processing	17
3.1 SRTM tools	23
3.2 GAMMA Command API	26
4 Sentinel-1 Tools	125
5 Auxiliary Data Tools	129
6 Datacube Tools	131
7 Ancillary Functions	133
8 Changelog	137
8.1 0.6 / 2018-11-20	137
8.2 0.7 / 2019-01-03	138
8.3 0.8 / 2019-02-11	139
9 Indices and tables	141
Python Module Index	143

Drivers

This is the core module of package pyroSAR. It contains the drivers for the different SAR image formats and offers functionality for retrieving metadata, unpacking images, downloading ancillary files like DEMs and Orbit State Vector files as well as archiving scenes in a database. The *ID* class and its subclasses allow easy and standardized access to the metadata of images from different SAR sensors.

classes

<i>ID</i>	Abstract class for SAR meta data handlers
<i>CEOS_PSR</i>	Handler class for ALOS-PALSAR data in CEOS format
<i>CEOS_ERS</i>	Handler class for ERS data in CEOS format
<i>ESA</i>	Handler class for SAR data in ESA format (Envisat ASAR, ERS-1/2)
<i>SAFE</i>	Handler class for Sentinel-1 data
<i>TSX</i>	Handler class for TerraSAR-X and TanDEM-X data
<i>Archive</i>	Utility for storing SAR image metadata in a spatialite database

functions

<i>identify</i>	identify a SAR scene and return the appropriate meta-data handler object
<i>identify_many</i>	wrapper function for returning metadata handlers of all valid scenes in a list, similar to function <i>identify()</i> .
<i>filter_processed</i>	Filter a list of pyroSAR objects to those that have not yet been processed and stored in the defined directory.
<i>findfiles</i>	find files in a scene archive, which match a pattern
<i>getFileObj</i>	Load a file in a SAR scene archive into a readable file object.

Continued on next page

Table 2 – continued from previous page

<code>parse_date</code>	this function gathers known time formats provided in the different SAR products and converts them to a common standard of the form YYYYMMDDTHHMMSS
-------------------------	--

class `pyroSAR.drivers.Archive` (*dbfile*, *custom_fields=None*)

Bases: `object`

Utility for storing SAR image metadata in a spatialite database

Parameters

- **dbfile** (*str*) – the database file. This file might either point to an existing database or will be created otherwise.
- **custom_fields** (*dict*) – a dictionary containing additional non-standard database column names and data types; the names must be attributes of the SAR scenes to be inserted (i.e. `id.attr`) or keys in their meta attribute (i.e. `id.meta['attr']`)

Examples

select all Sentinel-1 A/B scenes stored in the database, which

- overlap with a test site
- were acquired in Ground-Range-Detected (GRD) Interferometric Wide Swath (IW) mode before 2018
- contain a VV polarization image
- have not been processed to directory *outdir* before

```
>>> from pyroSAR import Archive
>>> from pyroSAR.spatial import Vector
>>> archive = Archive('/path/to/dbfile.db')
>>> site = Vector('/path/to/site.shp')
>>> outdir = '/path/to/processed/results'
>>> maxdate = '20171231T235959'
>>> selection_proc = archive.select(vectorobject=site,
>>>                                processdir=outdir,
>>>                                maxdate=maxdate,
>>>                                sensor=('S1A', 'S1B'),
>>>                                product='GRD',
>>>                                acquisition_mode='IW',
>>>                                vv=1)
>>> archive.close()
```

Alternatively, the *with* statement can be used. In this case to just check whether one particular scene is already registered in the database:

```
>>> from pyroSAR import identify, Archive
>>> scene = identify('S1A_IW_SLC__1SDV_20150330T170734_20150330T170801_005264_
→ 006A6C_DA69.zip')
>>> with Archive('/path/to/dbfile.db') as archive:
>>>     print(archive.is_registered(scene))
```

close()

close the database connection

export2shp (*shp*)

export the database to a shapefile

Parameters **shp** (*str*) – the name of the shapefile to be written

filter_scenelist (*scenelist*)

Filter a list of scenes by file names already registered in the database.

Parameters **scenelist** (*list* of *str* or *pyroSAR.drivers.ID*) – the scenes to be filtered

Returns the file names of the scenes whose basename is not yet registered in the database

Return type *list*

get_colnames ()

Return the names of the database table.

Returns the column names of the data table

Return type *list*

get_tablenames ()

Return the names of all tables in the database

Returns the table names

Return type *list*

get_unique_directories ()

Get a list of directories containing registered scenes

Returns the directory names

Return type *list*

import_outdated (*dbfile*, *verbose=False*)

import an older data base in csv format

Parameters

- **dbfile** (*str*) – the file name of the old data base
- **verbose** (*bool*) – should status information and a progress bar be printed into the console?

insert (*scene_in*, *verbose=False*, *test=False*)

Insert one or many scenes into the database

Parameters

- **scene_in** (*str* or *list*) – a SAR scene or a list of scenes to be inserted
- **verbose** (*bool*) – should status information and a progress bar be printed into the console?
- **test** (*bool*) – should the insertion only be tested or directly be committed to the database?

is_registered (*scene*)

Simple check if a scene is already registered in the database.

Parameters **scene** (*str*) – the SAR scene

Returns is the scene already registered?

Return type *bool*

move (*scenelist*, *directory*)

Move a list of files while keeping the database entries up to date. If a scene is registered in the database (in either the data or duplicates table), the scene entry is directly changed to the new location.

Parameters

- **scenelist** (*list*) – the file locations
- **directory** (*str*) – a folder to which the files are moved

select (*vectorobject=None*, *mindate=None*, *maxdate=None*, *processdir=None*, *recursive=False*, *polarizations=None*, *verbose=False*, ***args*)
select scenes from the database

Parameters

- **vectorobject** (*Vector*) – a geometry with which the scenes need to overlap
- **mindate** (*str*) – the minimum acquisition date in format YYYYmmddTHHMMSS
- **maxdate** (*str*) – the maximum acquisition date in format YYYYmmddTHHMMSS
- **processdir** (*str*) – a directory to be scanned for already processed scenes; the selected scenes will be filtered to those that have not yet been processed
- **recursive** (*bool*) – should also the subdirectories of the processdir be scanned?
- **polarizations** (*list*) – a list of polarization strings, e.g. ['HH', 'VV']
- **verbose** (*bool*) – print details about the selection including the SQL query?
- ****args** – any further arguments (columns), which are registered in the database. See [`get_colnames\(\)`](#)

Returns the file names pointing to the selected scenes

Return type *list*

select_duplicates (*outname_base=None*, *scene=None*)

Select scenes from the duplicates table. In case both *outname_base* and *scene* are set to None all scenes in the table are returned, otherwise only those that match the attributes *outname_base* and *scene* if they are not None.

Parameters

- **outname_base** (*str*) – the basename of the scene
- **scene** (*str*) – the scene name

Returns the selected scene(s)

Return type *list*

size

get the number of scenes registered in the database

Returns the number of scenes in (1) the main table and (2) the duplicates table

Return type *tuple*

class `pyroSAR.drivers.CEOS_ERS` (*scene*)

Bases: `pyroSAR.drivers.ID`

Handler class for ERS data in CEOS format

Sensors:

- ERS1

- ERS2

Reference: ER-IS-EPO-GS-5902-3: Annex C. ERS SAR.SLC/SLC-I. CCT and EXABYTE (ESA 1998)

getCorners ()

derive the corner coordinates from a SAR scene

Returns dictionary with keys *xmin*, *xmax*, *ymin* and *ymax*

Return type `dict`

scanMetadata ()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns the derived attributes

Return type `dict`

unpack (*directory*, *overwrite=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?

class `pyroSAR.drivers.CEOS_PSR` (*scene*)

Bases: `pyroSAR.drivers.ID`

Handler class for ALOS-PALSAR data in CEOS format

Sensors:

- PSR1
- PSR2

PALSAR-1:

Reference: NEB-070062B: ALOS/PALSAR Level 1.1/1.5 product Format description (JAXA 2009)

Products / processing levels:

- 1.0
- 1.1
- 1.5

Acquisition modes:

- AB: [SP][HWDPC]
- **A: supplemental remarks of the sensor type:**
 - S: Wide observation mode
 - P: all other modes
- **B: observation mode**
 - H: Fine mode
 - W: ScanSAR mode
 - D: Direct downlink mode

- P: Polarimetry mode
- C: Calibration mode

PALSAR-2:

Reference: ALOS-2/PALSAR-2 Level 1.1/1.5/2.1/3.1 CEOS SAR Product Format Description ([JAXA 2014](#)).

Products / processing levels:

- 1.0
- 1.1
- 1.5

Acquisition modes:

- SBS: Spotlight mode
- UBS: Ultra-fine mode Single polarization
- UBD: Ultra-fine mode Dual polarization
- HBS: High-sensitive mode Single polarization
- HBD: High-sensitive mode Dual polarization
- HBQ: High-sensitive mode Full (Quad.) polarimetry
- FBS: Fine mode Single polarization
- FBD: Fine mode Dual polarization
- FBQ: Fine mode Full (Quad.) polarimetry
- WBS: Scan SAR nominal [14MHz] mode Single polarization
- WBD: Scan SAR nominal [14MHz] mode Dual polarization
- WWS: Scan SAR nominal [28MHz] mode Single polarization
- WWD: Scan SAR nominal [28MHz] mode Dual polarization
- VBS: Scan SAR wide mode Single polarization
- VBD: Scan SAR wide mode Dual polarization

`getCorners()`

derive the corner coordinates from a SAR scene

Returns dictionary with keys *xmin*, *xmax*, *ymin* and *ymax*

Return type `dict`

`led_filename`

`scanMetadata()`

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns the derived attributes

Return type `dict`

`unpack(directory, overwrite=False)`

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?

```
class pyroSAR.drivers.ESA(scene)
```

Bases: `pyroSAR.drivers.ID`

Handler class for SAR data in ESA format (Envisat ASAR, ERS-1/2)

Sensors:

- ASAR
- ERS1
- ERS2

```
getCorners()
```

derive the corner coordinates from a SAR scene

Returns dictionary with keys *xmin*, *xmax*, *ymin* and *ymax*

Return type `dict`

```
scanMetadata()
```

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns the derived attributes

Return type `dict`

```
unpack(directory, overwrite=False)
```

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?

```
class pyroSAR.drivers.ID(metadict)
```

Bases: `object`

Abstract class for SAR meta data handlers

```
bbox(outname=None, overwrite=True)
```

get the bounding box of a scene either as a vector object or written to a shapefile

Parameters

- **outname** (*str*) – the name of the shapefile to be written
- **overwrite** (*bool*) – overwrite an existing shapefile?

Returns the vector object if *outname* is None, None otherwise

Return type `Vector` or `None`

```
compression
```

check whether a scene is compressed into an tarfile or zipfile or not at all

Returns either 'zip', 'tar' or None

Return type `str` or `None`

examine (*include_folders=False*)

check whether any items in the SAR scene structure (i.e. files/folders) match the regular expression pattern defined by the class. On success the item is registered in the object as attribute *file*.

Parameters **include_folders** (*bool*) – also match folder (or just files)?

Raises `IOError`

export2dict ()

Return the uuid and the metadata that is defined in self.locals as a dictionary

export2sqlite (*dbfile*)

Export relevant metadata to a sqlite database

Parameters **dbfile** (*str*) – the database file

findfiles (*pattern, include_folders=False*)

find files in the scene archive, which match a pattern; see *findfiles()*

Parameters

- **pattern** (*str*) – the regular expression to match
- **include_folders** (*bool*) – also match folders (or just files)?

Returns the matched file names

Return type `list`

gdalinfo ()

read metadata directly from the GDAL SAR image drivers

Parameters **scene** (*str*) – an archive containing a SAR scene

Returns the metadata attributes

Return type `dict`

getCorners ()

derive the corner coordinates from a SAR scene

Returns dictionary with keys *xmin*, *xmax*, *ymin* and *ymax*

Return type `dict`

getFileObj (*filename*)

Load a file into a readable file object.

Parameters **filename** (*str*) – the name of a file in the scene archive, easiest to get with method *findfiles()*

Returns a file pointer object

Return type `BytesIO`

getGammaImages (*directory=None*)

list all files processed by GAMMA

Parameters **directory** (*str*) – the directory to be scanned; if left empty the object attribute *gammadir* is scanned

Returns the file names of the images processed by GAMMA

Return type `list`

Raises `IOError`

getHGT ()

get the names of all SRTM HGT tiles overlapping with the SAR scene

Returns names of the SRTM HGT tiles

Return type `list`

is_processed (*outdir*, *recursive=False*)

check whether a scene has already been processed and stored in the defined output directory (and subdirectories if scanned recursively)

Parameters **outdir** (*str*) – the directory to be checked

Returns does an image matching the scene pattern exist?

Return type `bool`

outname_base (*extensions=None*)

parse a string containing basic information about the scene in standardized format. Currently this id contains the sensor (4 digits), acquisition mode (4 digits), orbit (1 digit) and acquisition start time (15 digits), e.g. *SIA_IW_A_20150523T122350*

Parameters **extensions** (*list of str*) – the names of additional parameters to append to the basename, e.g. ['orbitNumber_rel']

Returns a standardized name unique to the scene

Return type `str`

static parse_date (*x*)

this function gathers known time formats provided in the different SAR products and converts them to a common standard of the form *YYYYMMDDTHHMMSS*.

Parameters **x** (*str*) – the time stamp

Returns the converted time stamp in format *YYYYmmddTHHMMSS*

Return type `str`

scanMetadata ()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns the derived attributes

Return type `dict`

summary ()

print the set of standardized scene metadata attributes

unpack (*directory*, *overwrite=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?

class `pyroSAR.drivers.SAFE` (*scene*)

Bases: `pyroSAR.drivers.ID`

Handler class for Sentinel-1 data

Sensors:

- S1A
- S1B

References:

- S1-RS-MDA-52-7443 Sentinel-1 IPF Auxiliary Product Specification
- MPC-0243 Masking “No-value” Pixels on GRD Products generated by the Sentinel-1 ESA IPF

getCorners()

derive the corner coordinates from a SAR scene

Returns dictionary with keys *xmin*, *xmax*, *ymin* and *ymax*

Return type `dict`

getOSV(outdir, osvType='POE')

download Orbit State Vector files for the scene

Parameters

- **outdir** (*str*) – the directory of OSV files; subdirectories POEORB and RESORB are created automatically
- **osvType** (`{ 'POE', 'RES' }`) – the type of orbit file either ‘POE’, ‘RES’ or a list of both

See also:

`pyroSAR.S1.auxil.OSV()`

removeGRDBorderNoise()

mask out Sentinel-1 image border noise. See `removeGRDBorderNoise()`

scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns the derived attributes

Return type `dict`

unpack(directory, overwrite=False)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?

class pyroSAR.drivers.TSX(scene)

Bases: `pyroSAR.drivers.ID`

Handler class for TerraSAR-X and TanDEM-X data

Sensors:

- TSX1
- TDX1

References:

- TX-GS-DD-3302 TerraSAR-X Basic Product Specification Document

- TX-GS-DD-3303 TerraSAR-X Experimental Product Description
- TD-GS-PS-3028 TanDEM-X Experimental Product Description
- TerraSAR-X Image Product Guide (Airbus Defence and Space)

Acquisition modes:

- ST: Staring Spotlight
- HS: High Resolution SpotLight
- HS300: High Resolution SpotLight 300 MHz
- SL: SpotLight
- SM: StripMap
- SC: ScanSAR
- WS: Wide ScanSAR

Polarisation modes:

- Single (S): all acquisition modes
- Dual (D): High Resolution SpotLight (HS), SpotLight (SL) and StripMap (SM)
- Twin (T): StripMap (SM) (experimental)
- Quad (Q): StripMap (SM) (experimental)

Products:

- SSC: Single Look Slant Range Complex
- MGD: Multi Look Ground Range Detected
- GEC: Geocoded Ellipsoid Corrected
- EEC: Enhanced Ellipsoid Corrected

getCorners ()

derive the corner coordinates from a SAR scene

Returns dictionary with keys *xmin*, *xmax*, *ymin* and *ymax*

Return type `dict`

scanMetadata ()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns the derived attributes

Return type `dict`

unpack (directory, overwrite=False)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?

`pyroSAR.drivers.filter_processed(scenelist, outdir, recursive=False)`

Filter a list of pyroSAR objects to those that have not yet been processed and stored in the defined directory. The search for processed scenes is either done in the directory only or recursively into subdirectories. The scenes must have been processed with pyroSAR in order to follow the right naming scheme.

Parameters

- **scenelist** (*list*) – a list of pyroSAR objects
- **outdir** (*str*) – the processing directory
- **recursive** (*bool*) – scan *outdir* recursively into subdirectories?

Returns a list of those scenes, which have not been processed yet

Return type *list*

`pyroSAR.drivers.findfiles(scene, pattern, include_folders=False)`

find files in a scene archive, which match a pattern

Parameters

- **scene** (*str*) – the SAR scene to be scanned, can be a directory, a zip or tar.gz archive
- **pattern** (*str*) – the regular expression to match
- **include_folders** (*bool*) – also match folders (or just files)?

Returns the matched file names

Return type *list*

`pyroSAR.drivers.getFileObj(scene, filename)`

Load a file in a SAR scene archive into a readable file object.

Parameters

- **scene** (*str*) – the scene archive. Can be either a directory or a compressed archive of type *zip* or *tar.gz*.
- **filename** (*str*) – the name of a file in the scene archive, easiest to get with method *findfiles()*

Returns a file object

Return type *BytesIO*

`pyroSAR.drivers.identify(scene)`

identify a SAR scene and return the appropriate metadata handler object

Parameters **scene** (*str*) – a file or directory name

Returns a pyroSAR metadata handler

Return type a subclass object of *ID*

`pyroSAR.drivers.identify_many(scenes)`

wrapper function for returning metadata handlers of all valid scenes in a list, similar to function *identify()*. Prints a progressbar.

Parameters **scenes** (*list*) – the file names of the scenes to be identified

Returns a list of pyroSAR metadata handlers

Return type *list*

`pyroSAR.drivers.parse_date(x)`

this function gathers known time formats provided in the different SAR products and converts them to a common standard of the form YYYYMMDDTHHMMSS

Parameters *x* (*str* or *datetime*) – the time stamp to be converted

Returns the converted time stamp in format YYYYmmddTHHMMSS

Return type *str*

SNAP Processing

```
pyroSAR.snap.util.geocode(infile, outdir, t_srs=4326, tr=20, polarizations='all', shapefile=None, scaling='dB', geocoding_type='Range-Doppler', removeS1BoderNoise=True, offset=None, externalDEMFile=None, externalDEMNoDataValue=None, externalDEMAApplyEGM=True, basename_extensions=None, test=False, export_extra=None)
```

wrapper function for geocoding SAR images using ESA SNAP

Parameters

- **infile** (*str* or *ID*) – the SAR scene to be processed
- **outdir** (*str*) – The directory to write the final files to.
- **t_srs** (*int*, *str* or *osr.SpatialReference*) – A target geographic reference system in WKT, EPSG, PROJ4 or OPENGIS format. See function `spatialist.auxil.crsConvert()` for details. Default: 4326.
- **tr** (*int* or *float*, *optional*) – The target resolution in meters. Default is 20
- **polarizations** (*list* or {'VV', 'HH', 'VH', 'HV', 'all'}, *optional*) – The polarizations to be processed; can be a string for a single polarization e.g. 'VV' or a list of several polarizations e.g. ['VV', 'VH']. Default is 'all'.
- **shapefile** (*str* or *Vector*, *optional*) – A vector geometry for subsetting the SAR scene to a test site. Default is None.
- **scaling** ({'dB', 'db', 'linear'}, *optional*) – Should the output be in linear or decibel scaling? Default is 'dB'.
- **geocoding_type** ({'Range-Doppler', 'SAR simulation cross correlation'}, *optional*) – The type of geocoding applied; can be either 'Range-Doppler' (default) or 'SAR simulation cross correlation'
- **removeS1BoderNoise** (*bool*, *optional*) – Enables removal of S1 GRD border noise (default).

- **offset** (*tuple, optional*) – A tuple defining offsets for left, right, top and bottom in pixels, e.g. (100, 100, 0, 0); this variable is overridden if a shapefile is defined. Default is None.
- **externalDEMFile** (*str or None, optional*) – The absolute path to an external DEM file. Default is None.
- **externalDEMNoDataValue** (*int, float or None, optional*) – The no data value of the external DEM. If not specified (default) the function will try to read it from the specified external DEM.
- **externalDEMApplyEGM** (*bool, optional*) – Apply Earth Gravitational Model to external DEM? Default is True.
- **basename_extensions** (*list of str*) – names of additional parameters to append to the basename, e.g. ['orbitNumber_rel']
- **test** (*bool, optional*) – If set to True the workflow xml file is only written and not executed. Default is False.
- **export_extra** (*list or None*) –

a list of image file IDs to be exported to outdir. The following IDs are currently supported:

- incidenceAngleFromEllipsoid
- localIncidenceAngle
- projectedLocalIncidenceAngle

Note: If only one polarization is selected and not extra products are defined the results are directly written to GeoTiff. Otherwise the results are first written to a folder containing ENVI files and then transformed to GeoTiff files (one for each polarization/extra product). If GeoTiff would directly be selected as output format for multiple polarizations then a multilayer GeoTiff is written by SNAP which is considered an unfavorable format

Examples

geocode a Sentinel-1 scene and export the local incidence angle map with it

```
>>> from pyroSAR.snap import geocode
>>> filename = 'S1A_IW_GRDH_1SDV_20141012T162337_20141012T162402_002799_00326F_
↳ 8197.zip'
>>> geocode(infile=filename, outdir='outdir', tr=20, scaling='dB', export_extra=[
↳ 'localIncidenceAngle'])
```

See also:

`pyroSAR.drivers.ID`, `spatialist.vector.Vector`, `spatialist.auxil.crsConvert()`

GAMMA Processing

<i>convert2gamma</i>	general function for converting SAR images to GAMMA format
<i>correctOSV</i>	correct GAMMA parameter files with orbit state vector information from dedicated OSV files
<i>geocode</i>	general function for geocoding SAR images with GAMMA
<i>ISPPar</i>	Reader for ISP parameter files of the GAMMA software package
<i>multilook</i>	multilooking of SLC and MLI images
<i>ovs</i>	compute DEM oversampling factors for a target resolution in meters
<i>par2hdr</i>	Create an ENVI HDR file from a Gamma PAR file
<i>process</i>	wrapper function to execute GAMMA commands via module <code>subprocess</code>
<i>S1_deburst</i>	Debursting of Sentinel-1 SLC imagery in GAMMA
<i>UTM</i>	convert a gamma parameter file corner coordinate from EQA to UTM

```
pyroSAR.gamma.geocode(scene, dem, tempdir, outdir, targetres, scaling='linear', func_geoback=2,
                        func_interp=2, nodata=(0, -99), sarSimCC=False, osvdir=None, al-
                        low_RES_OSV=False, cleanup=True, normalization_method=2, ex-
                        port_extra=None)
```

general function for geocoding SAR images with GAMMA

Parameters

- **scene** (*str* or *ID*) – the SAR scene to be processed
- **dem** (*str*) – the reference DEM in GAMMA format
- **tempdir** (*str*) – a temporary directory for writing intermediate files
- **outdir** (*str*) – the directory for the final GeoTiff output files

- **targetres** (*int*) – the target resolution in meters
- **scaling** (*{'linear', 'db'} or list*) – the value scaling of the backscatter values; either 'linear', 'db' or a list of both, i.e. ['linear', 'db']
- **func_geoback** (*{0, 1, 2, 3}*) –
backward geocoding interpolation mode (see GAMMA command geocode_back)
 - 0: nearest-neighbor
 - 1: bicubic spline
 - 2: bicubic-log spline, interpolates log(data)
 - 3: bicubic-sqrt spline, interpolates sqrt(data)

NOTE: bicubic-log spline and bicubic-sqrt spline modes should only be used with non-negative data!
- **func_interp** (*{0, 1, 2, 3}*) –
output lookup table values in regions of layover, shadow, or DEM gaps (see GAMMA command gc_map)
 - 0: set to (0., 0.)
 - 1: linear interpolation across these regions
 - 2: actual value
 - 3: nn-thinned
- **nodata** (*tuple*) – the nodata values for the output files; defined as a tuple with two values, the first for linear, the second for logarithmic scaling
- **sarSimCC** (*bool*) – perform geocoding with SAR simulation cross correlation? If False, geocoding is performed with the Range-Doppler approach using orbit state vectors
- **osvdir** (*str*) – a directory for Orbit State Vector files; this is currently only used by for Sentinel-1 where two subdirectories POEORB and RESORB are created; if set to None, a subdirectory OSV is created in the directory of the unpacked scene.
- **allow_RES_OSV** (*bool*) – also allow the less accurate RES orbit files to be used? Otherwise the function will raise an error if no POE file exists
- **cleanup** (*bool*) – should all files written to the temporary directory during function execution be deleted after processing?
- **normalization_method** (*{1, 2}*) –
the topographic normalization approach to be used
 - 1: first geocoding, then terrain flattening
 - 2: first terrain flattening, then geocoding; see [Small 2011](#)
- **export_extra** (*list or None*) –
a list of image file IDs to be exported to outdir
 - format is GeoTiff if the file is geocoded and ENVI otherwise. Non-geocoded images can be converted via Gamma command data2tiff yet the output was found impossible to read with GIS software
 - scaling of SAR image products is applied as defined by parameter *scaling*
 - see Notes for ID options

Note:

intermediate output files

DEM products are named <scene identifier>_<ID>, e.g. *SIA_IW_A_20141012T162337_inc_geo*

SAR products will additionally contain the polarization, e.g. *SIA_IW_A_20141012T162337_VV_grd_mli*

IDs in brackets are only written if selected by *export_extra*

- images in range-Doppler geometry
 - **grd**: the ground range detected SAR intensity image
 - **grd_mli**: the multi-looked grd image with approached target resolution
 - specific to normalization method 2:
 - * **pix_ellip_sigma0**: ellipsoid-based pixel area
 - * **pix_area_sigma0**: actual illuminated area as obtained from integrating DEM-facets (command `pixel_area`)
 - * **pix_fine**: refined pixel area normalization factor (`pix_ellip_sigma0 / pix_area_sigma0`)
 - * **grd_mli_pan**: the pixel area normalized MLI (`grd_mli * pix_fine`)
 - images in map geometry
 - **dem_seg_geo**: dem subsetting to the extent of the intersect between input DEM and SAR image
 - (**u_geo**): zenith angle of surface normal vector n (angle between z and n)
 - (**v_geo**): orientation angle of n (between x and projection of n in xy plane)
 - **inc_geo**: local incidence angle (between surface normal and look vector)
 - (**psi_geo**): projection angle (between surface normal and image plane normal)
 - **pix_geo**: pixel area normalization factor (command `gc_map`)
 - **ls_map_geo**: layover and shadow map (in map projection)
 - (**sim_sar_geo**): simulated SAR backscatter image
 - additional files
 - **lut_init**: initial geocoding lookup table
 - files specific to SAR simulation cross-correlation geocoding
 - **lut_fine**: refined geocoding lookup table
 - **diffpar**: ISP offset/interferogram parameter file
 - **offs**: offset estimates (fcomplex)
 - **coffs**: culled range and azimuth offset estimates (fcomplex)
 - **coffsets**: culled offset estimates and cross correlation values (text format)
 - **ccp**: cross-correlation of each patch (0.0->1.0) (float)
-

Examples

geocode a Sentinel-1 scene and export the local incidence angle map with it

```
>>> from pyroSAR.gamma import geocode
>>> filename = 'S1A_IW_GRDH_1SDV_20141012T162337_20141012T162402_002799_00326F_
↳ 8197.zip'
>>> geocode(scene=filename, dem='demfile', outdir='outdir', targetres=20, scaling=
↳ 'db', export_extra=['inc_geo'])
```

`pyroSAR.gamma.convert2gamma` (*id*, *directory*, *S1_noiseremoval*=*True*, *logpath*=*None*, *outdir*=*None*, *shellscript*=*None*)

general function for converting SAR images to GAMMA format

Parameters

- **id** (*ID*) – an SAR scene object of type `pyroSAR.ID` or any subclass
- **directory** (*str*) – the output directory for the converted images
- **S1_noiseremoval** (*bool*) – only Sentinel-1: should noise removal be applied to the image?
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`class pyroSAR.gamma.ISPPar` (*filename*)

Bases: `object`

Reader for ISP parameter files of the GAMMA software package

This class allows to read all information from files in GAMMA’s parameter file format. Each key-value pair is parsed and added as attribute. For instance if the parameter file contains the pair ‘sensor: TSX-1’ an attribute named ‘sensor’ with the value ‘TSX-1’ will be available.

The values are converted to native Python types, while unit identifiers like ‘dB’ or ‘Hz’ are removed. Please see the GAMMA reference manual for further information on the actual file format.

Parameters **filename** (*str*) – the Gamma parameter file

Examples

```
>>> from pyroSAR.gamma import ISPPar
>>> with ISPPar('S1A_IW___A_20141115T181801_VH_grd.par') as par:
...     print(par) # print an overview of all available metadata
...     print(par.keys) # print all parameter names
...     for key, value in par.envidict().items():
...         print('{0}: {1}'.format(key, value)) # print the ENVI HDR compliant_
↳ metadata
```

keys

the names of all parameters

Type `list`

envidict (*nodata*=*None*)

export relevant metadata to a ENVI HDR file compliant format

Parameters `nodata` (*str* or *None*) – a no data value to write to the HDR file via attribute ‘data ignore value’

Returns a dictionary containing attributes translated to ENVI HDR naming

Return type `dict`

`pyroSAR.gamma.process` (*cmd*, *outdir=None*, *logfile=None*, *logpath=None*, *inlist=None*, *void=True*, *shellscript=None*)

wrapper function to execute GAMMA commands via module `subprocess`

Parameters

- `cmd` (*list*) – the command line arguments
- `outdir` (*str*) – the directory to execute the command in
- `logfile` (*str*) – a file to write the command log to; overrides parameter `logpath`
- `logpath` (*str*) – a directory to write logfiles to; the file will be named {GAMMA command}.log, e.g. `gc_map.log`; is overridden by parameter `logfile`
- `inlist` (*list*) – a list of values, which is passed as interactive inputs via stdin
- `void` (*bool*) – return the stdout and stderr messages?
- `shellscript` (*str*) – a file to write the Gamma commands to in shell format

Returns the stdout and stderr messages if `void` is `False`, otherwise `None`

Return type tuple of *str* or *None*

`pyroSAR.gamma.ovs` (*parfile*, *targetres*)

compute DEM oversampling factors for a target resolution in meters

Parameters

- `parfile` (*str*) – a GAMMA DEM parameter file
- `targetres` (*int* or *float*) – the target resolution in meters

Returns the oversampling factors for latitude and longitude

Return type tuple of *float*

`pyroSAR.gamma.S1_deburst` (*burst1*, *burst2*, *burst3*, *name_out*, *rlks=5*, *azlks=1*, *replace=False*, *logpath=None*, *outdir=None*, *shellscript=None*)

Debursting of Sentinel-1 SLC imagery in GAMMA

The procedure consists of two steps. First antenna pattern deramping and then mosaicing of the single deramped bursts. For mosaicing, the burst boundaries are calculated from the number of looks in range (*rlks*) and azimuth (*azlks*), in this case 5 range looks and 1 azimuth looks. Alternately 10 range looks and 2 azimuth looks could be used.

Parameters

- `burst1` (*str*) – burst image 1
- `burst2` (*str*) – burst image 2
- `burst3` (*str*) – burst image 3
- `name_out` (*str*) – the name of the output file
- `rlks` (*int*) – the number of looks in range
- `azlks` (*int*) – the number of looks in azimuth

- **replace** (*bool*) – replace the burst images by the new file? If True, the three burst images will be deleted.
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.correctOSV(id, osvdir=None, osvType='POE', logpath=None, outdir=None, shellscript=None)`

correct GAMMA parameter files with orbit state vector information from dedicated OSV files

Parameters

- **id** (*ID*) – the scene to be corrected
- **osvdir** (*str*) – the directory of OSV files; subdirectories POEORB and RESORB are created automatically
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.multilook(infile, outfile, targetres, logpath=None, outdir=None, shellscript=None)`
multilooking of SLC and MLI images

if the image is in slant range the ground range resolution is computed by dividing the range pixel spacing by the sine of the incidence angle

the looks in range and azimuth are chosen to approximate the target resolution by rounding the ratio between target resolution and ground range/azimuth pixel spacing to the nearest integer

an ENVI HDR parameter file is automatically written for better handling in other software

Parameters

- **infile** (*str*) – a SAR image in GAMMA format with a parameter file of name <infile>.par
- **outfile** (*str*) – the name of the output GAMMA file
- **targetres** (*int*) – the target resolution in ground range
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.par2hdr(parfile, hdrfile, modifications=None, nodata=None)`

Create an ENVI HDR file from a Gamma PAR file

Parameters

- **parfile** (*str*) – the Gamma parfile
- **hdrfile** (*str*) – the ENVI HDR file
- **modifications** (*dict* or *None*) – a dictionary containing value deviations to write to the HDR file
- **nodata** (*str* or *None*) – a no data value to write to the HDR file via attribute 'data ignore value'

Examples

```
>>> from pyroSAR.gamma.auxil import par2hdr
>>> par2hdr('dem_seg.par', 'inc.hdr')
# write a HDR file for byte data based on a parfile of float data
>>> par2hdr('dem_seg.par', 'ls_map.hdr', modifications={'data_type': 1})
```

See also:

`spatialist.envi.HDRObject`, `spatialist.envi.hdr()`

class `pyroSAR.gamma.UTM` (*parfile*)

Bases: `object`

convert a gamma parameter file corner coordinate from EQA to UTM

Parameters `parfile` (*str*) – the Gamma parameter file to read the coordinate from

Example

```
>>> from pyroSAR.gamma import UTM
>>> print(UTM('gamma.par').zone)
```

3.1 SRTM tools

A collection of functions to handle digital elevation models in Gamma

<code>dem_autocreate</code>	automatically create a DEM in Gamma format for a defined spatial geometry
<code>dempar</code>	create GAMMA parameter text files for DEM files
<code>fill</code>	interpolate missing values in the SRTM DEM (value - 32768)
<code>hgt</code>	concatenate hgt file names overlapping with multiple SAR scenes
<code>hgt_collect</code>	automatic downloading and unpacking of srtm tiles
<code>makeSRTM</code>	Create a DEM in Gamma format from SRTM tiles
<code>mosaic</code>	mosaicing of multiple DEMs
<code>swap</code>	byte swapping from small to big endian (as required by GAMMA)

`pyroSAR.gamma.srtm.dem_autocreate` (*geometry*, *demType*, *outfile*, *buffer*=0.01, *logpath*=None, *username*=None, *password*=None)

automatically create a DEM in Gamma format for a defined spatial geometry

the following steps will be performed:

- collect all tiles overlapping with the geometry
 - if they don't yet exist locally they will automatically be downloaded
 - the tiles will be downloaded into the SNAP auxdata directory structure, e.g.

\$HOME/.snap/auxdata/dem/SRTM 3Sec

- create a mosaic GeoTiff of the same spatial extent as the input geometry plus a defined buffer using `gdalwarp`
- subtract the EGM96-WGS84 Geoid-Ellipsoid difference and convert the result to Gamma format using Gamma command `srtm2dem`
 - this correction is not done for TanDEM-X data, which contains ellipsoid heights; see [here](#)

Parameters

- **geometry** (*spatialist.vector.Vector*) – a vector geometry delimiting the output DEM size; CRS must be WGS84 LatLon (EPSG 4326)
- **demType** (*str*) – the type of DEM to be used; see `dem_autoload()` for options
- **outfile** (*str*) – the name of the final DEM file
- **buffer** (*float*) – a buffer in degrees to create around the geometry
- **logpath** (*str*) – a directory to write Gamma logfiles to
- **username** (*str* or *None*) – (optional) the user name for services requiring registration; see `dem_autoload()`
- **password** (*str* or *None*) – (optional) the password for the registration account

`pyroSAR.gamma.srtm.dempar (dem, logpath=None)`

create GAMMA parameter text files for DEM files

currently only EQA and UTM projections with WGS84 ellipsoid are supported

Parameters

- **dem** (*str*) – the name of the DEM
- **logpath** (*str*) – a directory to write logfiles to

`pyroSAR.gamma.srtm.fill (dem, dem_out, logpath=None, replace=False)`

interpolate missing values in the SRTM DEM (value -32768)

Parameters

- **dem** (*str*) – the input DEM to be filled
- **dem_out** (*str*) – the name of the filled DEM
- **logpath** (*str*) – a directory to write logfiles to
- **replace** (*bool*) – delete *dem* once finished?

`pyroSAR.gamma.srtm.hgt (parfiles)`

concatenate hgt file names overlapping with multiple SAR scenes

- this list is read for corner coordinates of which the next integer lower left latitude and longitude is computed
- hgt files are supplied in 1 degree equiangular format named e.g. N16W094.hgt (with pattern [NS][0-9]{2}[EW][0-9]{3}.hgt
- For north and east hemisphere the respective absolute latitude and longitude values are smaller than the lower left coordinate of the SAR image
- west and south coordinates are negative and hence the nearest lower left integer absolute value is going to be larger

Parameters **parfiles** (*list of str or pyroSAR.ID*) – a list of Gamma parameter files or pyroSAR ID objects

Returns the names of hgt files overlapping with the supplied parameter files/objects

Return type *list*

`pyroSAR.gamma.srtm.hgt_collect` (*parfiles, outdir, demdir=None, arcsec=3*)
automatic downloading and unpacking of srtm tiles

Parameters

- **parfiles** (*list of str or pyroSAR.ID*) – a list of Gamma parameter files or pyroSAR ID objects
- **outdir** (*str*) – a target directory to download the tiles to
- **demdir** (*str or None*) – an additional directory already containing hgt tiles
- **arcsec** (*{1, 3}*) – the spatial resolution to be used

Returns the names of all local hgt tiles overlapping with the parfiles

Return type *list*

`pyroSAR.gamma.srtm.makeSRTM` (*scenes, srtmdir, outname*)
Create a DEM in Gamma format from SRTM tiles

- coordinates are read to determine the required DEM extent and select the necessary hgt tiles
- mosaics SRTM DEM tiles, converts them to Gamma format and subtracts offset to WGS84 ellipsoid

intended for SRTM products downloaded from:

- USGS: <https://gdex.cr.usgs.gov/gdex/>
- CGIAR: <http://srtm.csi.cgiar.org>

Parameters

- **scenes** (*list of str or pyroSAR.ID*) – a list of Gamma parameter files or pyroSAR ID objects to read the DEM extent from
- **srtmdir** (*str*) – a directory containing the SRTM hgt tiles
- **outname** (*str*) – the name of the final DEM file

`pyroSAR.gamma.srtm.mosaic` (*demlist, outname, byteorder=1, gammapar=True*)
mosaicing of multiple DEMs

Parameters

- **demlist** (*list*) – a list of DEM names to be mosaiced
- **outname** (*str*) – the name of the final mosaic file
- **byteorder** (*{0, 1}*) – the byte order of the mosaic
 - 0: small endian
 - 1: big endian
- **gammapar** (*bool*) – create a Gamma parameter file for the mosaic?

`pyroSAR.gamma.srtm.swap` (*data, outname*)
byte swapping from small to big endian (as required by GAMMA)

Parameters

- **data** (*str*) – the DEM file to be swapped
- **outname** (*str*) – the name of the file to write

3.2 GAMMA Command API

This is an attempt to make it easier to execute Gamma commands by offering automatically parsed Python functions. Thus, instead of executing the command via shell:

```
offset_fit offs ccp off.par coffs - 0.15 3 0 > offset_fit.log
```

one can wrap it in a Python script:

```
import os
from pyroSAR.gamma.api import isp

workdir = '/data/gamma_workdir'

parameters = {'offs': os.path.join(workdir, 'offs'),
              'ccp': os.path.join(workdir, 'ccp'),
              'OFF_par': os.path.join(workdir, 'off.par'),
              'coffs': os.path.join(workdir, 'coffs'),
              'thres': 0.15,
              'npoly': 3,
              'interact_flag': 0,
              'logpath': workdir}

isp.offset_fit(**parameters)
```

A file *offset_fit.log* containing the output of the command is written in both cases. Any parameters, which should not be written and need to be set to - in the shell can be omitted in the Python call since all optional parameters of the functions are already defined with '-' as a default. The documentation can be called like with any Python function:

```
from pyroSAR.gamma.api import isp
help(isp.offset_fit)
```

3.2.1 Parser Documentation

`pyroSAR.gamma.parser.autoparse()`

automatic parsing of Gamma commands. This function will detect the Gamma installation via environment variable *GAMMA_HOME*, detect all available modules (e.g. ISP, DIFF) and parse all of the module's commands via function *parse_module()*. A new Python module will be created called *gammaparse*, which is stored under *\$HOME/pyrosar*. Upon importing the *pyroSAR.gamma* submodule, this function is run automatically and module *gammaparse* is imported as *api*.

Examples

```
>>> from pyroSAR.gamma.api import diff
>>> print('create_dem_par' in dir(diff))
True
```

```
pyroSAR.gamma.parser.parse_command(command, indent='')
```

Parse the help text of a Gamma command to a Python function including a docstring. The docstring is in rst format and can thus be parsed by e.g. sphinx. This function is not intended to be used by itself, but rather within function `parse_module()`.

Parameters

- **command** (*str*) – the name of the gamma command
- **indent** (*str*) – the Python function indentation string; default: four spaces

Returns the full Python function text

Return type *str*

```
pyroSAR.gamma.parser.parse_module(bindir, outfile)
```

parse all Gamma commands of a module to functions and save them to a Python script.

Parameters

- **bindir** (*str*) – the *bin* directory of a module containing the commands
- **outfile** (*str*) – the name of the Python file to write

Examples

```
>>> import os
>>> from pyroSAR.gamma.parser import parse_module
>>> outname = os.path.join(os.environ['HOME'], 'isp.py')
>>> parse_module('/cluster/GAMMA_SOFTWARE-20161207/ISP/bin', outname)
```

3.2.2 API Demo

This is a demonstration of an output script as generated automatically by function `parse_module()` for the Gamma module *ISP*. Within each function, the command name and all parameters are passed to function `process()`, which converts all input to str and then calls the command via the `subprocess` module.

```
pyroSAR.gamma.parser_demo.ASAR_LO_phase_drift(SLC1_par, SLC2_par, OFF_par,
                                                ph_drift, logpath=None, outdir=None,
                                                shellscript=None)
```

Calculate interferometric phase correction due to drift of the ASAR local oscillator

Copyright 2015, Gamma Remote Sensing, v1.1 3-Dec-2015 clw

Parameters

- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **ph_drift** – (output) interferometric phase correction due to drift of the ASAR LO (radians)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ASAR_XCA (ASA_XCA, antenna, swath='-', pol='-', logpath=None,
                                     outdir=None, shellscript=None)
```

Interpretation of ASAR external calibration data file (ASA_XCA)

Copyright 2006, Gamma Remote Sensing, v1.1 7-June-2006 awi/uw/clw

Parameters

- **ASA_XCA** – (input) ASAR external calibration data file (binary)
- **antenna** –
(output) 1-way antenna gain pattern file or ‘-’ (if not provided) or ‘all’ to generate all ASAR antenna diagrams
- **swath** – ASAR swath (IS1,IS2,... IS7;SS1,SS2,... SS5)
- **pol** – polarization (HH,VV,HV,VH)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.DELFT_vec2 (SLC_par, DELFT_dir, nstate='-', interval='-',
                                       ODR='-', logpath=None, outdir=None,
                                       shellscript=None)
```

Extract and interpolate DELFT ERS-1, ERS-2, and ENVISAT state vectors

Copyright 2012, Gamma Remote Sensing, v2.6 clw 24-Oct-2012

Parameters

- **SLC_par** – (input) ISP image parameter file
- **DELFT_dir** –
directory containing Delft orbit arclist and ODR files for ERS-1, ERS-2 or ENVISAT

– NOTE: enter . for current directory
- **nstate** – number of state vectors to generate (enter - for default (>= 15))
- **interval** – time interval between state vectors in the ISP image parameter file (s) (default: 10.0)
- **ODR** – ODR file to use (include path) rather than ODR file determined from the Delft orbit arclist
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.DORIS_vec (SLC_par, DOR, nstate='-', logpath=None, outdir=None,
                                       shellscript=None)
```

Extract ENVISAT DORIS state vectors and write to an ISP image parameter file

Copyright 2008, Gamma Remote Sensing, v1.4 11-Jun-2008 clw

Parameters

- **SLC_par** – (input/output)ISP SLC/MLI image parameter file
- **DOR** – (input) ASAR DORIS data file (DOR_VOR_AXVF)
- **nstate** – number of state vectors to extract (enter - for default: 11)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ERS_ASF_SLC (orbit, device, logpath=None, outdir=None,  
                                         shellscript=None)
```

Parameters

- **orbit** – orbit identifier (example: orbit number)
- **device** – EXABYTE tape device (no rewind); example: /dev/rmt/0mn
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ERS_ESA_PRI (orbit, device, logpath=None, outdir=None,  
                                         shellscript=None)
```

Parameters

- **orbit** – orbit identifier (example: orbit number)
- **device** – EXABYTE tape device (no rewind); example: /dev/rmt/0mn
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ERS_ESA_SLC (orbit, device, logpath=None, outdir=None,  
                                         shellscript=None)
```

Parameters

- **orbit** – orbit identifier (example: orbit number)
- **device** – EXABYTE tape device (no rewind); example: /dev/rmt/0mn
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.GRD_to_SR(GRD_par, MLI_par, OFF_par, in_file, out_file, rlks='',  
                                     azlks='', interp_mode='', sr_rsp='', sr_azsp='', log-  
                                     path=None, outdir=None, shellscript=None)
```

Conversion to slant range for ISP MLI and INSAR ground range data of type float

Copyright 2014, Gamma Remote Sensing, v2.0 4-Apr-2014 uw/clw

Parameters

- **GRD_par** – (input) SLC parameter file of output ground range image
- **MLI_par** –
(output) MLI ISP image parameter file for slant range image (float)
 - NOTE: delete an existing MLI parameter file to recalculate the output MLI parameters
- **OFF_par** – (input) ISP offset/interferogram parameter file of input image (enter - image in MLI geometry)
- **in_file** – (input) ground range image (float)
- **out_file** – (output) slant range image (float)
- **rlks** – multi-looking in range (prior to resampling, default=1)
- **azlks** – multi-looking in azimuth (prior to resampling, default=1)
- **interp_mode** –
interpolation mode
 - 0: nearest neighbor (default)
 - 1: spline
 - 2: spline log
- **sr_rsp** – output image slant range sample spacing (m) (default = $c/(2*\text{adc_sampling_rate})$)
- **sr_azsp** – output image azimuth sample spacing (m) (default = (input image azimuth spacing) * azlks)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.INTF_SLC(pass1, pass2, rlks, azlks, algorithm='', cc_win=''-  
                                   ', r_pos='', az_pos='', logpath=None, outdir=None,  
                                   shellscript=None)
```

Parameters

- **pass1** – pass 1 identifier (example: pass number) reference
- **pass2** – pass 2 identifier (example: pass number)
- **rlks** – number of range looks
- **azlks** – number of azimuth looks
- **algorithm** –

algorithm used to determine offsets: 1=intensity image cross correlation (default)
2=fringe visibility

- **cc_win** – window used for estimation of the correlation coefficient (default=3)
- **r_pos** – range position of center of image patch for initial offset
- **az_pos** – azimuth position of center of image patch for initial offset
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.MLI_cat (MLI_1, MLI_2, MLI1_par, MLI2_par, MLI_3, MLI3_par,
                                     logpath=None, outdir=None, shellscript=None)
```

Concatenate two MLI images using bicubic spline interpolation

Copyright 2015, Gamma Remote Sensing, v1.0 23-Jul-2015 awi

Parameters

- **MLI_1** – (input) MLI-1 image (single-look)
- **MLI_2** – (input) MLI-2 image to be appended to MLI-1
- **MLI1_par** – (input) MLI-1 ISP image parameter file
- **MLI2_par** – (input) MLI-2 ISP image parameter file
- **MLI_3** – (output) concatenated MLI image
- **MLI3_par** – (output) ISP image parameter file for concatenated image
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.MLI_copy (MLI_in, MLI_in_par, MLI_out, MLI_out_par, roff='-
', nr='-', loff='-', nl='-', logpath=None, outdir=None,
                                     shellscript=None)
```

Copy MLI data file with options for segment extraction

Copyright 2013, Gamma Remote Sensing, v4.4 10-Jan-2013 uw/clw

Parameters

- **MLI_in** – (input) multi-look intensity image (float format)
- **MLI_in_par** – (input) ISP image parameter file for input MLI
- **MLI_out** – (output) selected MLI section (float format)
- **MLI_out_par** – (output) ISP image parameter file for output MLI
- **roff** – offset to starting range sample (enter - for default: 0)
- **nr** – number of range samples (enter - for default: to end of line)
- **loff** – offset to starting line (enter - for default: 0)
- **nl** – number of lines to copy (enter - for default: to end of file)

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.OPOD_vec (SLC_par, OPOD_dir, nstate='- ', logpath=None, out-  
dir=None, shellscript=None)
```

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/OPOD_vec

Copyright 2017, Gamma Remote Sensing, v1.2 7-Dec-2017 clw/awi

Extract Sentinel state vectors from an OPOD file and write these state vectors to an SLC parameter file

The required OPOD file located in a specified directory containing either restituted or precise state vectors

Parameters

- **SLC_par** – (input/output)ISP SLC/MLI image parameter file
- **OPOD_dir** –
(input) directory containing Sentinel-1 precise or restituted OPOD orbit data file (AUX_POEORB or AUX_)
https://qc.sentinel1.eo.esa.int/aux_resorb/
- **nstate** – number of state vectors to extract (default: include 60 sec extension at the start and end of the SLC data)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ORB_filt (SLC_par_in, SLC_par_out, interval='- ', extra='- ', log-  
path=None, outdir=None, shellscript=None)
```

Filter state vectors using a least-squares polynomial model

Copyright 2017, Gamma Remote Sensing, v1.1 21-Jun-2017 clw

Parameters

- **SLC_par_in** – (input) ISP image parameter file at least 5 state vectors
- **SLC_par_out** – (output) ISP image parameter file with state vectors filtered using least-squares
- **interval** – time interval between state vectors (enter - for default: state vector time interval in SLC_par)
- **extra** – extra time for state vectors at start and end of image (sec.) (enter - for default: 5.0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ORB_prop_SLC (SLC_par, nstate='- ', interval='- ', extra='- '  
, mode='- ', logpath=None, outdir=None,  
shellscript=None)
```

Calculate state vectors using orbit propagation and interpolation

Copyright 2008, Gamma Remote Sensing, v1.8 11-Jun-2008 clw/awi

Parameters

- **SLC_par** – (input) ISP image parameter file with at least 1 state vector
- **nstate** – number of state vectors to calculate (enter - for default: nstate from image duration + extra)
- **interval** – time interval between state vectors (enter - for default: state vector time interval in SLC_par)
- **extra** – extra time for state vectors at start and end of image (sec.) (enter - for default: 30.0)
- **mode** –
 - orbit propagation mode:**
 - 0: polynomial interpolation (default, if 3 or more state vectors available)
 - 1: integration of the equations of motion (default, if less than 3 state vectors available)
 - 2: interpolate between state vectors, minimum of 3 state vectors; interpolation of the equations of motion outside of the time span of the existing state vectors
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ORMM_vec(SLC_par, ORRM, nstate='-', logpath=None, outdir=None, shellscript=None)
```

Calculate state vectors extraction from ORRM file

Copyright 2008, Gamma Remote Sensing, v1.4 15-Nov-2004 clw

Parameters

- **SLC_par** – (input/output) ISP SLC/MLI image parameter file
- **ORMM** – (input) ORRM state vector file
- **nstate** – number of state vectors (default=5, maximum=1024)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.PRC_vec(SLC_par, PRC, nstate='-', logpath=None, outdir=None, shellscript=None)
```

State vectors from ERS PRC orbit data for ISP processing clw/uw

Copyright 2008, Gamma Remote Sensing, v1.7 clw 11-Jun-2008

Parameters

- **SLC_par** – (input/output) ISP SLC/MLI image parameter file
- **PRC** – (input) PRC state vector file
- **nstate** – number of state vectors (default=5, maximum=1024)
- **logpath** (*str* or *None*) – a directory to write command logfiles to

- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.RSAT2_vec(SLC_par, RSAT2_orb, nstate='-', logpath=None, outdir=None, shellscript=None)
```

Extract Radarsat-2 state vectors from a definitive orbit file

Copyright 2010, Gamma Remote Sensing, v1.0 clw 13-May-2010

Parameters

- **SLC_par** – (input) ISP image parameter file
- **RSAT2_orb** – Radarsat-2 definitive orbit data file available from MDA. (orbit_number_def.orb)
- **nstate** – number of state vectors to extract (enter - for default: 9)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.S1_BURST_tab(SLC1_tab, SLC2_tab, BURST_tab, logpath=None, outdir=None, shellscript=None)
```

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/S1_BURST_tab

Copyright 2017, Gamma Remote Sensing, v1.0 25-May-2017 clw

Calculate Sentinel BURST_tab based on parameters extracted from SLC parameter files listed in SLC1_tab and SLC2_tab

Running SLC_copy_S1_TOPS using BURST_tab will generate SLC-2 data with matching bursts for each swath of SLC-1 and SLC-2

Parameters

- **SLC1_tab** – (input) 3 column list of the reference TOPS SLC swaths in row order IW1, IW2, IW3
- **SLC2_tab** – (input) 3 column list of TOPS SLC-2 swaths to be resampled to the geometry of the reference SLC1 in row order IW1, IW2, IW3.
- **BURST_tab** – (output) 2 column list of the first and last bursts to copy from each swath, one line for each swath
- **BURST_tab** – line entries: first_burst last_burst Note: first burst is 1
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.S1_BURST_tab_from_zipfile(zipfile_list, zipfile_ref, burst_number_table_ref='-', cleaning='-', logpath=None, outdir=None, shellscript=None)
```

Parameters

- **zipfile_list** –
(input) ASCII file containing S1 zip filename(s) of one data take indicate - to generate burst_number_table of reference TOPS SLC
- **zipfile_ref** – (input) S1 zip filename for the reference TOPS SLC
- **burst_number_table_ref** –
(input) ASCII file containing first/last burst numbers selected indicate - to use all bursts as present in the reference TOPS SLC zipfile
- **cleaning** –
flag to indicate if intermediate files are deleted (default=1: yes, 0: not deleted)
intermediate and output filenames are generated based on the zip file names
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.S1_GRD_preproc(S1_list, MLI_dir, pol, log, logpath=None, outdir=None, shellscript=None)
```

Preprocessing of Sentinel-1 TOPS GRD products, extract GRD data and generate MLI products
Copyright 2018, Gamma Remote Sensing, v1.2 19-Mar-2018 clw/cm

Parameters

- **S1_list** – (input) single column text file. Entries are directories (including path) containing Sentinel-1 TOPS GRD products
- **MLI_dir** –
directory for output SLC data files and SLC parameter files
– NOTE: output file names have the form : 20150119_hh.mli
- **pol** – SLC polarization to extract (hh,hv,vh,vv)
- **log** –
(output) S1 GRD pre-processing log file

-c	(option) apply radiometric calibration factor without noise subtraction
-n	(option) apply radiometric calibration factor with noise subtraction
-t	(option) include full timestamp YYYYMMDDtHH-MMSS in SLC and SLC_par filenames, default YYYYMMDD
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.S1_OPOD_vec(SLC_par, OPOD, nstate='', logpath=None, out-  
dir=None, shellscript=None)
```

Extract Sentinel-1 OPOD state vectors and copy into the ISP image parameter file

Copyright 2017, Gamma Remote Sensing, v1.3 09-Mar-2017 awi/clw

Parameters

- **SLC_par** – (input/output)ISP SLC/MLI image parameter file
- **OPOD** –
(input) Sentinel-1 OPOD orbit data file (AUX_POEORB or AUX_RESORB)
https://qc.sentinel1.eo.esa.int/aux_resorb/
- **nstate** – number of state vectors to extract (default: include 60 sec extension at the start and end of the SLC data)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.S1_TOPS_preproc(S1_list, SLC_dir, pol, log, logpath=None, out-  
dir=None, shellscript=None)
```

Preprocessing of Sentinel-1 TOPS SLC products, extract SLC data and generate SLC_tab

Copyright 2018, Gamma Remote Sensing, v2.2 10-Jan-2018 clw/awi

Parameters

- **S1_list** – (input) single column text file. Entries are directories (including path) containing Sentinel-1 TOPS SLC products
- **SLC_dir** –
directory for output SLC data files and SLC parameter files Note: output file names have the form : 20150119_iw1_hh.slc
- **pol** – SLC polarization to extract (hh,hv,vh,vv)
- **log** –
(output) S1 SLC pre-processing log file
- **-c** (option) apply radiometric calibration factor without noise subtraction
- **-n** (option) apply radiometric calibration factor with noise subtraction
- **-s** (option) output is SCOMPLEX format (default: FCOMPLEX)
- **-t** (option) include full timestamp YYYYMMDDtHH-MMSS in SLC and SLC_par filenames, default YYYYMMDD
- **-m MLI_dir** (option) calculate MLI images and store in MLI_dir, enter . for current directory
- **-r rlks** (option) number of MLI range looks (default: 10)

-a azlks (option) number of MLI azimuth looks (default: 2)

-b SLC_tab (option) SLC_tab filename, by
default SLC_tab_YYMMDD or
SLC_tab_YYYYMMDDtHHMMSS

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.S1_burstloc(annotation_XML, logpath=None, outdir=None,
                                       shellscript=None)
```

Print Burst information found in the Sentinel-1 annotation file

Copyright 2018, Gamma Remote Sensing, v1.1 7-Feb-2018 awi/cm

Parameters

- **annotation_XML** – (input) Sentinel-1 L1 XML annotation file
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.S1_extract_png(zipfile, logpath=None, outdir=None,
                                          shellscript=None)
```

Parameters

- **zipfile** – (input) Sentinel-1 zipfile (GRD or SLC)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.S1_import_SLC_from_zipfiles(zipfile_list,
                                                        burst_number_table_ref='-
', pol='- ', dtype='- ',
swath_flag='- ', cleaning='-
', noise_mode='- ', log-
path=None, outdir=None,
shellscript=None)
```

Parameters

- **zipfile_list** – (input) ASCII file containing S1 zip filename(s) of one data take (one per line, in correct sequence)
- **burst_number_table_ref** –
(input) ASCII file containing first/last burst numbers selected indicate - to use all bursts present in the indicated zipfiles

- **pol** –
polarization flag (default: -, other values are vh, hh, hv) indicate - to use all polarizations available in the indicated zipfiles
- **dtype** – output data type: default=0 (FCOMPLEX), 1: SCOMPLEX
- **swath_flag** – flag to select sub-swaths to read (default=0 (as listed in burst_number_table_ref, all if no
- **burst_number_table_ref** –
provided), 1,2,3 (1 sub-swath only), 4 (1&2), 5 (2&3) OPOD_dir directory with OPOD state vector files (default: .)
- **cleaning** – flag to indicate if intermediate files are deleted (default = 1 -> deleted, 0: not deleted)
- **noise_mode** –
noise mode (default=1: apply noise correction; 2: do not apply and write out noise file
resulting files: burst SLC files (per polarization, with SLC_tab, SLC, SLC_par, TOPS_par and optionally SLC.noise) (concatenated, empty bursts added where necessary) at selected polarizations
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SBI_INT(RSLC1, RSLC_par1, RSLC2, RSLC_par2, sbi, off, sbi_pwr,  
                                   par_out, norm_sq='-', rlks='-', azlks='-', iwflg='-', cflg='-',  
                                   logpath=None, outdir=None, shellscript=None)
```

Parameters

- **RSLC1** – (input) master single-look complex image (fcomplex or scomplex)
- **RSLC_par1** – (input) SLC ISP image parameter file of RSLC1
- **RSLC2** – (input) co-registered slave SLC image (fcomplex or scomplex)
- **RSLC_par2** – (input) SLC ISP image parameter file of RSLC2
- **sbi** – (output) multi-look split-beam interferogram (fcomplex)
- **off** – (output) ISP offset parameter file for multi-look split-beam interferogram (ascii)
- **sbi_pwr** – (output) multi-look reference backscatter intensity image (float)
- **par_out** – (output) SLC/MLI ISP image parameter file of sbi_pwr
- **norm_sq** – normalized squint difference parameter (default: 0.5)
- **rlks** – number of range looks in output split-beam interferogram (default: 1)
- **azlks** – number of azimuth looks in output split-beam interferogram (default: 1)
- **iwflg** –
inverse weighting flag:
 - 0: do not remove azimuth processing spectral window (default)

- 1: apply inverse of azimuth compression processing window
- **cflg** –
flag to indicate if intermediate data (e.g. filtered slc) are deleted
 - 0: intermediate data are deleted (default)
 - 1: intermediate data are NOT deleted file names for band-pass filtered SLC are generated automatically by adding the letter b / f for the backward / forward looking beam
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_adf (SLC, ref_SLC, ref_SLC_par, SLC_filt, mode='-', alpha='-',
                                     nfft_r='-', nfft_az='-', r_step='-', az_step='-',
                                     mwin_r='-', mwin_az='-', logpath=None, outdir=None,
                                     shellscript=None)
```

Adaptive filtering of SLC data based on the local PSD of a reference SLC image

Copyright 2017, Gamma Remote Sensing, v1.2 29-Sep-2017 clw

Parameters

- **SLC** – (input) SLC to be filtered (FCOMPLEX or SCOMPLEX)
- **ref_SLC** – (input) reference SLC
- **ref_SLC_par** – (input) reference SLC parameter file
- **SLC_filt** – (output) output filtered SLC using the power spectrum of the reference SLC
- **mode** –
SLC filtering mode (enter - for default):
 - 0: 1D range PSD filter
 - 1: 1D azimuth PSD filter
 - 2: 2D range PSD * azimuth PSD filter
 - 3: 2D median-filtered PSD filtering (default)
- **alpha** – exponent to apply to PSD value (enter - for default: 0.30)
- **nfft_r** – range filter FFT window size, 2**N, 16->1024, (enter - for default: 128)
- **nfft_az** – azimuth filter FFT window size, 2**N, 16->1024, (enter - for default: 128)
- **r_step** – range processing step (enter - for default: nfft_r/4)
- **az_step** – azimuth processing step (enter - for default: nfft_az/4)
- **mwin_r** – range median window size for median PSD filtering (enter - for default: 5)
- **mwin_az** – azimuth median window size for median PSD filtering (enter - for default: 5)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_burst_copy (SLC, SLC_par, TOPS_par, SLC_out,  
                                           SLC_out_par, burst_num, drflg='- ',  
                                           SLC_par2='- ', dtype='- ', logpath=None,  
                                           outdir=None, shellscript=None)
```

Copy selected burst from Sentinel-1 TOPS SLC to a file

Copyright 2017, Gamma Remote Sensing, v1.4 24-Nov-2017 awi/clw/cm

Parameters

- **SLC** – (input) Sentinel-1 TOPS mode burst SLC
- **SLC_par** – (input) SLC parameter file for the TOPS burst SLC
- **TOPS_par** – (input) TOPS parameter file for the TOPS burst SLC
- **SLC_out** – (output) SLC file containing a single burst
- **SLC_out_par** – (output) SLC parameter file for the single burst SLC
- **burst_num** – burst number of selected burst (1->number of bursts in the SLC)
- **drflg** –
deramp phase flag (enter - for default)
 - 0: no modification of the burst SLC phase (default)
 - 1: subtract TOPS Doppler phase ramp (deramp)
- **SLC_par2** – (output) SLC parameter file for the single burst SLC with deramped phase (drflg: 1, enter - for none)
- **dtype** –
output data type (enter - for default: same as input data):
 - 0: FCOMPLEX
 - 1: SCOMPLEX
 - NOTE: the program also supports FLOAT data; no data type conversion from or to FLOAT is possible
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_burst_corners (SLC_par, TOPS_par, kml='- ',  
                                              , logpath=None, outdir=None,  
                                              shellscript=None)
```

Calculate corner geographic coordinates of Sentinel-1 TOPS SLC bursts

Copyright 2017, Gamma Remote Sensing, v1.1 22-Aug-2017 awi/rc/cw

Parameters

- **SLC_par** – (input) SLC parameter file for the TOPS burst SLC
- **TOPS_par** – (input) TOPS parameter file for the TOPS burst SLC
- **kml** – (output) kml output file
- **logpath** (*str* or *None*) – a directory to write command logfiles to

- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_cat (SLC_1, SLC_2, SLC1_par, SLC2_par, OFF_par, SLC_3,
                                     SLC3_par, dopflg='-', iflg='-', phflg='-', gainflg='-', log-
                                     path=None, outdir=None, shellscript=None)
```

Concatenate two SLC images using 2-D SINC interpolation

Copyright 2018, Gamma Remote Sensing, v1.9 30-Apr-2018 clw/cm

Parameters

- **SLC_1** – (input) SLC-1 image
- **SLC_2** – (input) SLC-2 image to be appended to SLC-1
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset parameter file containing offset polynomials between SLC-1 and SLC-2
- **SLC_3** – (output) concatenated SLC
- **SLC3_par** – (output) ISP image parameter file for concatenated image
- **dopflg** –
Doppler flag: (enter - for default)
 - 0: ignore Doppler centroid information, assume 0 Hz Doppler centroid
 - 1: use Doppler centroid information for interpolation (default)
- **iflg** –
input data type flag: (enter - for default)
 - 0: input data are SLC images, use data type specified in SLC_par files (SCOMPLEX or FCOMPLEX) (default)
 - 1: input scenes are interferograms, force FCOMPLEX data type
- **phflg** –
phase offset correction flag: (enter - for default)
 - 0: no phase offset correction for SLC-2 (default)
 - 1: apply phase offset correction to SLC-2
- **gainflg** –
gain correction flag: (enter - for default)
 - 0: no gain correction for SLC-2 (default)
 - 1: apply gain correction to SLC-2
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_cat_S1_TOPS (SLC_tab1, SLC_tab2, SLC_tab3, log-  
path=None, outdir=None, shellscript=None)
```

Concatenate adjacent Sentinel-1 TOPS SLC images

Copyright 2018, Gamma Remote Sensing v2.2 8-May-2018 clw/cm

Parameters

- **SLC_tab1** –
(input) 3 column list of the reference TOPS SLC swaths in row order IW1, IW2, IW3... (earlier time)
SLC_tab line entries: SLC SLC_par TOPS_par
- **SLC_tab2** – (input) 3 column list of TOPS SLC-2 swaths in the same order as the
SLC_tab1 IW1, IW2, IW3... (later time)
- **SLC_tab3** – (input) 3 column list of the output concatenated TOPS swaths in the order
IW1, IW2, IW3...
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_copy (SLC_in, SLC_par_in, SLC_out, SLC_par_out, fcase='-  
, sc='- ', roff='- ', nr='- ', loff='- ', nl='- ', swap='-  
, header_lines='- ', logpath=None, outdir=None,  
shellscript=None)
```

Copy SLC with options for data format conversion, segment extraction, and byte swapping

Copyright 2015, Gamma Remote Sensing, v5.1 13-Aug-2015 uw/clw

Parameters

- **SLC_in** – (input) SLC (FCOMPLEX or SCOMPLEX format)
- **SLC_par_in** – (input) ISP SLC parameter file for input SLC
- **SLC_out** – (output) selected SLC section (FCOMPLEX or SCOMPLEX format)
- **SLC_par_out** – (output) ISP SLC parameter file of output SLC
- **fcase** –
data format conversion (enter - for default: output format = input format)
 - 1: FCOMPLEX → FCOMPLEX (default sc = 1.0)
 - 2: FCOMPLEX → SCOMPLEX (default sc = 10000.0)
 - 3: SCOMPLEX → FCOMPLEX (default sc = 0.0001)
 - 4: SCOMPLEX → SCOMPLEX (default sc = 1.0)
- **sc** – scale factor for input SLC data (enter - for default)
- **roff** – offset to starting range sample (enter - for default: 0)
- **nr** – number of range samples (enter - for default: to end of line)
- **loff** – offset to starting line (enter - for default: 0)
- **nl** – number of lines to copy (enter - for default: to end of file)
- **swap** –

swap data (enter - for default)

- 0: normal (default)
- 1: swap real/imaginary part of complex data
- 2: swap left/right (near/far range)

• **header_lines** –**number of input file header lines (enter - for default: 0)**

- NOTE: CEOS format SLC data have 1 header line
- NOTE: file offset pointer size (bytes): 8

• **logpath** (*str* or *None*) – a directory to write command logfiles to• **outdir** (*str* or *None*) – the directory to execute the command in• **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_copy_S1_TOPS (SLC1_tab,      SLC2_tab,      BURST_tab,
                                             dtype='-',    logpath=None,   outdir=None,
                                             shellscript=None)
```

Copy multiple bursts from a Sentinel-1 TOPS SLC to an output TOPS SLC
 Copyright 2018, Gamma Remote Sensing v2.4 25-Apr-2018 clw/cm

Parameters• **SLC1_tab** –

(input) 3 column list of TOPS SLC-1 swaths to be copied in row order IW1, IW2, IW3:
 SLC_tab line entries: SLC SLC_par TOPS_par

• **SLC2_tab** – (input) 3 column list of the output copied SLC-1 TOPS swaths in the order IW1, IW2, IW3• **BURST_tab** – (input) 2 column list of the first and last burst to copy from each swath, one line for each swath• **BURST_tab** –

line entries: first_burst last_burst Note: first burst is 1, enter - to select last physical burst

Note: if first_burst <= 0, then blank bursts are generated at the start of the output swath
 if last_burst exceeds the number of bursts in the input data swath, then blank bursts are appended to the end of the output swath

• **dtype** –

output data type (enter - for default: same as input data):

- 0: FCOMPLEX
- 1: SCOMPLEX
- NOTE: the program also supports FLOAT data; no data type conversion from or to FLOAT is possible

• **logpath** (*str* or *None*) – a directory to write command logfiles to• **outdir** (*str* or *None*) – the directory to execute the command in• **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_copy_WB(SLC_tab, SLC2_dir, logpath=None, outdir=None,
                                         shellscript=None)
```

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/SLC_copy_WB

Copyright 2011, Gamma Remote Sensing, v1.1 9-Apr-2011 clw

Create a new set of SLCs for all beams in a PALSAR WB ScanSAR image

Parameters

- **SLC_tab** – (input) two column list of input SLC files and SLC ISP image parameter files (including paths) (text)
- **SLC2_dir** –
directory to contain copied segments of the input SLC data and the associated parameter files

– NOTE: current directory is denoted using .
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_corners(SLC_par, terra_alt='-', kml='-', logpath=None, out-
                                         dir=None, shellscript=None)
```

Calculate SLC/MLI image corners in geodetic latitude and longitude (deg.)

Copyright 2017, Gamma Remote Sensing, v1.8 13-Dec-2017 clw/awi/cm

Parameters

- **SLC_par** – (input) ISP SLC/MLI image parameter file
- **terra_alt** – (input) average terrain altitude (enter - for default: 300.000 meters)
- **kml** – (output) kml output file (enter - for none)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_deramp(SLC_1, SLC_par1, SLC_2, SLC_par2, mode,
                                       dop_ph='-', logpath=None, outdir=None,
                                       shellscript=None)
```

Calculate and subtract Doppler phase from an SLC image

Copyright 2016, Gamma Remote Sensing, v1.5 4-Feb-2016 clw

Parameters

- **SLC_1** – (input) SLC data file (fcomplex or scomplex format)
- **SLC_par1** – (input) SLC parameter file with Doppler information
- **SLC_2** – (output) SLC with Doppler phase removed (or added)
- **SLC_par2** – (output) SLC parameter file for the output SLC
- **mode** –

mode of operation:

- 0: subtract Doppler phase ramp (deramp)
- 1: add Doppler phase ramp (reramp)

• **dop_ph** –

(output) Doppler phase (FLOAT) Note: SLC_par1 contains the Doppler polynomial that is used to calculate the Doppler phase ramp

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_deramp_S1_TOPS (SLC1_tab, SLC2_tab, mode, ph-
                                              flg, logpath=None, outdir=None,
                                              shellscript=None)
```

Calculate and subtract S1 TOPS Doppler phase from burst SLC data

Copyright 2018, Gamma Remote Sensing v1.6 25-Apr-2018 clw/cm

Parameters• **SLC1_tab** –

(input) 3 column list of TOPS SLC-1 swaths to be deramped in row order IW1, IW2, IW3:
SLC_tab line entries: SLC SLC_par TOPS_par

• **SLC2_tab** – (input) 3 column list of the output deramped SLC-1 TOPS swaths in the order IW1, IW2, IW3• **mode** –**mode of operation:**

- 0: subtract TOPS Doppler phase (deramp)
- 1: add Doppler phase ramp (reramp)

• **phflg** –**deramp phase flag:**

- 0: do not save TOPS Doppler phase (default)
- 1: save TOPS Doppler phase, output filename is the same as the deramped SLC with extension .dph

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_freq_shift (SLC, SLC_par, SLC_shift, SLC_shift_par,
                                           freq_shift, logpath=None, outdir=None,
                                           shellscript=None)
```

ISP Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/SLC_freq_shift.c

Copyright 2017, Gamma Remote Sensing, v1.0 28-Sep-2017 clw

Shift the effective radar carrier frequency of an SLC image by a specified amount

Parameters

- **SLC** – (input) SLC file (FCOMPLEX or SCOMPLEX)
- **SLC_par** – (input) SLC parameter file
- **SLC_shift** – (output) SLC data with shifted radar carrier frequency
- **SLC_shift_par** – (output) SLC parameter file with shifted radar carrier frequency
- **freq_shift** – radar carrier frequency shift (Hz)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_interp (SLC_2, SLC1_par, SLC2_par, OFF_par, SLC_2R,  
                                       SLC2R_par, loff='- ', nlines='- ', mode='- ', order='- ',  
                                       logpath=None, outdir=None, shellscript=None)
```

SLC complex image resampling using 2-D Lanczos or B-spline interpolation

Copyright 2017, Gamma Remote Sensing, v4.6 4-Dec-2017 clw/cm

Parameters

- **SLC_2** – (input) SLC-2 image to be resampled to the geometry of the SLC-1 reference image
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **SLC_2R** – (output) single-look complex image 2 coregistered to SLC-1
- **SLC2R_par** – (output) SLC-2R ISP image parameter file for coregistered image
- **loff** – offset to first valid output line (in SLC-1 lines) (enter - for default: 0)
- **nlines** – number of valid output lines (enter - or 0 for default: to end of file)
- **mode** –
 interpolation mode (enter - for default)
 - 0: Lanczos (default)
 - 1: B-spline
- **order** – Lanczos interpolator order / B-spline degree 4 -> 9 (enter - for default: 4)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_interp_S1_TOPS (SLC2_tab, SLC2_par, SLC1_tab,  
                                                SLC1_par, OFF_par, SLC2R_tab,  
                                                SLC_2R='- ', SLC2R_par='- ', mode='- ',  
                                                order='- ', logpath=None, outdir=None,  
                                                shellscript=None)
```

Resample S1 TOPS (IW mode) SLC using global offset polynomial
 Copyright 2018, Gamma Remote Sensing v2.4 25-Apr-2018 clw/cm

Parameters

- **SLC2_tab** –
 (input) 3 column list of TOPS SLC-2 swaths to be resampled to the geometry of the reference SLC1 in row order
 SLC_tab line entries: SLC SLC_par TOPS_par
- **SLC2_par** – (input) SLC parameter file of TOPS SLC-2 mosaic, SLC-2 is generated from the TOPS swaths listed in SLC2_tab
- **SLC1_tab** – (input) 3 column list of the reference TOPS SLC swaths in row order IW1, IW2, IW3
- **SLC1_par** – (input) SLC parameter file of the reference TOPS SLC-1 mosaic, SLC-1 is generated from the TOPS swaths listed in SLC1_tab
- **OFF_par** – (input) global ISP offset and interferogram parameter file, the offset model is determined from the TOPS SLC mosaics
- **SLC2R_tab** – (input) 3 column list of the output resampled SLC-2 TOPS swaths in the order IW1, IW2, IW3
- **SLC_2R** – (output) resampled mosaic generated from the swaths listed in SLC2R_tab, coregisted to the TOPS SLC-1 mosaic (enter - for none)
- **SLC2R_par** – (output) SLC parameter file associated with the resampled TOPS SLC-2R mosaic (enter - for none)
- **mode** –
 interpolation mode (enter - for default)
 - 0: Lanczos (default)
 - 1: B-spline
- **order** – Lanczos interpolator order / B-spline degree 4 -> 9 (enter - for default: 4)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_interp_map(SLC_2, SLC1_par, SLC2_par, OFF_par,
                                           SLC_2R, SLC2R_par, OFF_par2, coeffs_sm,
                                           loff='-', nlines='-', mode='-', order='-', log-
                                           path=None, outdir=None, shellscript=None)
```

SLC image resampling using a 2-D offset map
 Copyright 2017, Gamma Remote Sensing, v3.9 4-Dec-2017 clw/uw/cm

Parameters

- **SLC_2** – (input) SLC-2 image to be resampled to the reference SLC-1 reference image
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file

- **SLC_2R** – (output) single-look complex image 2 coregistered to SLC-1
- **SLC2R_par** – (output) SLC-2R ISP image parameter file for co-registered image
- **OFF_par2** – (input) ISP offset/interferogram parameter file used for residual offsets map (coffs_sm)
- **coffs_sm** – (input) smoothed residual range and azimuth offsets (fcomplex)
- **loff** – offset to first valid output line (in SLC-1 lines) (enter - for default: 0)
- **nlines** – number of valid output lines (enter - or 0 for default: to end of file)
- **mode** –
 - interpolation mode (enter - for default)**
 - 0: Lanczos (default)
 - 1: B-spline
- **order** – Lanczos interpolator order / B-spline degree 4 -> 9 (enter - for default: 4)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_intf (SLC_1, SLC_2R, SLC1_par, SLC2R_par, OFF_par, interf, rlks, azlks, loff='-', nlines='-', sps_flg='-', azf_flg='-', rp1_flg='-', rp2_flg='-', SLC_1s='-', SLC_2Rs='-', SLC_1s_par='-', SLC_2Rs_par='-', az_beta='-', logpath=None, outdir=None, shellscript=None)
```

Interferogram generation from co-registered SLC data

Copyright 2017, Gamma Remote Sensing, v5.5 clw/uw/cm 26-Aug-2017

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)
- **SLC_2R** – (input) single-look complex image 2 coregistered to SLC-1
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2R_par** – (input) SLC-2R ISP image parameter file for the co-registered image
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **interf** – (output) interferogram from SLC-1 and SLC-2R
- **rlks** – number of range looks
- **azlks** – number of azimuth looks
- **loff** – offset to starting line relative to SLC-1 for interferogram (default=0)
- **nlines** – number of SLC lines to process (enter - for default: to end of file)
- **sps_flg** –
 - range spectral shift flag:**
 - 1: apply range spectral shift filter (default)
 - 0: do not apply range spectral shift filter

- **azf_flg** –
azimuth common band filter flag:
 - 1: apply azimuth common-band filter (default)
 - 0: do not apply azimuth common band filter
- **rp1_flg** –
SLC-1 range phase mode
 - 0: nearest approach (zero-Doppler) phase
 - 1: ref. function center (Doppler centroid) (default)
- **rp2_flg** –
SLC-2 range phase mode
 - 0: nearest approach (zero-Doppler) phase
 - 1: ref. function center (Doppler centroid) (default)
- **SLC_1s** – SLC-1 after range spectral shift and azimuth common-band filtering (FCOMPLEX format) (enter - for none)
- **SLC_2Rs** – SLC-2R after range spectral shift and azimuth common-band filtering (FCOMPLEX format) (enter - for none)
- **SLC_1s_par** – SLC-1s ISP image parameter file (enter - for none)
- **SLC_2Rs_par** – SLC-2Rs ISP image parameter file (enter - for none)
- **az_beta** – azimuth common-band filter Kaiser window parameter (default: 2.120)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_mosaic_S1_TOPS(SLC_tab, SLC, SLC_par, rlks, azlks,
                                              wflg='-', SLCR_tab='-', logpath=None,
                                              outdir=None, shellscript=None)
```

Calculate SLC mosaic of Sentinel-1 TOPS burst SLC data

Copyright 2018, Gamma Remote Sensing v3.6 25-Apr-2018 clw/awi/cm

Parameters

- **SLC_tab** – (input) 3 column list of SLC, SLC_par, Sentinel-1 TOPS_par sorted in the order IW1, IW2, IW3...
- **SLC** – (output) SLC mosaic image
- **SLC_par** – (output) SLC mosaic image parameter file
- **rlks** – number of range looks used to determine burst window boundaries for the mosaic
- **azlks** – number of azimuth looks used to determine burst window boundaries for the mosaic
- **wflg** –
burst window calculation flag:

- 0: use existing burst window parameters if they exist, otherwise calculate burst window parameters (default)
- 1: calculate burst window parameters from burst parameters and the number of range and azimuth looks

- **SLCR_tab** –

(input) SLC_tab of the reference scene, 3 column list of SLC, SLC_par, TOPS_par sorted sorted in the order

- NOTE: When generating a mosaic of a resampled SLC, the SLC_tab of the reference scene is required

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_ovr (SLC, SLC_par, SLC_ovr, SLC_ovr_par, r_ovr, log-  
path=None, outdir=None, shellscript=None)
```

ISP Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/SLC_ovr.c

Copyright 2017, Gamma Remote Sensing, v1.9 12-Oct-2017 clw/cm

Oversample or subsample SLC data in slant-range

Parameters

- **SLC** – (input) SLC file (fcomplex or scomplex)
- **SLC_par** – (input) SLC parameter file of SLC file
- **SLC_ovr** – (output) range resampled SLC file (fcomplex or scomplex)
- **SLC_ovr_par** – (output) SLC parameter file of range resampled SLC data file
- **r_ovr** –
integer range oversampling factor (2 → 16) if r_ovr < 0, the SLC will be subsampled,
integer range subsampling factor (-2 → -16)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_ovr2 (SLC, SLC_par, SLC_ovr, SLC_ovr_par, r_ovr, az_ovr='-',  
logpath=None, outdir=None, shellscript=None)
```

Parameters

- **SLC** – (input) SLC file (SCOMPLEX or FCOMPLEX, e.g. 20141126.SLC)
- **SLC_par** – (input) SLC parameter file (e.g. 20141126.SLC.par)
- **SLC_ovr** – (output) oversampled SLC file (same format as SLC)
- **SLC_ovr_par** – (output) SLC parameter file of oversampled SLC file
- **r_ovr** – range oversampling factor (e.g. 2.5)

- **az_ovr** – azimuth oversampling factor (e.g. 2.5, default = 1.0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SLC_phase_shift (SLC_1, SLC_par1, SLC_2, SLC_par2,
                                             ph_shift, logpath=None, outdir=None,
                                             shellscript=None)
```

Add a constant phase from an SLC image

Copyright 2015, Gamma Remote Sensing, v1.1 1-Dec-2015 clw

Parameters

- **SLC_1** – (input) SLC data file (fcomplex or scomplex format)
- **SLC_par1** – (input) SLC parameter file
- **SLC_2** – (output) SLC with phase shift
- **SLC_par2** – (output) SLC parameter file for the output SLC
- **ph_shift** –
phase shift to add to SLC phase (radians)
 - NOTE: Used to apply a constant phase shift of -1.25 radians to Sentinel-1 TOPS SLC data from swath IW1 acquired up to 10-Mar-2015. Used to apply a constant phase shift of -3.83 radians to Sentinel-1 TOPS SLC data with H-POL on receive (e.g. VH) acquired up to 10-Mar-2015.
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.SR_to_GRD (MLI_par, OFF_par, GRD_par, in_file, out_file, rlks='-',
                                       azlks='-', interp_mode='-', grd_rsp='-', grd_azsp='-',
                                       logpath=None, outdir=None, shellscript=None)
```

Conversion to ground range for ISP MLI and INSAR data of type float

Copyright 2009, Gamma Remote Sensing, v1.9 7-May-2009 uw/clw

Parameters

- **MLI_par** – (input) MLI image parameter file of input slant range image (float)
- **OFF_par** – (input) ISP offset/interferogram parameter file of input image (enter - image in MLI geometry)
- **GRD_par** – (input/output) image parameter file of output ground range image
- **in_file** – (input) slant range image (float)
- **out_file** – (output) ground range image (float)
- **rlks** – multi-looking in range (prior to resampling, default=1)
- **azlks** – multi-looking in azimuth (prior to resampling, default=1)
- **interp_mode** –

interpolation mode

- 0: nearest neighbor (default)
- 1: spline
- 2: spline log
- **grd_rsp** – output image ground range sample spacing (m) (default = (input image azimuth spacing) * azlks)
- **grd_azsp** – output image azimuth sample spacing (m) (default = (input image azimuth spacing) * azlks)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.TX_SLC_preproc(TSX_list, SLC_dir, log, logpath=None, outdir=None, shellscript=None)
```

Preprocessing of TerraSAR-X TDX1 and TSX1 SLC products using par_TX_SLC

Copyright 2013, Gamma Remote Sensing, v1.2 22-Oct-2013 clw

Parameters

- **TSX_list** –
(input) **single column text file with directories (including path)** containing path to directory containing product XML for IMAGEDATA/*.cos files
- **SLC_dir** – directory for output SLC data files and SLC parameter files
- **log** – (output) processing log file
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.UNWRAP(interf, cc, pwr, unwrap, flag, width, lines, corr_thr='-'  
pwr_thr='- ', r_init='- ', az_init='- ', r1='- ', r2='- ', l1='- '  
l2='- ', logpath=None, outdir=None, shellscript=None)
```

Parameters

- **interf** – interferogram filename (*.int, *.flt)
- **cc** – correlation filename (*.cc)
- **pwr** – intensity image (*.pwr, *.mli)
- **unwrap** – unwrap output file (*.unw)
- **flag** – unwrapping flag file (*.flag)
- **width** – interferogram width
- **lines** – number of interferogram lines
- **corr_thr** – threshold for correlation in the unwrapping mask (default=.3)

- **pwr_thr** – intensity threshold for phase unwrapping neutrons, multiples of average (default = 6.)
- **r_init** – range seed location in the interferogram
- **az_init** – azimuth seed location in the interferogram
- **r1** – starting range sample offset to unwrap
- **r2** – ending range sample offset to unwrap
- **l1** – starting line offset to unwrap
- **l2** – ending line offset to unwrap
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.UNWRAP_PAR(interf_par, interf, cc, pwr, unwrap, flag, corr_thr='-',
                                       pwr_thr='-', r_init='-', az_init='-', r1='-', r2='-',
                                       l1='-', l2='-', logpath=None, outdir=None,
                                       shellscript=None)
```

Parameters

- **interf_par** – interferogram parameter file *.off
- **interf** – interferogram filename (*.int, *.flt)
- **cc** – correlation filename (*.cc)
- **pwr** – intensity image (*.pwr, *.mli)
- **unwrap** – unwrap output file (*.unw)
- **flag** – unwrapping flag file (*.flag)
- **corr_thr** – threshold for correlation in the unwrapping mask (default=.3)
- **pwr_thr** – intensity threshold for phase unwrapping neutrons, multiples of average (default = 6.)
- **r_init** – range seed location in the interferogram
- **az_init** – azimuth seed location in the interferogram
- **r1** – starting range sample offset to unwrap
- **r2** – ending range sample offset to unwrap
- **l1** – starting line offset to unwrap
- **l2** – ending line offset to unwrap
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.adapt_filt(int, sm, width, low_SNR_thr='-', filt_width='-',
                                     xmin='-', xmax='-', ymin='-', ymax='-', log-
                                     path=None, outdir=None, shellscript=None)
```

Adaptive bandpass filtering of interferograms

Copyright 2016, Gamma Remote Sensing, v3.5 clw 17-Feb-2016

Parameters

- **int** – (input) complex interferogram image filename
- **sm** – (output) smoothed interferogram filename
- **width** – number of samples/row
- **low_SNR_thr** – low SNR threshold (default = .25);
- **filt_width** – filter width in pixels (default = 1.0)
- **xmin** – offset to starting range pixel (default = 0)
- **xmax** – offset last range pixel (default = width-1)
- **ymin** – offset to starting azimuth row (default = 0)
- **ymax** – offset to last azimuth row (default = nlines-1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.adf(interf, sm, cc, width, alpha='-', nfft='-', cc_win='-', step='-',
                              loff='-', nlines='-', wfrac='-', logpath=None, outdir=None,
                              shellscript=None)
```

Adaptive spectral filtering for complex interferograms

Copyright 2017, Gamma Remote Sensing, v3.6 12-Oct-2017 clw

Parameters

- **interf** – (input) interferogram (fcomplex)
- **sm** – (output) filtered interferogram (fcomplex)
- **cc** – (output) coherence derived from filtered interferogram (float)
- **width** – number of samples/line
- **alpha** – exponent for non-linear filtering (enter - for default: 0.40)
- **nfft** – filtering FFT window size, 2*N, 8 → 512, (enter - for default: 32)
- **cc_win** – coherence parameter estimation window size odd, max: 15 (enter - for default: 5)
- **step** – processing step (enter - for default: nfft/8)
- **loff** – offset to starting line to process (enter - for default: 0)
- **nlines** – number of lines to process (enter - for default: to end of file)
- **wfrac** – minimum fraction of points required to be non-zero in the filter window (enter - for default: 0.200)

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.af_SLC (SLC_par, SLC, rwin='-', azwin='-', dr='-', daz='-', thres='-',
                                   a1_flg='-', b0_flg='-', offsets='-', n_ovr='-', roff='-',
                                   azoff='-', logpath=None, outdir=None, shellscript=None)
```

Focus testing for SLC data using autofocus estimation of effective velocity

Copyright 2016, Gamma Remote Sensing, v1.4 16-Feb-2016 clw/uw

Parameters

- **SLC_par** – (input) ISP SLC image parameter file
- **SLC** – (input) single-look complex image
- **rwin** – range window size (enter - for default: 1024)
- **azwin** – azimuth window size (enter - for default: 4096)
- **dr** – range sample increment (enter - for default: 1024, enter 0 for single patch)
- **daz** – azimuth line increment (enter - for default: 8192, enter 0 for single patch)
- **thres** – offset estimation SNR threshold (enter - for default: 10.000)
- **a1_flg** –
fit a1 for first derivative of the effective velocity w.r.t.range
 - 0: no (default)
 - 1: yes
- **b0_flg** –
fit b0 for first derivative of the effective velocity w.r.t. along-track time
 - 0: no (default)
 - 1: yes
- **offsets** – (output) range and azimuth offsets and SNR data in text format, enter - for no output
- **n_ovr** – SLC oversampling factor (1,2,4: enter - for default: 1)
- **roff** – range offset for single patch center
- **azoff** – azimuth offset for single patch center
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ave_image (im_list, width, ave, start='-', nlines='-', pixav_x='-',
                                       pixav_y='-', zflag='-', nmin='-', logpath=None, out-
                                       dir=None, shellscript=None)
```

Calculate average of a stack of images (float format)

Copyright 2015, Gamma Remote Sensing, v1.9 20-Nov-2015 clw

Parameters

- **im_list** – (input) text file with names of co-registered images in column 1 (float)
- **width** – number of samples/line
- **ave** – (output) average of input image data files (float)
- **start** – starting line (default: 1)
- **nlines** – number of lines to process (enter - for default: entire file)
- **pixav_x** – number of pixels to average in width (default: 1)
- **pixav_y** – number of pixels to average in height (default: 1)
- **zflag** –
zero flag:
 - 0: interpret 0.0 as missing data value (default)
 - 1: interpret 0.0 as valid data
- **nmin** – minimum number of images required to calculate the average if zflag = 0 (default: 3/4*nfiles)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.az_integrate` (*data*, *width*, *azi*, *cflg*, *scale*='-', *lz*='-', *logpath*=*None*, *outdir*=*None*, *shellscript*=*None*)

Calculate azimuth integral of float data (unwrapped phase or azimuth offsets)

Copyright 2012, Gamma Remote Sensing, v1.2 6-Feb-2012

Parameters

- **data** – (input) input data (example: SBI dtrapped phase) (float)
- **width** – (input) number of range samples/line
- **azi** – (output) input data integrated along azimuth (float)
- **cflg** –
integration constant flag:
 - 0: set azimuth integral value to 0.0 at specified line
 - 1: set average of the azimuth integral to 0.0
- **scale** – scale factor to apply to the data (enter - for default, default: 1.0)
- **lz** – line offset where the azimuth integral is set to 0.0 (cflg = 0, enter - for default, default: 0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.az_spec_SLC (SLC, SLC_par, spectrum, roff='- ', namb='- ', pltflg='- ',
                                         logpath=None, outdir=None, shellscript=None)
```

Doppler centroid estimate from SLC images

Copyright 2016, Gamma Remote Sensing, v2.9 clw 15-Feb-2016

Parameters

- **SLC** – (input) SAR image data file (fcomplex or scomplex format)
- **SLC_par** – (input) ISP SLC image parameter file
- **spectrum** – (output) Doppler spectrum (text format)
- **roff** – range sample offset to center of estimation window (enter - for default=center_swath)
- **namb** – number of multiples of the PRF to add to the estimated centroid (default=0)
- **pltflg** –
azimuth spectrum plotting flag:
 - 0: none (default)
 - 1: output plot in PNG format
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.base_copy (SLC1_par, baseline_1, SLC2_par, baseline_2,
                                       time_rev='- ', logpath=None, outdir=None,
                                       shellscript=None)
```

Calculate baseline file for a subsection of a reference SLC

Copyright 2003, Gamma Remote Sensing, v1.1 6-Jan-2003 ts/clw/uw

Parameters

- **SLC1_par** – (input) ISP image parameter file of the reference SLC
- **baseline_1** – (input) baseline file derived using the reference SLC geometry
- **SLC2_par** – (input) ISP image parameter file corresponding to the subsection of the reference SLC
- **baseline_2** – (output) baseline file derived using the geometry and timing of the SLC subsection
- **time_rev** – SLC image normal=1, time-reversed = -1 (default=1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.base_est_fft (interf, SLC1_par, OFF_par, baseline, nazfft='- ',
                                         r_samp='- ', az_line='- ', logpath=None, outdir=None,
                                         shellscript=None)
```

Estimate baseline from interferogram fringe spectrum

Copyright 2016, Gamma Remote Sensing, v2.1 clw/uw 20-Feb-2016

Parameters

- **interf** – (input) multi-look interferogram with range phase
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **baseline** – (output) baseline file
- **nazfft** – size of azimuth FFT (lines read from file, 2**N) (default: 512)
- **r_samp** – range pixel offset to center of the FFT window (default: center)
- **az_line** – line offset from start of the interf. for the FFT window (default=center)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.base_init(SLC1_par, SLC2_par, OFF_par, interf, baseline,  
                                     mflag='-', nrfft='-', nazfft='-', r_samp='-', az_line='-',  
                                     logpath=None, outdir=None, shellscript=None)
```

Estimate initial baseline using orbit state vectors, offsets, and interferogram phase

Copyright 2016, Gamma Remote Sensing, v2.5 clw/uw 19-Feb-2016

Parameters

- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file (enter - for none)
- **interf** – (input) unflattened interferogram (enter - for none)
- **baseline** – (output) baseline parameter file
- **mflag** – baseline estimation method flag (enter - for default)
- **mflag** –
 - b_para b_perp input**
 - 0: orbits orbits p1,p2 (default)
 - 1: offsets offsets p1,p2,off
 - 2: orbits fft p1,p2,off,int
 - 3: offsets fft p1,p2,off,int
 - 4: fft fft p1,off,int
- **nrfft** – size of range FFT (512, 1024,...) (enter - for default determined from image width)
- **nazfft** – size of azimuth FFT (512, 1024,...) (enter - for default determined from image azimuth lines)

- **r_samp** – range pixel offset to center of the FFT window (enter - for default, default: range center)
- **az_line** –
line offset from start of the interf. for the FFT window (enter - for default, default: azimuth center)
- NOTE: Not all input data files are required for the different methods enter - for files that are not provided
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.base_ls(SLC_par, OFF_par, gcp_ph, baseline, ph_flag='-',
                                   bc_flag='-', bn_flag='-', bcdot_flag='-', bndot_flag='-',
                                   bperp_min='-', SLC2R_par='-', logpath=None, outdir=None, shellscript=None)
```

Least squares baseline estimation using terrain heights

Copyright 2005, Gamma Remote Sensing, v2.2 5-Sep-2005 clw/uw

Parameters

- **SLC_par** – (input) ISP parameter file of the reference SLC
- **OFF_par** – (input) ISP interferogram/offset parameter file
- **gcp_ph** – (input) ground control point heights + extracted unwrapped phase (text format)
- **baseline** – (input) baseline parameter file
- **ph_flag** – restore range phase ramp (default=0: do not restore 1: restore)
- **bc_flag** – cross-track baseline component estimate (0:orbit derived 1:estimate from data, default=1)
- **bn_flag** – normal baseline component estimate (0:orbit derived 1:estimate from data, default=1)
- **bcdot_flag** – cross-track baseline rate estimate (0:orbit derived 1:estimate from data, default=1)
- **bndot_flag** – normal baseline rate estimate (0:orbit derived 1:estimate from data, default=0)
- **bperp_min** – minimum perpendicular baseline required for L.S estimation (m, default=10.0)
- **SLC2R_par** – (input) parameter file of resampled SLC, required if SLC-2 frequency differs from SLC-1
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.base_orbit(SLC1_par, SLC2_par, baseline, logpath=None, outdir=None, shellscript=None)
```

Estimate baseline from orbit state vectors

Copyright 2015, Gamma Remote Sensing, v4.2 clw 18-Apr-2018

Parameters

- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **baseline** – (output) baseline file (text format, enter - for none)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.base_perp(baseline, SLC1_par, OFF_par, time_rev='- ', log-  
path=None, outdir=None, shellscript=None)
```

Calculate baseline components perpendicular and parallel to look vector

Copyright 2005, Gamma Remote Sensing, v3.5 10-May-2005 clw/uw

Parameters

- **baseline** – (input) baseline file (text)
- **SLC1_par** – (input) ISP parameter file of SLC-1 (reference SLC)
- **OFF_par** – (input) ISP interferogram/offset parameter file
- **time_rev** – SLC image normal=1 (default), image time-reversed = -1
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.bpf(data_in, data_out, width, fc_x, bw_x, fc_y, bw_y, roff='- ', azoff='- '  
, nr='- ', naz='- ', data_type='- ', f_mode='- ', beta='- ', fir_len='- '  
, logpath=None, outdir=None, shellscript=None)
```

Interferometric SAR Processor (ISP): Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/bpf.c

Copyright 2016, Gamma Remote Sensing, v1.7 clw 3-Mar-2016

Bandpass filter for 2-dimensional complex image data (FCOMPLEX or SCOMPLEX format)

Parameters

- **data_in** – (input) input data file (fcomplex, scomplex, float)
- **data_out** – (output) output data file (fcomplex, scomplex, float)
- **width** – number of samples/line
- **fc_x** – normalized x-coord. (across) filter center frequency (range: -0.5 → 0.5)
- **bw_x** – normalized x-coord. bandwidth (range: 0 → 1.0)
- **fc_y** – normalized y-coord. (down) filter center frequency (range: -0.5 → 0.5)
- **bw_y** – normalized y-coord. bandwidth (range: 0 → 1.0)
- **roff** – offset to starting range to filter (default: 0)

- **azoff** – offset to starting azimuth to filter (default: 0)
- **nr** – number of range pixels to filter (default - : width - roff)
- **naz** – number of azimuth lines to filter (default - : nlines - azoff)
- **data_type** – data type (default 0:fcomplex, 1:scomplex, 2:float)
- **f_mode** – fill mode (default 0:force filtered value to 0.0 for input value 0.0, 1:no forcing)
- **beta** – Kaiser window beta parameter (default - : 1.000)
- **fir_len** – finite impulse response filter length (default - : 128)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.bpf_ssi(SLC, SLC_par, SLC_flow, SLC_flow_par, SLC_fhigh,
                                   SLC_fhigh_par, rbs='-', logpath=None, outdir=None,
                                   shellscript=None)
```

Parameters

- **SLC** – (input) SLC (FCOMPLEX or SCOMPLEX, SLC should not be resampled)
- **SLC_par** – (input) SLC parameter file
- **SLC_flow** – (output) low frequency band filtered SLC (FCOMPLEX or SCOMPLEX)
- **SLC_flow_par** – (output) low frequency band filtered SLC parameter file
- **SLC_fhigh** – (output) high frequency band filtered SLC (FCOMPLEX or SCOMPLEX)
- **SLC_fhigh_par** – (output) high frequency band filtered SLC parameter file (FCOMPLEX or SCOMPLEX)
- **rbs** –
relative range spectrum band separation (default = 0.6666 → lowest and highest third of processing bandwidth)
indicate - for the output files to only calculate filtering parameters
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.bridge(int, flag, unw, bridge, width, xmin='-', xmax='-', ymin='-',
                                 ymax='-', logpath=None, outdir=None, shellscript=None)
```

Phase unwrap new regions with bridges to regions already unwrapped

Copyright 2010, Gamma Remote Sensing, v1.5 clw 4-Nov-2010

Parameters

- **int** – (input) interferogram (fcomplex)
- **flag** – (input) unwrapping flag file
- **unw** – (input/output) unwrapped phase (float)

- **bridge** – (input) bridge data file (text format)
- **width** – number of samples/row
- **xmin** – starting range pixel offset to unwrap (default = 0)
- **xmax** – last range pixel offset to unwrap (default=width-1)
- **ymin** – starting azimuth row offset to unwrap, relative to start (default = 0)
- **ymax** – last azimuth row offset to unwrap, relative to start (default = nlines-1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.cc_wave(interf, MLI_1, MLI_2, cc, width, bx='-', by='-', wflg='-',  
                                   xmin='-', xmax='-', ymin='-', ymax='-', logpath=None,  
                                   outdir=None, shellscript=None)
```

Estimate interferometric coherence

Copyright 2017, Gamma Remote Sensing, v6.0 24-Oct-2017 clw/uw

Parameters

- **interf** – (input) normalized complex interferogram
- **MLI_1** – (input) intensity image of the first scene (float) (enter - for none)
- **MLI_2** – (input) intensity image of the second scene (float) (enter - for none)
- **cc** – (output) estimated degree of coherence filename
- **width** – number of samples/line
- **bx** – estimation window size in columns (enter - for default:5.0)
- **by** – estimation window size in lines (enter - for default:5.0)
- **wflg** –
 - estimation window (enter - for default):**
 - 0: rectangular (default)
 - 1: triangular
 - 2: Gaussian
 - 3: normalized vector sum with rectangular window
 - NOTE: This estimator does not use the MLI data, even when specified
- **xmin** – starting range pixel offset (enter - for default: 0)
- **xmax** – last range pixel offset (enter - for default: width - 1)
- **ymin** – starting azimuth row offset, relative to start (enter - for default: 0)
- **ymax** – last azimuth row offset, relative to start (enter - for default: nlines - 1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.clear_flag(flag, width, flag_bits, xmin, xmax, ymin, ymax, logpath=None, outdir=None, shellscript=None)`

Clear phase unwrapping flag bits

Copyright 2005, Gamma Remote Sensing, v1.6 clw 17-Oct-2005

Parameters

- **flag** – (input)phase unwrapping flag filename
- **width** – number of samples/row
- **flag_bits** –
byte with value of flag(s) to be cleared: Charges = 3 Guides = 4 Low SNR = 8 Visited = 16 BRANCH PT. = 32 Cuts = 64 Lawn = 128
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row offset, relative to start (default = 0)
- **ymax** – last azimuth row offset, relative to start (default = nlines-1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.corr_flag(corr, flag, width, corr_thr, xmin='-', xmax='-', ymin='-', ymax='-', border='-', logpath=None, outdir=None, shellscript=None)`

Low correlation region detection for phase unwrapping

Copyright 2005, Gamma Remote Sensing, v2.4 1-Mar-2005 clw/uw

Parameters

- **corr** – (input)interferometric correlation file
- **flag** – (input/output) phase unwrapping flag filename
- **width** – number of samples/row
- **corr_thr** – correlation threshold (0 → 1.0)
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row offset, relative to start (default = 0)
- **ymax** – last azimuth row offset, relative to start (default = nlines-1)
- **border** – effective range of low coherence pixels to set low coherence flag (default=2)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.create_offset(SLC1_par, SLC2_par, OFF_par, algorithm='-',  
                                         rlks='-', azlks='-', iflg='-', logpath=None, out-  
                                         dir=None, shellscript=None)
```

Create and update ISP offset and interferogram parameter files

Copyright 2015 Gamma Remote Sensing v5.3 clw/uw 10-Nov-2015

Parameters

- **SLC1_par** – (input) SLC-1/MLI-1 ISP image parameter filename (reference)
- **SLC2_par** – (input) SLC-2/MLI-2 ISP image parameter filename
- **OFF_par** – (input/output) ISP offset/interferogram parameter file
- **algorithm** –
 offset estimation algorithm
 - 1: intensity cross-correlation (default)
 - 2: fringe visibility
- **rlks** – number of interferogram range looks (enter - for default: 1)
- **azlks** – number of interferogram azimuth looks (enter - for default: 1)
- **iflg** –
 interactive mode flag (enter - for default)
 - 0: non-interactive
 - 1: interactive (default)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.dcomp_sirc(infile, outfile, samples, loff='-', nlines='-', log-  
                                         path=None, outdir=None, shellscript=None)
```

Extract SIR-C SLC compressed single-pol data

Copyright 2009, Gamma Remote Sensing, v1.4 16-Oct-2009 clw

Parameters

- **infile** – (input) SIR-C single-pol SLC compressed data
- **outfile** – (output) complex floating point data
- **samples** – number of polarimetric samples per input line (4 bytes/sample)
- **loff** – offset to starting line (default: 0)
- **nlines** – number of lines to copy (default: entire file, 0 = entire file)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.dcomp_sirc_quad(infile, outfile, samples, parameter, loff='-'
                                          ', nlines=-', logpath=None, outdir=None,
                                          shellscript=None)
```

Extract SIR-C MLC or SLC compressed quad-pol data

Copyright 2009, Gamma Remote Sensing, v1.4 16-Oct-2009 uw/clw

Parameters

- **infile** – (input) SIR-C SLC or MLC quad-pol compressed data
- **outfile** – (output) complex floating point data
- **samples** – number of polarimetric samples per input line (10 bytes/sample)
- **parameter** –
 polarimetric parameter to extract from SLC or MLC product:
 - 0: SLC total power
 - 1: SLC-HH
 - 2: SLC-HV
 - 3: SLC-VH
 - 4: SLC-VV
 - 5: MLC total power
 - 6: MLC-HVHV*
 - 7: MLC-VVVV*
 - 8: MLC-HHHH*
 - 9: MLC-HHHV*
 - 10: MLC-HHVV*
 - 11: MLC-HVVV*
- **loff** – offset to starting line (default: 0)
- **nlines** – number of lines to copy (default: entire file, 0 = entire file)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.error_stat(d1, d2, width, dtype, roff, loff, nr, nl, report, log-
                                     path=None, outdir=None, shellscript=None)
```

Calculate statistics for two data files and their difference (FLOAT or FCOMPLEX)

Copyright 2017, Gamma Remote Sensing, v1.2 clw 7-Jan-2016

Parameters

- **d1** – (input) data file 1
- **d2** – (input) data file 2
- **width** – image line width (samples/line)

- **dtype** –
data type for d1 and d2:
 - 0: FLOAT
 - 1: FCOMPLEX
- **roff** – sample offset to region start (enter - for default: 0)
- **loff** – line offset to region start (enter - for default: 0)
- **nr** – region width (samples, enter - for default: width - roff)
- **nl** – number of lines in the region (enter - for default: data_lines - loff)
- **report** –
output text file (keyword:value format) keywords: data_1, data_2, d1_mean, d2_mean, d1_stddev, d2_stddev, root_mean_square_error, normalized_mean_square_error, cross_correlation_coefficient, cross_correlation_angle, total_samples, non_zero_samples, valid_fraction
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.fill_gaps(data_in, width, data_out, method='-', max_dist='-',  
                                     bp_flag='-', win='-', ds_method='-', ds_size='-',  
                                     ds_data='-', logpath=None, outdir=None,  
                                     shellscript=None)
```

Fill gaps in 2D raster file

Copyright 2017, Gamma Remote Sensing, v1.6 6-Apr-2017 cm

Parameters

- **data_in** – (input) input data file (float)
- **width** – width of input data
- **data_out** – (output) output data file (float)
- **method** –
method flag (enter - for default: 1)
 - 0: Laplace interpolation and linear extrapolation - least squares solution
 - 1: Laplace interpolation and linear extrapolation - smaller system of linear equations than in method #0 in case of few missing values - least squares solution (default)
 - 2: Laplace interpolation and linear extrapolation - solves a direct linear system of equations for the missing values (not a least squares solution)
 - 3: biharmonic interpolation - implementation similar to method #1 - least squares solution
 - 4: spring analogy: assumes springs (with a nominal length of zero) connect each node with every neighbor - least squares solution

- 5: average of the 8 nearest neighbors - this method solves a direct linear system for the missing values (not a least squares solution) hints: small gaps: use method #0, #1 or #3 - large gaps: use method #2, #4 or #5 - most demanding: method #3
- **max_dist** – maximum interpolation / extrapolation distance in pixels (enter - or 0 for default: unlimited)
- **bp_flag** –
perform block processing (enter - for default: 0)
 - 0: no block processing (default)
 - 1: block processing (faster, avoid overflow, however might be slightly less accurate) when block processing is selected, a two-step process is carried out: 1: solving the downsampled array (coarse processing), 2: block processing
- **win** – block size (pixels, $10 < \text{win} < 1000$, enter - for default: 100)
- **ds_method** –
method flag (0 - 5, same choices as for [method] option) (enter - for default: same as [method])
 hint: for an input containing large gaps, method #2, #4 or #5 may yield more appropriate results.
- **ds_size** – maximum size of downsampled data (for both width and height) (pixels, $\text{ds_size} > 10$, enter - for default: 400)
- **ds_data** – (output) write intermediate data after solving the downsampled array (float)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.fspf (data_in, data_out, width, dtype='-', r_max='-', spf_type='-',
                                MLI_par='-', logpath=None, outdir=None, shellscript=None)
```

ISP Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/fspf.c
 Copyright 2014, Gamma Remote Sensing, v1.2 28-May-2014 of/clw/uw
 Fast spatial filter for 2D data

Parameters

- **data_in** – (input) input image data
- **data_out** – (output) spatially filtered image data
- **width** – number of samples/row
- **dtype** –
data type (enter - for default):
 - 0: FCOMPLEX
 - 1: SCOMPLEX
 - 2: FLOAT (default)
- **r_max** – maximum filter radius (range samples) (enter - for default: 64)
- **spf_type** –
spatial filter type (enter - for default):

- 0: uniform average (default for fcomplex and scomplex)
- 1: triangular weighted average: $1 - (r/r_max)$
- 2: quadratic weighted average: $1 - (r/r_max)**2$
- 3: Gaussian weighted average: $\exp(-2.*(r**2/r_max**2))$
- 4: linear least-squares (default for float data)
- **MLI_par** – MLI or SLC parameter file with the same number of looks as the input image, required for GPRI data
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.gcp_phase (unw, OFF_par, gcp, gcp_ph, win_sz='-', logpath=None,
                                      outdir=None, shellscript=None)
```

Extract unwrapped phase at GCP locations

Copyright 2006, Gamma Remote Sensing, v1.5 8-Mar-2006 clw

Parameters

- **unw** – (input) unwrapped interferometric phase
- **OFF_par** – (input) ISP interferogram/offset parameter file
- **gcp** – (input) ground control point data (text format)
- **gcp_ph** – (output) ground control point data + extracted unwrapped phase (text)
- **win_sz** – window size for averaging phase for each GCP, must be odd (default: 1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.get_GAMMA_RASTER (mode, logpath=None, outdir=None,
                                             shellscript=None)
```

Script to determine the default extension for raster images or the operating system type v1.1 clw/uw

11-Nov-2016

Parameters

- **mode** –

Specify the script string output:

- 0: raster file extension (ras, bmp, or tif)
- 1: OS type: Linux, MINGW64_NT-10.0, CYGWIN_NT-10.0, darwin...
- NOTE: The default raster format on Linux systems is SUN_RASTER (*.ras), for all other operating systems it is BMP (*.bmp). SUN_RASTER and BMP images are limited in size to 32767 x 32767. TIFF files do not have this limitation. To set the default image raster format for Gamma programs, set the environment variable GAMMA_RASTER as follows: bash: export GAMMA_RASTER=SUN_RASTER
#extension: ras export GAMMA_RASTER=BMP #extension: bmp export

GAMMA_RASTER=TIFF #extension: tif csh,tcsh: setenv GAMMA_RASTER
 SUN_RASTER #extension: ras setenv GAMMA_RASTER BMP #extension: bmp
 setenv GAMMA_RASTER TIFF #extension: tif Environment variables can be set
 either in processing scripts, or in the shell initialization file (e.g. .bashrc) Pro-
 grams in the Gamma software that generate raster image files query the value of
 GAMMA_RASTER if it has been defined. This script can be called from within
 another script to determine the default raster image format or OS type: bash:
 \$ext='get_GAMMA_RASTER 0' csh,tcsh: set ext='get_GAMMA_RASTER 0' The
 variable \$ext can then be used to specify the format of the output raster file by using
 it to construct the output file name: bash: \$my_raster=\$my_name"."\$ext csh,tcsh: set
 my_raster=\$my_name"."\$ext OS: Linux GAMMA_RASTER: Undefined variable.

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.grasses` (*int, flag, unw, width, xmin='-', xmax='-', ymin='-', ymax='-', xinit='-', yinit='-', init_ph='-', logpath=None, outdir=None, shellscript=None*)

Phase unwrapping by region growing

Copyright 2005, Gamma Remote Sensing, v4.2 1-Mar-2005 clw/uw

Parameters

- **int** – (input) interferogram filename
- **flag** – (input) unwrapping flag filename
- **unw** – (output) unwrapped phase filename
- **width** – number of samples/row
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default=width-1)
- **ymin** – starting azimuth row offset, relative to start (default = 0)
- **ymax** – last azimuth row offset, relative to start (default = nlines-1)
- **xinit** – starting range pixel for unwrapping (default = width/2)
- **yinit** – starting row to unwrap (default = height/2)
- **init_ph** – flag to set phase at starting point to 0.0 (default 0: not set to 0.0, 1: set to 0.0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.hgt_map` (*unw, SLC_par, OFF_par, baseline, hgt, gr, ph_flag='-', loff='-', nlines='-', SLC2R_par='-', logpath=None, outdir=None, shellscript=None*)

Interferometric height/ground range estimation vs. slant range

Copyright 2005, Gamma Remote Sensing, v5.1 clw/uw 9-Sep-2005

Parameters

- **unw** – (input) unwrapped interferometric phase
- **SLC_par** – (input) ISP parameter file for the reference SLC
- **OFF_par** – (input) ISP offset/interferogram processing parameters
- **baseline** – (input) baseline parameter file
- **hgt** – (output) height file (in slant range geometry) relative to the WGS-84 ellipsoid
- **gr** – (output) cross-track ground ranges on the WGS-84 ellipsoid (in slant range geometry)
- **ph_flag** – restore phase slope flag (0:no phase change default=1:add back phase ramp)
- **loff** – offset to starting line (default = 0)
- **nlines** – number of lines to calculate (enter - for default: to end of file)
- **SLC2R_par** – (input) parameter file of resampled SLC, required if SLC-2 frequency differs from SLC-1
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.image_stat` (*image*, *width*, *roff*, *loff*, *nr*, *nl*, *report*, *logpath=None*,
outdir=None, *shellscript=None*)

Calculate mean, standard deviation and number of non-zero values for a rectangular image region (float format)
Copyright 2016, Gamma Remote Sensing, v1.3 3-Nov-2016

Parameters

- **image** – (input) image data file (float)
- **width** – image line width (samples/line)
- **roff** – sample offset to region start (enter - for default: 0)
- **loff** – line offset to region start (enter - for default: 0)
- **nr** – region width (samples, enter - for default: width - roff)
- **nl** – number of lines in the region (enter - for default: image_lines - loff)
- **report** –
output text file (keyword:value format) keywords: file, mean, stdev, total_samples,
non_zero_samples, fraction_valid)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.init_offset` (*SLC_1*, *SLC_2*, *SLC1_par*, *SLC2_par*, *OFF_par*,
rlks='-', *azlks='-'*, *rpos='-'*, *azpos='-'*, *offr='-'*,
offaz='-', *thres='-'*, *rwin='-'*, *azwin='-'*, *cflag='-'*,
logpath=None, *outdir=None*, *shellscript=None*)

Determine initial offset between SLC images using correlation of image intensity
Copyright 2016, Gamma Remote Sensing, v3.1 clw 12-Apr-2016

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)
- **SLC_2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **rlks** – number of range looks (default: 1)
- **azlks** – number of azimuth looks (default: 1)
- **rpos** – center of patch in range (samples) (enter - for default: image center)
- **azpos** – center of patch in azimuth (lines) (enter - for default: image center)
- **offr** – initial range offset (samples) (enter - for default: 0)
- **offaz** – initial azimuth offset (lines) (enter - for default: 0)
- **thres** – cross-correlation threshold (enter - for default: 0.150)
- **rwin** – range window size (default: 512)
- **azwin** – azimuth window size (default: 512)
- **cflag** –
 - copy offsets to the range and azimuth offset polynomials in the OFF_par**
 - 0: do not copy
 - 1: copy constant range and azimuth offset (default)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.init_offset_orbit(SLC1_par, SLC2_par, OFF_par, rpos='-',
                                             azpos='-', cflag='-', logpath=None, outdir=None, shellscript=None)
```

Initial SLC image offset estimation from orbit state-vectors and image parameters

Copyright 2016, Gamma Remote Sensing, v1.7 21-Apr-2016 clw/uw

Parameters

- **SLC1_par** – (input) SLC-1 parameter file
- **SLC2_par** – (input) SLC-2 parameter file
- **OFF_par** – (input/output) ISP/offset parameter file
- **rpos** – range position for offset estimation (enter - for default: center of SLC-1)
- **azpos** – azimuth position for offset estimation (enter - for default: center of SLC-1)
- **cflag** –
 - copy offsets to the range and azimuth offset polynomials in the OFF_par**
 - 0: do not copy

- 1: copy constant range and azimuth offset (default)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.interf_SLC(SLC_1, SLC_2, SLC1_par, SLC2_par, OFF_par,  
                                       MLI_1, MLI_2, interf, nrlk='-', nazlk='-', loff='-',  
                                       nltot='-', rfilt='-', azfilt='-', s_off='-', logpath=None,  
                                       outdir=None, shellscript=None)
```

Interferogram generation using a pair of SLC images

Copyright 2009, Gamma Remote Sensing, v4.10 clw/uw 27-Oct-2009

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)
- **SLC_2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **MLI_1** – (output) multi-look intensity image 1
- **MLI_2** – (output) multi-look intensity image 2
- **interf** – interferogram from SLC-1 and SLC-2
- **nrlk** – number of interferogram range looks (default: 2)
- **nazlk** – number of interferogram azimuth looks (default: 10)
- **loff** – offset to starting line of interferogram (relative to start of SLC-1) (default: 0)
- **nltot** – number of SLC lines to process (default: 0, to end of file)
- **rfilt** –
 range common band filtering flag
 - 0: OFF
 - 1: ON (default)
- **azfilt** –
 azimuth common band filtering flag
 - 0: OFF
 - 1: ON (default)
- **s_off** –
 offset to the nominal range spectral shift (frac. of range sampling freq.) (default: 0.0)
 - NOTE: enter - as filename to avoid creation of corresponding output file
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in

- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.interp_ad(data_in, data_out, width, r_max='-', np_min='-',
                                     np_max='-', w_mode='-', type='-', cp_data='-', log-
                                     path=None, outdir=None, shellscript=None)
```

Weighted interpolation of gaps in 2D data using an adaptive smoothing window

Copyright 2018, Gamma Remote Sensing, v2.1 13-Jun-2018 clw/uw

Parameters

- **data_in** – (input) data with gaps
- **data_out** – (output) data with gaps filled by interpolation
- **width** – number of samples/row
- **r_max** – maximum interpolation window radius (default(-): 16)
- **np_min** – minimum number of points used for the interpolation (default(-): 16)
- **np_max** – maximum number of points used for the interpolation (default(-): 16)
- **w_mode** –
data weighting mode (enter - for default):
 - 0: constant
 - 1: $1 - (r/r_max)$
 - 2: $1 - (r/r_max)**2$ (default)
 - 3: $\exp(-2. * (r**2/r_max**2))$
- **type** –
input and output data type:
 - 0: FCOMPLEX
 - 1: SCOMPLEX
 - 2: FLOAT (default)
 - 3: INT
 - 4: SHORT
- **cp_data** –
copy data flag:
 - 0: do not copy input data values to output
 - 1: copy input data values to output (default)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ionosphere_check(SLC, par, rwin='-', azwin='-', thresh='-',
                                             rstep='-', azstep='-', logpath=None, out-
                                             dir=None, shellscript=None)
```

Parameters

- **SLC** – (input) SLC image (e.g. 20070214.slc)
- **par** – (input) SLC parameter file (e.g. 20070214.slc.par)
- **rwin** – range window size used in offset estimation (default = 256)
- **azwin** – azimuth window size used in offset estimation (default = 256)
- **thresh** – threshold value used in offset estimation (default = 0.1)
- **rstep** – range step used in offset estimation (default = rwin/4)
- **azstep** – azimuth step used in offset estimation (default = azwin/4)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.make_tab(list, tab, definition, logpath=None, outdir=None,
                                   shellscript=None)
```

Parameters

- **list** – (input) list file (ascii)
- **tab** – (output) table file (ascii)
- **definition** –
definition used to generate line of output table (example 1: '\$1.slc \$1.slc.par') (example 2: '\$1_\$2.base \$1_\$2.off')
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.mask_data(data_in, width, data_out, mask, dtype='-', log-
                                   path=None, outdir=None, shellscript=None)
```

Mask float or fcomplex data using an 8-bit SUN/BMP/TIFF format raster image

Copyright 2016, Gamma Remote Sensing, v1.4 10-Dec-2016 clw

Parameters

- **data_in** – (input) data file (FLOAT or FCOMPLEX format)
- **width** – width of input data file
- **data_out** – (output) data file, same data format as input
- **mask** –
(input) mask file, SUN/BMP/TIFF raster format, 8-bits/pixel output data values are set to 0.0 at all locations where the mask is black (0,0,0) or dn = 0 * NOTE: mask file must have the same width as the input data file

- **dtype** –
data format:
 - 0: FLOAT (default)
 - 1: FCOMPLEX
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.mcf(interf, wgt, mask, unw, width, tri_mode='-', roff='-', loff='-',
                               nr='-', nlines='-', npat_r='-', npat_az='-', overlap='-',
                               r_init='-', az_init='-', init_flag='-', logpath=None, outdir=None,
                               shellscript=None)
```

Phase unwrapping using Minimum Cost Flow (MCF) and triangulation
 Copyright 2016, Gamma Remote Sensing, v2.2 clw/uw 30-Nov-2016

Parameters

- **interf** – (input) interferogram (*.int,*.flt)(fcomplex)
- **wgt** – (input) weight factors (0 -> 1.0) file (float)(enter - for uniform weight)
- **mask** – (input) validity mask (SUN/BMP/TIFF raster format, value 0 -> pixel not used)(enter - if no mask)
- **unw** – (output) unwrapped phase image (*.unw)(float)
- **width** – number of samples/row
- **tri_mode** –
triangulation mode
 - 0: filled triangular mesh (default)
 - 1: Delaunay triangulation
- **roff** – offset to starting range of section to unwrap (default: 0)
- **loff** – offset to starting line of section to unwrap (default: 0)
- **nr** – number of range samples of section to unwrap (default(-): width - roff)
- **nlines** – number of lines of section to unwrap (default(-): total number of lines - loff)
- **npat_r** – number of patches in range
- **npat_az** – number of patches in azimuth
- **overlap** – overlap between patches in pixels (overlap >= 7, default(-): 512)
- **r_init** – phase reference point range offset (default(-): roff)
- **az_init** – phase reference point azimuth offset (default(-): loff)
- **init_flag** –
flag to set phase at reference point
 - 0: use initial point phase value (default)
 - 1: set phase to 0.0 at initial point

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.mk_ptarg(RSLC_tab, cal_dir, r_samp, az_samp, osf='-', log-  
                                     path=None, outdir=None, shellscript=None)
```

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/mk_ptarg

Copyright 2012, Gamma Remote Sensing, v1.5 24-Apr-2012 clw

Perform point target analysis on a stack of coregistered SLCs

Parameters

- **RSLC_tab** –

(input) two column list of coregistered SLC filenames and SLC parameter filenames (including paths) (ascii)

1. SLC filename (includes path)
2. SLC parameter filename (includes path)

- **cal_dir** – directory for output calibration results
- **r_samp** – (input) calibration target range sample number
- **az_samp** – (input) calibration target azimuth line number
- **osf** –

SLC over-sampling factor 2, 4, 8, 16, 32, 64 (default: 16)

- **-s scale** (option) set image display scale factor (default: 0.3)
- **-e exp** (option) set image display exponent (default: 0.5)

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.mk_ptarg_cal(CR_tab, SLC, SLC_par, cal_dir, sigma, c_rpos,  
                                         c_azpos, osf='-', logpath=None, outdir=None,  
                                         shellscript=None)
```

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/mk_ptarg_cal

Copyright 2016, Gamma Remote Sensing, v1.8 19-Feb-2016 clw

Perform point target analysis and calibration factor evaluation for a set of point targets

Parameters

- **CR_tab** –

(input) 3 column list of row and sample number of corner reflectors

1. Corner reflector id
2. SLC column (includes path)
3. SLC row (includes path)

- **SLC** – SLC image

- **SLC_par** – SLC_parameter file
- **cal_dir** – directory for output calibration results
- **sigma** – Radar cross-section of the corner reflectors
- **c_rpos** – range sample number of the center of the region used to estimate region
- **c_azpos** – azimuth line of the center of the region used to estimate clutter
- **osf** –
SLC over-sampling factor 2, 4, 8, 16, 32, 64 (default: 16)
- **-s scale** (option) set image display scale factor (default: 0.2)
- **-e exp** (option) set image display exponent (default: 0.5)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.mk_tab3` (*dir*, *ext_1*, *ext_2*, *ext_3*, *tab*, *logpath=None*, *outdir=None*, *shellscript=None*)

Copyright 2014, Gamma Remote Sensing, v1.0 27-Jun-2014 clw
Generate SLC_tab, MLI_tab, or RAW_list for processing

Parameters

- **dir** – (input) directory including paths that contain the data files
- **ext_1** – (input) pattern to select data files (examples: slc, raw...), (enter - for all files in the directory)
- **ext_2** – (input) pattern to select parameter files that match the data (enter - for none, examples: slc.par, raw_par, raw.par)
- **ext_3** – (input) pattern to select parameter files that match the data (enter - for none, examples: ppar)
- **tab** –
(output) list of data filenames and associated parameter files (including paths) (text)
– NOTE: The current directory is denoted using .
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.mosaic_WB` (*data_tab*, *dtype*, *data_out*, *data_par_out*, *sc_flg='-'*, *log-path=None*, *outdir=None*, *shellscript=None*)

ISP: Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/mosaic_WB.c
Copyright 2018, Gamma Remote Sensing, v1.3 26-Apr-2018 clw/cm
Mosaic Wide-Beam ScanSAR data processed by the MSP

Parameters

- **data_tab** – (input) 2 column list of data and ISP image parameter files for the beams in the mosaic (text)
- **dtype** –
(input) input data type:
 - 0: FLOAT
 - 1: FCOMPLEX
- **data_out** – (output) output image mosaic
- **data_par_out** – (output) ISP image parameter file for output image mosaic
- **sc_flg** –
intensity scaling flag:
 - 0: do not scale different beam data values
 - 1: use overlap regions to scale beam intensities (default)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.multi_S1_TOPS (SLC_tab, MLI, MLI_par, rlks, azlks, wflg='-'  
                                          , SLCR_tab='-', logpath=None, outdir=None,  
                                          shellscript=None)
```

Calculate MLI mosaic from Sentinel-1 TOPS SLC burst data (FCOMPLEX and SCOMPLEX)

Copyright 2018, Gamma Remote Sensing v3.4 26-Apr-2018 awi/clw/uw/cm

Parameters

- **SLC_tab** – (input) 3 column list of SLC, SLC_par, Sentinel-1 TOPS_par, rows sorted in the order IW1, IW2, IW3
- **MLI** – (output) multi-look intensity image
- **MLI_par** – (output) MLI image parameter file
- **rlks** – number of range looks
- **azlks** – number of azimuth looks
- **wflg** –
burst window calculation flag:
 - 0: use existing burst window parameters if they exist, otherwise calculate burst window parameters (default)
 - 1: calculate burst window parameters from burst parameters and the number of range and azimuth looks
- **SLCR_tab** –
(input) SLC_tab of the reference scene, 3 column list of SLC, SLC_par, TOPS_par sorted in the order IW1,
 - NOTE: When generating an MLI mosaic of a resampled SLC, the SLC_tab of the reference scene is required

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.multi_SLC_WSS (SLC, SLC_par, MLI, MLI_par, logpath=None, outdir=None, shellscript=None)
```

Calculate multi-look intensity image (MLI) from a ASAR Wide-Swath SLC

Copyright 2008, Gamma Remote Sensing v1.2 08-Jan-2008 clw/awi

Parameters

- **SLC** – (input) ASAR Wide-Swath SLC image
- **SLC_par** – (input) ASAR Wide-Swath SLC image parameter file
- **MLI** – (output) multi-look intensity image
- **MLI_par** – (output) MLI image parameter file
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.multi_cpx (data_in, OFF_par_in, data_out, OFF_par_out, rlks='-',  
                                         azlks='-', loff='-', nlines='-', roff='-', nsamp='-', logpath=None, outdir=None, shellscript=None)
```

Calculate multi-look averaged or interpolated 2D image (fcomplex data)

Copyright 2013, Gamma Remote Sensing, v2.5 28-Mar-2013 clw/uw

Parameters

- **data_in** – (input) input fcomplex image file
- **OFF_par_in** – (input) offset parameter file for input image
- **data_out** – (output) output multi-look or interpolated fcomplex data file
- **OFF_par_out** – (input/output) offset parameter file for output, if already exists, then used as input
- **rlks** – number of range looks, values < -1, interpreted as an image oversampling factor (default: 1)
- **azlks** – number azimuth looks, values < -1, interpreted as an image oversampling factor (default: 1)
- **loff** – line offset to starting line (default: 0)
- **nlines** – number of lines (default: 0, to end of file)
- **roff** – offset to starting range sample (default:0)
- **nsamp** – number of range samples to extract (default: 0, to end of line)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.multi_look(SLC, SLC_par, MLI, MLI_par, rlks, azlks, loff='-'  
                                         ', nlines=-', scale=-', exp=-', logpath=None, out-  
                                         dir=None, shellscript=None)
```

Calculate a multi-look intensity (MLI) image from an SLC image

Copyright 2018, Gamma Remote Sensing, v4.4 12-Jan-2018 clw/uw/cm

Parameters

- **SLC** – (input) single-look complex image
- **SLC_par** – (input) SLC ISP image parameter file
- **MLI** – (output) multi-look intensity image
- **MLI_par** – (output) MLI ISP image parameter file
- **rlks** – number of range looks
- **azlks** – number of azimuth looks
- **loff** – offset to starting line (enter - for default: 0)
- **nlines** – number of SLC lines to process (enter - for default: entire file)
- **scale** – scale factor for output MLI (enter - for default: 1.0)
- **exp** – exponent for the output MLI (enter - for default: 1.0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.multi_look2(SLC, SLC_par, MLI, MLI_par, r_dec, az_dec,  
                                         r_enl=-', az_enl=-', lanczos=-', scale=-', exp=-'  
                                         ', logpath=None, outdir=None, shellscript=None)
```

Calculate an MLI image from an SLC with separate averaging window dimensions and decimation factors

Copyright 2017, Gamma Remote Sensing, v1.0 3-Jan-2018 clw/cm

Parameters

- **SLC** – (input) single-look complex image
- **SLC_par** – (input) SLC ISP image parameter file
- **MLI** – (output) multi-look intensity image
- **MLI_par** – (output) MLI ISP image parameter file
- **r_dec** – range decimation factor (integer)
- **az_dec** – azimuth decimation factor (integer)
- **r_enl** – number of SLC range samples to average, (integer) (enter - for default: r_dec)
- **az_enl** – number of SLC azimuth lines to average, (integer) (enter - for default: az_dec)
- **lanczos** – Lanczos interpolator order 5 -> 9 (enter - for default: 7)
- **scale** – scale factor for output MLI (enter - for default: 1.0)
- **exp** – exponent for the output MLI (enter - for default: 1.0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to

- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.multi_look_MLI (MLI_in, MLI_in_par, MLI_out, MLI_out_par,
                                             rlks, azlks, loff='-', nlines='-', scale='-', log-
                                             path=None, outdir=None, shellscript=None)
```

Multi-looking of intensity (MLI) images

Copyright 2017, Gamma Remote Sensing, v1.7 31-Aug-2017 clw/uw/cm

Parameters

- **MLI_in** – (input) multi-look intensity image (MLI) file (float)
- **MLI_in_par** – (input) MLI parameter file
- **MLI_out** – (output) multi-looked MLI image (float)
- **MLI_out_par** – (output) MLI parameter file for output MLI
- **rlks** – range looks for multi-looking
- **azlks** – azimuth looks for multi-looking
- **loff** – offset to starting line (enter - for default = 0)
- **nlines** – number of input MLI lines to process (enter - for default = entire file)
- **scale** – scale factor for output MLI (enter - for default = 1.0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.multi_real (data_in, OFF_par_in, data_out, OFF_par_out, rlks='-',
                                         azlks='-', loff='-', nlines='-', roff='-', nsamp='-',
                                         logpath=None, outdir=None, shellscript=None)
```

Calculate multi-look averaged or interpolated 2D image (float data)

Copyright 2012, Gamma Remote Sensing, v2.5 16-Jul-2013 clw/uw

Parameters

- **data_in** – (input) input float image file
- **OFF_par_in** – (input) interferogram/offset parameter file for input image
- **data_out** – (output) output multi-look or interpolated float data file
- **OFF_par_out** – (input/output) interferogram/offset parameter file for output, if already existing, used as input
- **rlks** – number of range looks, values < -1, interpreted as an image oversampling factor (default: 1)
- **azlks** – number azimuth looks, values < -1, interpreted as an image oversampling factor (default: 1)
- **loff** – line offset to starting line (default:0)
- **nlines** – number of lines (default:0, to end of file)

- **roff** – offset to starting range sample (default:0)
- **nsamp** – number of range samples to extract (default:0, to end of line)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.neutron(intensity, flag, width, n_thres, ymin='-', ymax='-', log-  
path=None, outdir=None, shellscript=None)
```

Generate phase unwrapping neutrons using image intensity

Copyright 2014, Gamma Remote Sensing, v2.3 20-Jan-2014 clw/uw

Parameters

- **intensity** – (input) image intensity
- **flag** – (input) phase unwrapping flag file
- **width** – number of samples/row
- **n_thres** – neutron threshold, multiples of the average intensity (default=6.0)
- **ymin** – offset to starting azimuth row (default = 0)
- **ymax** – offset to last azimuth row (default = nlines-1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.offset_SLC(SLC_1, SLC_2, SLC1_par, SLC2_par, OFF_par, offs,  
snr, rwin='-', azwin='-', offsets='-', n_ovr='-', nr='-',  
'naz='-', thres='-', ISZ='-', pflag='-', logpath=None,  
outdir=None, shellscript=None)
```

Offsets between SLC images using fringe visibility

Copyright 2016, Gamma Remote Sensing, v2.9 clw 4-Mar-2016

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)
- **SLC_2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates (fcomplex)
- **snr** – (output) offset estimation SNR (float)
- **rwin** – search window size (range pixels, (enter - for default from offset parameter file))
- **azwin** – search window size (azimuth lines, (enter - for default from offset parameter file))
- **offsets** – (output) range and azimuth offsets and SNR data in text format, enter - for no output

- **n_ovr** – SLC oversampling factor (integer $2^{**}N$ (1,2,4) default = 2)
- **nr** – number of offset estimates in range direction (enter - for default from offset parameter file)
- **naz** – number of offset estimates in azimuth direction (enter - for default from offset parameter file)
- **thres** – offset estimation quality threshold (enter - for default from offset parameter file)
- **ISZ** – search chip interferogram size (in non-oversampled pixels, default=16)
- **pflag** – print flag (0:print offset summary default=1:print all offset data)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.offset_SLC_tracking(SLC_1, SLC_2, SLC1_par, SLC2_par,
                                                OFF_par, offs, snr, rsw='-', azsw='-',
                                                offsets='-', n_ovr='-', thres='-',
                                                rstep='-', azstep='-', rstart='-', rstop='-',
                                                azstart='-', azstop='-', ISZ='-',
                                                pflag='-', logpath=None, outdir=None,
                                                shellscript=None)
```

Offset tracking between SLC images using fringe visibility

Copyright 2016, Gamma Remote Sensing, v3.6 clw 4-Mar-2016

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)
- **SLC_2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates (fcomplex)
- **snr** – (output) offset estimation SNR (float)
- **rsw** – range search window size (range pixels) (enter - for default from offset parameter file)
- **azsw** – azimuth search window size (azimuth lines) (enter - for default from offset parameter file)
- **offsets** – (output) range and azimuth offsets and SNR data in text format, enter - for no output
- **n_ovr** – SLC over-sampling factor (integer $2^{**}N$ (1,2,4) default: 2)
- **thres** – offset estimation quality threshold (enter - for default from offset parameter file)
- **rstep** – step in range pixels (enter - for default: rsw/2)
- **azstep** – step in azimuth pixels (enter - for default: azsw/2)
- **rstart** – starting range pixel (enter - for default: rsw/2)

- **rstop** – ending range pixel (enter - for default: nr - rsw/2)
- **azstart** – starting azimuth line (enter - for default: azsw/2)
- **azstop** – ending azimuth line (enter - for default: nlines - azsw/2)
- **ISZ** – search chip interferogram size (in non-oversampled pixels, default: 16)
- **pflag** –
 print flag:
 - 0: print offset summary (default)
 - 1: print all offset data
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.offset_add(OFF_par1, OFF_par2, OFF_par3, logpath=None, outdir=None, shellscript=None)
```

Add range and azimuth offset polynomial coefficients

Copyright 2008, Gamma Remote Sensing, v1.1 12-Feb-2008 clw

Parameters

- **OFF_par1** – (input) ISP offset/interferogram parameter file
- **OFF_par2** – (input) ISP offset/interferogram parameter file
- **OFF_par3** –
 (output) ISP offset/interferogram parameter file with sums of the range and azimuth offset polynomials in OFF_par1 and OFF_par2
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.offset_fit(offsets, ccp, OFF_par, coeffs='-', coffsets='-', thres='-', npoly='-', interact_mode='-', logpath=None, outdir=None, shellscript=None)
```

Range and azimuth offset polynomial estimation

Copyright 2011, Gamma Remote Sensing, v3.3 28-Nov-2015 clw/uw

Parameters

- **offsets** – (input) range and azimuth offset estimates for each patch (fcomplex)
- **ccp** – (input) cross-correlation or SNR of each patch (float)
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **coeffs** – (output) culled range and azimuth offset estimates (fcomplex, enter - for none)
- **coffsets** – (output) culled offset estimates and cross-correlation values (text format, enter - for none)
- **thres** – cross-correlation threshold (enter - for default from OFF_par)

- **npoly** – number of model polynomial parameters (enter - for default, 1, 3, 4, 6, default: 4)
- **interact_mode** –
interactive culling of input data:
 - 0: off (default)
 - 1: on
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.offset_plot_az` (*offset*, *r_min*, *r_max*, *r_plot*, *az_plot*, *log-path=None*, *outdir=None*, *shellscript=None*)

IPTA script: /usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/offset_plot_az

Copyright 2004, Gamma Remote Sensing, v1.3 17-Jan-2005 clw

extract range and azimuth offsets for a range window from an text offset file

Parameters

- **offset** – (input) list of range and azimuth offsets generated by `offset_pwr` (text)
- **r_min** – minimum range pixel number to extract range and azimuth offsets
- **r_max** – minimum range pixel number to extract range and azimuth offsets
- **r_plot** – range offsets xmgrace plot file
- **az_plot** – azimuth offsets xmgrace plot file
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.offset_plot_r` (*offset*, *az_min*, *az_max*, *r_plot*, *az_plot*, *log-path=None*, *outdir=None*, *shellscript=None*)

IPTA script: /usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/offset_plot_r

Copyright 2004, Gamma Remote Sensing, v1.3 17-Jan-2005 clw

extract range and azimuth offsets for an azimuth window from an text offset file

Parameters

- **offset** – (input) list of range and azimuth offsets generated by `offset_pwr` (text)
- **az_min** – minimum azimuth line number to extract range and azimuth offsets
- **az_max** – minimum azimuth line number to extract range and azimuth offsets
- **r_plot** – range offsets xmgrace plot file
- **az_plot** – azimuth offsets xmgrace plot file
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.offset_pwr (SLC1, SLC2, SLC1_par, SLC2_par, OFF_par, offs,
                                         ccp, rwin='-', azwin='-', offsets='-', n_ovr='-',
                                         nr='-', naz='-', thres='-', lanczos='-', bw_frac='-',
                                         deramp='-', int_filt='-', pflag='-', pltflg='-', ccs='-',
                                         logpath=None, outdir=None, shellscript=None)
```

Offset estimation between SLC images using intensity cross-correlation

Copyright 2017, Gamma Remote Sensing, v5.5 clw/cm 17-Nov-2017

Parameters

- **SLC1** – (input) single-look complex image 1 (reference)
- **SLC2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates in range and azimuth (fcomplex)
- **ccp** – (output) cross-correlation of each patch (0.0->1.0) (float)
- **rwin** – range patch size (range pixels, enter - for default from offset parameter file)
- **azwin** – azimuth patch size (azimuth lines, enter - for default from offset parameter file)
- **offsets** – (output) range and azimuth offsets and cross-correlation data in text format, enter - for no output
- **n_ovr** – SLC oversampling factor (integer 2**N (1,2,4), enter - for default: 2)
- **nr** – number of offset estimates in range direction (enter - for default from offset parameter file)
- **naz** – number of offset estimates in azimuth direction (enter - for default from offset parameter file)
- **thres** – cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
- **lanczos** – Lanczos interpolator order 5 -> 9 (enter - for default: 5)
- **bw_frac** – bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for default: 1.0)
- **deramp** –
 deramp SLC phase flag (enter - for default)
 - 0: no deramp (Doppler centroid close to 0) (default)
 - 1: deramp SLC phase
- **int_filt** –
 intensity low-pass filter flag (enter - for default)
 - 0: no filter
 - 1: low-pass filter of intensity data, highly recommended when no oversampling used (default)
- **pflag** –

print flag (enter - for default)

- 0: print offset summary (default)
- 1: print all offset data

• **pltflg** –**plotting flag (enter - for default)**

- 0: none (default)
- 1: screen output
- 2: screen output and PNG format plots
- 3: output plots in PDF format

• **ccs** –**(output) cross-correlation standard deviation of each patch (float)**

- NOTE: ScanSAR and TOPS data need to be previously deramped

• **logpath** (*str* or *None*) – a directory to write command logfiles to• **outdir** (*str* or *None*) – the directory to execute the command in• **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.offset_pwr_tracking(SLC1, SLC2, SLC1_par, SLC2_par,
                                                OFF_par, offs, ccp, rwin='-', azwin='-',
                                                offsets='-', n_ovr='-', thres='-',
                                                rstep='-', azstep='-', rstart='-',
                                                rstop='-', azstart='-', azstop='-',
                                                lanczos='-', bw_frac='-', deramp='-',
                                                int_filt='-', pflag='-', pltflg='-',
                                                ccs='-', logpath=None, outdir=None,
                                                shellscript=None)
```

Offset tracking between SLC images using intensity cross-correlation
 Copyright 2017, Gamma Remote Sensing, v5.7 clw/cm 17-Nov-2017

Parameters

- **SLC1** – (input) single-look complex image 1 (reference)
- **SLC2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates in range and azimuth (fcomplex)
- **ccp** – (output) cross-correlation of each patch (0.0->1.0) (float)
- **rwin** – range patch size (range pixels, enter - for default from offset parameter file)
- **azwin** – azimuth patch size (azimuth lines, enter - for default from offset parameter file)
- **offsets** – (output) range and azimuth offsets and cross-correlation data in text format, enter - for no output

- **n_ovr** – SLC oversampling factor (integer $2 \times N$ (1,2,4), enter - for default: 2)
- **thres** – cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
- **rstep** – step in range pixels (enter - for default: rwin/2)
- **azstep** – step in azimuth pixels (enter - for default: azwin/2)
- **rstart** – offset to starting range pixel (enter - for default: 0)
- **rstop** – offset to ending range pixel (enter - for default: nr-1)
- **azstart** – offset to starting azimuth line (enter - for default: 0)
- **azstop** – offset to ending azimuth line (enter - for default: nlines-1)
- **lanczos** – Lanczos interpolator order 5 -> 9 (enter - for default: 5)
- **bw_frac** – bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for default: 1.0)
- **deramp** –
deramp SLC phase flag (enter - for default)
 - 0: no deramp (Doppler centroid close to 0) (default)
 - 1: deramp SLC phase
- **int_filt** –
intensity low-pass filter flag (enter - for default)
 - 0: no filter
 - 1: low-pass filter of intensity data, highly recommended when no oversampling used (default)
- **pflag** –
print flag (enter - for default)
 - 0: print offset summary (default)
 - 1: print all offset data
- **pltflg** –
plotting flag (enter - for default)
 - 0: none (default)
 - 1: screen output
 - 2: screen output and PNG format plots
 - 3: output plots in PDF format
- **ccs** –
(output) cross-correlation standard deviation of each patch (float)
 - NOTE: ScanSAR and TOPS data need to be previously deramped
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.offset_pwr_tracking2(SLC1, SLC2, SLC1_par, SLC2_par,
OFF_par, offs, ccp, OFF_par2='-',
offs2='-', rwin='-', azwin='-',
offsets='-', n_ovr='-', thres='-',
rstep='-', azstep='-', rstart='-',
rstop='-', azstart='-', azstop='-',
bw_frac='-', deramp='-', int_filt='-',
pflag='-', pltflg='-', ccs='-',
logpath=None, outdir=None,
shellscript=None)
```

Intensity cross-correlation offset tracking with the initial offset for each patch determined from input offset map
 Copyright 2017, Gamma Remote Sensing, v1.5 clw/cm 20-Mar-2017

Parameters

- **SLC1** – (input) single-look complex image 1 (reference)
- **SLC2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates in range and azimuth (fcomplex)
- **ccp** – (output) cross-correlation of each patch (0.0->1.0) (float)
- **OFF_par2** – (input) ISP offset/interferogram parameter file of the offset map to determine initial offsets (enter - for none)
- **offs2** – (input) input range and azimuth offset map to determine initial offsets (enter - for none)
- **rwin** – range patch size (range pixels, enter - for default from offset parameter file)
- **azwin** – azimuth patch size (azimuth lines, enter - for default from offset parameter file)
- **offsets** – (output) range and azimuth offsets and cross-correlation data in text format, enter - for no output
- **n_ovr** – SLC oversampling factor (integer 2**N (1,2,4), enter - for default: 2)
- **thres** – cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
- **rstep** – step in range pixels (enter - for default: rwin/2)
- **azstep** – step in azimuth pixels (enter - for default: azwin/2)
- **rstart** – offset to starting range pixel (enter - for default: 0)
- **rstop** – offset to ending range pixel (enter - for default: nr-1)
- **azstart** – offset to starting azimuth line (enter - for default: 0)
- **azstop** – offset to ending azimuth line (enter - for default: nlines-1)
- **bw_frac** – bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for default: 1.0)
- **deramp** –

deramp SLC phase flag (enter - for default)

- 0: no deramp (Doppler centroid close to 0) (default)
- 1: deramp SLC phase

• int_filt –**intensity low-pass filter flag (enter - for default)**

- 0: no filter
- 1: low-pass filter of intensity data, highly recommended when no oversampling used (default)

• pflag –**print flag (enter - for default)**

- 0: print offset summary (default)
- 1: print all offset data

• pltflg –**plotting flag (enter - for default)**

- 0: none (default)
- 1: screen output
- 2: screen output and PNG format plots
- 3: output plots in PDF format

• ccs –**(output) cross-correlation standard deviation of each patch (float)**

- NOTE: ScanSAR and TOPS data need to be previously deramped

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.offset_sub(offs, OFF_par, offs_sub, logpath=None, outdir=None,  
                                     shellscript=None)
```

Subtraction of polynomial from range and azimuth offset estimates

Copyright 2017, Gamma Remote Sensing, v1.0 27-Mar-2017 cm

Parameters

- **offs** – (input) range and azimuth offset estimates (fcomplex)
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs_sub** – (output) range and azimuth offset estimates after polynomial subtraction (fcomplex)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.offset_tracking(offs, ccp, SLC_par, OFF_par, disp_map,
                                           disp_val='-', mode='-', thres='-',
                                           poly_flag='-', logpath=None, outdir=None,
                                           shellscript=None)
```

Conversion of range and azimuth offsets files to displacement map

Copyright 2017, Gamma Remote Sensing, v2.0 4-Apr-2017 ts/clw/uw

Parameters

- **offs** – (input) range and azimuth offset estimates (fcomplex)
- **ccp** – (input) cross-correlation of the offset estimates (float)
- **SLC_par** – (input) SLC parameter file of reference SLC
- **OFF_par** – (input) offset parameter file used in the offset tracking
- **disp_map** – (output) range and azimuth displacement estimates (fcomplex)
- **disp_val** – (output) range and azimuth displacement estimates and cross-correlation values (enter - for none) (text)
- **mode** –
 - flag indicating displacement mode:**
 - 0: displacement in range and azimuth pixels
 - 1: displacement in meters in slant range and azimuth directions
 - 2: displacement in meters in ground range and azimuth directions (default)
- **thres** – cross-correlation threshold to accept offset value (default from OFF_par)
- **poly_flag** –
 - flag indicating if trend calculated using offset polynomials from OFF_par is subtracted:**
 - 0: do not subtract polynomial trend from offset data
 - 1: subtract polynomial trend from offset data (default)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_ACS_ERS(CEOS_SAR_leader, SLC_par, logpath=None, outdir=None, shellscript=None)
```

ISP parameter file generation for ERS SLC data from the ACS processor

Copyright 2005, Gamma Remote Sensing, v1.3 17-Oct-2005 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in

- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.par_ASAR` (*ASAR_ERS_file*, *output_name*, *K_dB*='-', *logpath*=None, *outdir*=None, *shellscript*=None)

Extract SLC/MLI image parameters and images from ENVISAT ASAR SLC, WSS, APP, and PRI products
Copyright 2018, Gamma Remote Sensing, v2.7 9-Apr-2018 clw/uw/awi

Parameters

- **ASAR_ERS_file** – (input) ASAR or ERS data in ASAR format (SAR_IMS_1P) including header and image as provided by ESA
- **output_name** – (output) common part of output file names (e.g. YYYYMMDD date)
- **K_dB** –

Calibration factor in dB (nominal value for all ASAR modes: 55.0)

- NOTE: Use - to use the calibration factor provided in the ASAR file header
- NOTE: In the case that a calibration factor is specified on the command line, PRI images are converted to radiometrically calibrated ground-range intensity images in float format

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.par_ASF_91` (*CEOS_leader*, *CEOS_trailer*, *SLC_par*, *logpath*=None, *outdir*=None, *shellscript*=None)

SLC parameter file for data data from the Alaska SAR Facility (1991-1996)
Copyright 2008, Gamma Remote Sensing, v3.3 25-Mar-2008 clw/uw

Parameters

- **CEOS_leader** – (input) ASF CEOS leader file
- **CEOS_trailer** – (input) ASF CEOS trailer file
- **SLC_par** – (output) ISP SLC image parameter file
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

`pyroSAR.gamma.parser_demo.par_ASF_96` (*CEOS_SAR_leader*, *SLC_par*, *logpath*=None, *outdir*=None, *shellscript*=None)

ISP parameter file for ASF data 1996->present v1.1
Copyright 2003, Gamma Remote Sensing, v1.4 4-Aug-2003 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_ASF_PRI (CEOS_leader, CEOS_data, GRD_par, GRD, log-
                                         path=None, outdir=None, shellscript=None)
```

ISP parameter file for ASF detected ground range images (L1) Sep 1996 -> present
 Copyright 2014, Gamma Remote Sensing, v1.3 3-Apr-2014 clw/uw

Parameters

- **CEOS_leader** – (input) CEOS leader file
- **CEOS_data** – (input) CEOS data file binary)
- **GRD_par** – (output) ISP ground range image parameter file
- **GRD** –
 (output) ISP ground range image (enter - for none, float intensity)
 – NOTE: The input data converted to intensity using the expression: $(dn/1000.0)**2$
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_ASF_RSAT_SS (CEOS_leader, CEOS_data, GRD_par,
                                             GRD, logpath=None, outdir=None,
                                             shellscript=None)
```

ISP parameter file for ASF Radarsat-1 SCANSAR images
 Copyright 2004, Gamma Remote Sensing, v1.0 27-Aug-2004 clw/uw

Parameters

- **CEOS_leader** – (input) CEOS leader file (Radarsat-1 SCANSAR)
- **CEOS_data** – (input) CEOS data file (Radarsat-1 SCANSAR)
- **GRD_par** – (output) ISP image parameter file (example <orbit>.mli.par)
- **GRD** – (output) ISP image (example <orbit>.mli) (enter - for none, short integer)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_ASF_SLC (CEOS_leader, SLC_par, CEOS_data='-', SLC='-',
                                         logpath=None, outdir=None, shellscript=None)
```

Generate SLC image parameter file and reformat data
 Copyright 2012, Gamma Remote Sensing, v1.0 27-Aug-2012 clw/uw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file

- **SLC_par** – (output) ISP SLC parameter file (example <date>.slc.par)
- **CEOS_data** – (input) CEOS data file (example: dat_01.001)
- **SLC** – (output) SLC data with file and line headers removed (example: <date>.slc)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_ATLSCI_ERS (CEOS_SAR_leader, CEOS_Image, SLC_par, log-  
path=None, outdir=None, shellscript=None)
```

ISP parameter file for ATL-SCI ERS SLC data

Copyright 2003, Gamma Remote Sensing, v2.8 24-Nov-2003 clw

Parameters

- **CEOS_SAR_leader** – (input) CEOS SAR leader file (LEA_01.001)
- **CEOS_Image** – (input) CEOS image data segment (DAT_01.001)
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_CS_SLC (HDF5, trunk, logpath=None, outdir=None,  
shellscript=None)
```

Generate ISP SLC parameter and image files for Cosmo-Skymed SCS data

Copyright 2017, Gamma Remote Sensing, v1.8 26-Sep-2017 awi/ms/cw/uw

Parameters

- **HDF5** – (input) SCS data file in HDF5 format
- **trunk** –
(output) output file name trunk used for output filenames (example: yyyymmdd ->
yyyymmdd_pol_beamid.slc yyyymmdd_pol_beamid.slc.par)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_CS_SLC_TIF (GeoTIFF, XML, trunk, logpath=None, out-  
dir=None, shellscript=None)
```

Generate ISP SLC parameter and image files for Cosmo Skymed SCS data in GeoTIFF format

Copyright 2018, Gamma Remote Sensing, v1.5 7-Feb-2018 awi/ms/clw/cm

Parameters

- **GeoTIFF** – (input) SCS data file in GeoTIFF format
- **XML** – (input) SCS meta data file in XML format

- **trunk** –
(output) output file name trunk used for output filenames (example: yyyymmdd -> yyyymmdd_pol_beamid.slc yyyymmdd_pol_beamid.slc.par)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_EORC_JERS_SLC (CEOS_SAR_leader,          SLC_par,
                                              CEOS_data='-', SLC='-', logpath=None,
                                              outdir=None, shellscript=None)
```

Reformat EORC processed JERS-1 SLC and generate the ISP parameter file
Copyright 2008, Gamma Remote Sensing, v1.4 9-Oct-2008 clw

Parameters

- **CEOS_SAR_leader** – (input) CEOS SAR leader file for JERS SLC processed by EORC
- **SLC_par** – (output) ISP image parameter file
- **CEOS_data** – (input) CEOS format SLC data (IMOP_01.DAT, enter - for none)
- **SLC** – (output) reformatted JERS SLC (example: yyyymmdd.SLC, enter - for none)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_EORC_PALSAR (CEOS_leader,          SLC_par,          CEOS_data,
                                              SLC='-',          logpath=None,          outdir=None,
                                              shellscript=None)
```

Reformat EORC PALSAR + PALSAR2 level 1.1 CEOS format SLC data and generate the ISP parameter file
Copyright 2017, Gamma Remote Sensing, v2.8 18-Dec-2017 clw

Parameters

- **CEOS_leader** – (input) CEOS leader file for PALSAR or PALSAR-2 Level 1.1 SLC data (LED...)
- **SLC_par** – (output) ISP image parameter file (example: yyyymmdd.slc.par)
- **CEOS_data** – (input) PALSAR CEOS format Level 1.1 SLC (IMG...)
- **SLC** – (output) reformatted PALSAR SLC (example: yyyymmdd.slc, enter - for none)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_ERSDAC_PALSAR (ERSDAC_SLC_par,          SLC_par,
                                              logpath=None,          outdir=None,
                                              shellscript=None)
```

Generate the ISP image parameter file from ERSDAC PALSAR level 1.1 SLC data

Copyright 2011, Gamma Remote Sensing, v1.5 31-Oct-2011 clw

Parameters

- **ERSDAC_SLC_par** – (input) ERSDAC SLC parameter file Level 1.1 (PASL11*.SLC.par)
- **SLC_par** – (output) ISP image parameter file (example: yyyyymmdd.slc.par)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_ESA_ERS (CEOS_SAR_leader, SLC_par, inlist, CEOS_DAT='-  
, SLC='- ', logpath=None, outdir=None,  
shellscript=None)
```

ISP parameter file generation for ERS SLC data from the PGS, VMP, and SPF processors

Copyright 2012, Gamma Remote Sensing, v1.4 12-Jan-2012 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example: <date>.slc.par)
- **inlist** – a list of arguments to be passed to stdin
- **CEOS_DAT** – (input) CEOS data file (example: DAT_01.001)
- **SLC** – (output) SLC data with file and line headers removed (example: <date>.slc)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_GF3_SLC (GeoTIFF, annotation_XML, SLC_par, SLC='- ', log-  
path=None, outdir=None, shellscript=None)
```

Generate SLC parameter file and SLC image from a Gaofen-3 data set in GeoTIFF format

Copyright 2018, Gamma Remote Sensing, v1.2 8-Mar-2018 cm

Parameters

- **GeoTIFF** – (input) Gaofen-3 data file in GeoTIFF format (*.tiff) (enter - for none)
- **annotation_XML** – (input) Gaofen-3 annotation file in XML format (*.meta.xml)
- **SLC_par** – (output) ISP SLC parameter file (example: yyyyymmdd.slc.par)
- **SLC** – (output) ISP SLC data file (example: yyyyymmdd.slc) (enter - for none, SLC output will not be produced)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_IECAS_SLC (aux_data, slc_Re, slc_Im, date, SLC_par, SLC, log-  
path=None, outdir=None, shellscript=None)
```

Generate SLC parameter and image files for IECAS SLC data

Copyright 2011, Gamma Remote Sensing, v1.2 24-Jan-2011

Parameters

- **aux_data** – (input) IECAS SAR auxillary data (POS*.dat)
- **slc_Re** – (input) real part of complex SLC data
- **slc_Im** – (input) imaginary part of complex SLC data
- **date** – (input) acquisition date format: YYYYMMDD (example 20110121) from aux_data filename
- **SLC_par** – (output) ISP SLC parameter file
- **SLC** – (output) SLC image
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_KC_PALSAR_slr (factor_m, CEOS_leader, SLC_par, pol,
                                             pls_mode, KC_data, pwr, fdtab='-
                                             ', logpath=None, outdir=None,
                                             shellscript=None)
```

Generate ISP parameter file, Doppler table, and images for PALSAR KC Slant-Range data

Copyright 2013, Gamma Remote Sensing, v1.9.1 20-Aug-2013 ms,awi,clw

Parameters

- **factor_m** – (input) PALSAR Kyoto-Carbon parameter file
- **CEOS_leader** – (input) PALSAR Kyoto-Carbon leader file (LED)
- **SLC_par** – (output) ISP image parameter file (example: yyyyymmdd.mli.par)
- **pol** – polarization e.g. HH or HV
- **pls_mode** –
PALSAR acquisition mode:
 - 1: Fine Beam Single
 - 2: Fine Beam Double
 - 3: Wide Beam
- **KC_data** – (input) PALSAR Kyoto-Carbon data (short, little endian, amplitude)
- **pwr** – (output) PALSAR intensity (float, GAMMA Software endianness)
- **fdtab** – (output) table of output polynomials, one polynomial/block used as input to gc_map_fd
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_KS_DGM(HDF5, trunk, logpath=None, outdir=None,  
                                         shellscript=None)
```

Generate ISP SLC parameter and PRI image files for Kompsat DGM data

Copyright 2018, Gamma Remote Sensing, v1.1 7-Feb-2018 awi/cm

Parameters

- **HDF5** – (input) DGM data file in HDF5 format
- **trunk** –
(output) output file name trunk used for output filenames (example: yyyymmdd -> yyyymmdd_pol_beamid.slc yyyymmdd_pol_beamid.pri.par)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_KS_SLC(HDF5, trunk, logpath=None, outdir=None,  
                                         shellscript=None)
```

Generate ISP SLC parameter and image files for Kompsat SCS data

Copyright 2018, Gamma Remote Sensing, v1.5 7-Feb-2018 awi/clw/cm

Parameters

- **HDF5** – (input) SCS data file in HDF5 format
- **trunk** –
(output) output file name trunk used for output filenames (example: yyyymmdd -> yyyymmdd_pol_beamid.slc yyyymmdd_pol_beamid.slc.par)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_MSP(SAR_par, PROC_par, SLC_MLI_par, image_format='-',  
                                     logpath=None, outdir=None, shellscript=None)
```

ISP image parameter file from MSP processing parameter and sensor files

Copyright 2010, Gamma Remote Sensing, v3.3 2-Oct-2010 clw/uw

Parameters

- **SAR_par** – (input) MSP SAR sensor parameter file
- **PROC_par** – (input) MSP processing parameter file
- **SLC_MLI_par** – (output) ISP SLC/MLI image parameter file
- **image_format** –
image format flag (default: from MSP processing parameter file)
 - 0: fcomplex (pairs of 4-byte float)
 - 1: scomplex (pairs of 2-byte short integer)

– 2: float (4-bytes/value)

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_PRI (CEOS_SAR_leader, PRI_par, CEOS_DAT, PRI, log-
                                     path=None, outdir=None, shellscript=None)
```

ISP parameter file generation for ERS PRI data from the PGS and VMP processors

Copyright 2012, Gamma Remote Sensing, v1.6 12-Jan-2012 clw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file for PRI product
- **PRI_par** – (output) ISP image parameter file (example: <yyyymmdd>.pri.par)
- **CEOS_DAT** – (input) CEOS data file (example: DAT_01.001)
- **PRI** – (output) PRI data with file and line headers removed (example: <yyyymmdd>.pri)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_PRI_ESRIN_JERS (CEOS_SAR_leader, PRI_par, CEOS_DAT,
                                                PRI, logpath=None, outdir=None,
                                                shellscript=None)
```

ISP GRD parameter file for ESRIN processed JERS PRI data

Copyright 2008, Gamma Remote Sensing, v1.8 16-May-2008 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file for PRI product
- **PRI_par** – (output) ISP image parameter file (example: <yyyymmdd>.pri.par)
- **CEOS_DAT** – (input) CEOS data file (example: DAT_01.001)
- **PRI** – (output) PRI data with file and line headers removed (example: <yyyymmdd>.pri)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_PulsAR (CEOS_SAR_leader, SLC_par, logpath=None, out-
                                       dir=None, shellscript=None)
```

ISP parameter file generation for ERS SLC data from the PULSAR SAR processor

Copyright 2003, Gamma Remote Sensing, v1.2 4-Aug-2003 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_RISAT_GRD (CEOS_leader, BAND_META, GRD_par,  
                                           CEOS_image, GRD='-', line_dir='-', pix_dir='-  
                                           ', cal_flg='-', KdB='-', logpath=None, out-  
                                           dir=None, shellscript=None)
```

Read RISAT-1 Ground-Range data from a CEOS data set and perform radiometric calibration

Copyright 2015, Gamma Remote Sensing, v1.2 24-Feb-2015 clw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file (example: lea_01.001)
- **BAND_META** – (input) BAND_META.txt, additional RISAT system parameters for the scene (format keyword=value)
- **GRD_par** – (output) ISP GRD parameter file (example: YYYYMMDD.grd.par)
- **CEOS_image** – (input) CEOS Ground-Range image file (example: dat_01.001)
- **GRD** – (output) Ground-Range data with file and line headers removed (enter - for none: example: YYYYMMDD.grd)
- **line_dir** –
 set output image line direction (enter - for default):
 - 0: used value derived from CEOS leader file
 - 1: retain input data line direction (default)
 - -1: reverse input data line direction
- **pix_dir** –
 set output pixel direction (enter - for default):
 - 0: used value derived from CEOS leader file
 - 1: retain input data pixel direction (default)
 - -1: reverse input data pixel direction
- **cal_flg** –
 calibration flag (enter - for default):
 - 0: do not apply radiometric calibration
 - 1: apply radiometric calibration including KdB and incidence angle correction (default)
- **KdB** – calibration constant (dB) (enter - to use value in the CEOS leader)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format


```
pyroSAR.gamma.parser_demo.par_RISAT_SLC(CEOS_leader, BAND_META, SLC_par,
CEOS_image, SLC='-', line_dir='-', pix_dir='-',
cal_flg='-', KdB='-', logpath=None, outdir=None, shellscript=None)
```

Read RISAT-1 CEOS format SLC data and perform radiometric calibration

Copyright 2013, Gamma Remote Sensing, v1.1 3-Jun-2013 clw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file (example: lea_01.001)
- **BAND_META** – (input) BAND_META.txt, additional RISAT system parameters for the scene (format keyword=value)
- **SLC_par** – (output) ISP SLC image parameter file (example: YYYYMMDD.grd.par)
- **CEOS_image** – (input) CEOS SLC image file (example: dat_01.001)
- **SLC** – (output) SLC data with file and line headers removed (enter - for none: example: YYYYMMDD.grd)
- **line_dir** –
set output image line direction (enter - for default):
 - 0: used value derived from CEOS leader file
 - 1: retain input data line direction (default)
 - -1: reverse input data line direction
- **pix_dir** –
set output pixel direction (enter - for default):
 - 0: used value derived from CEOS leader file
 - 1: retain input data pixel direction (default)
 - -1: reverse input data pixel direction
- **cal_flg** –
calibration flag (enter - for default):
 - 0: do not apply radiometric calibration
 - 1: apply radiometric calibration including KdB and incidence angle correction (default)
- **KdB** – calibration constant (dB) (enter - to use value in the CEOS leader)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_RSAT2_SG(product_XML, lut_XML, GeoTIFF, polarization,
GRD_par, GRD, logpath=None, outdir=None,
shellscript=None)
```

Generate SLC parameter and ground range image files for Radarsat 2 SGF/SGX data

Copyright 2018, Gamma Remote Sensing, v1.9 7-Feb-2018 awi/cw/cm

Parameters

- **product_XML** – (input) Radarsat-2 product annotation XML file (product.xml)
- **lut_XML** – (input) Radarsat-2 calibration XML file (lutSigma.xml), use - for no calibration
- **GeoTIFF** – (input) image data file in GeoTIFF format (imagery_PP.tif)
- **polarization** – (input) image polarization: HH, VV, HV, VH
- **GRD_par** – (output) ISP GRD parameter file (example: yyyyymmdd_PP.grd.par)
- **GRD** – (output) float GRD data file (example: yyyyymmdd_pp.grd)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_RSAT2_SLC(product_XML, lut_XML, GeoTIFF, polarization,  
                                         SLC_par, SLC, logpath=None, outdir=None,  
                                         shellscript=None)
```

Generate SLC parameter and image files for Radarsat 2 SLC data from GeoTIFF

Copyright 2018, Gamma Remote Sensing, v2.6 7-Feb-2018 awi/clw/cm

Parameters

- **product_XML** – (input) Radarsat-2 product annotation XML file (product.xml)
- **lut_XML** – (input) Radarsat-2 calibration XML file (lutSigma.xml), use - for no calibration
- **GeoTIFF** – (input) image data file in GeoTIFF format (imagery_PP.tif)
- **polarization** – (input) image polarization: HH, VV, HV, VH
- **SLC_par** – (output) ISP SLC parameter file (example: yyyyymmdd_pp.slc.par)
- **SLC** – (output) SLC data file (example: yyyyymmdd_pp.slc)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_RSAT_SCW(CEOS_leader, CEOS_trailer, CEOS_data,  
                                         GRD_par, GRD, sc_dB='-', dt='-', logpath=None,  
                                         outdir=None, shellscript=None)
```

ISP parameter file for SCANSAR Wide Swath Data

Copyright 2012, Gamma Remote Sensing, v2.0 14-Feb-2012 clw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file
- **CEOS_trailer** – (input) CEOS SAR trailer file
- **CEOS_data** – (input) CEOS data file binary
- **GRD_par** – (output) ISP ground range image parameter file (example <orbit>.mli.par)
- **GRD** – (output) ISP ground range image (example <orbit>.mli) (enter - for none, float)
- **sc_dB** – intensity scale factor in dB (enter - for default: 0.00)

- **dt** – azimuth image time offset (s) (enter - for default = 0.0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_RSAT_SGF (CEOS_leader, CEOS_data, GRD_par, GRD,
                                         sc_dB='-', dt='-', logpath=None, outdir=None,
                                         shellscript=None)
```

ISP parameter file for RSI/Atlantis Radarsat SGF (ground range) and SCANSAR SCW16 data

Copyright 2012, Gamma Remote Sensing, v2.2 14-Feb-2012 clw

Parameters

- **CEOS_leader** – (input) CEOS leader file (RSI SGF or SCW16 products, LEA_01.001)
- **CEOS_data** – (input) CEOS data file (RSI SGF or SCW16 products, DAT_01.001)
- **GRD_par** – (output) ISP ground range image parameter file (example <orbit>.mli.par)
- **GRD** – (output) ISP ground range image (example <orbit>.grd.par) (enter - for none, float)
- **sc_dB** – intensity scale factor in dB (enter - for default: 0.00)
- **dt** – azimuth image time offset (s) (enter - for default = 0.0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_RSAT_SLC (CEOS_leader, SLC_par, CEOS_data, SLC='-',
                                         sc_dB='-', dt='-', logpath=None, outdir=None,
                                         shellscript=None)
```

ISP parameter file for RSI/Atlantis/ASF processed Radarsat SLC data

Copyright 2012, Gamma Remote Sensing, v4.0 5-Sep-2012 clw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file (example: lea_01.001)
- **SLC_par** – (output) ISP SLC parameter file (example: <date>.slc.par)
- **CEOS_data** – (input) CEOS data file (example: dat_01.001)
- **SLC** – (output) SLC data with file and line headers removed (example: <date>.slc)
- **sc_dB** – intensity scale factor in dB (enter - for default: 60.00)
- **dt** – azimuth image time offset (s) (enter - for default = 0.0)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_RSI_ERS (CEOS_SAR_leader, SLC_par, logpath=None, out-
                                         dir=None, shellscript=None)
```

ISP parameter file for RSI processed ERS SLC data

Copyright 2003, Gamma Remote Sensing, v1.7 4-Aug-2003 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_S1_GRD (GeoTIFF, annotation_XML, calibration_XML,  
                                         noise_XML, MLI_par, MLI, GRD_par='-', GRD='-'  
                                         , eflag='-', rps='-', noise_pwr='-', logpath=None,  
                                         outdir=None, shellscript=None)
```

Generate MLI and GRD images and parameter files from a Sentinel-1 GRD product

Copyright 2018, Gamma Remote Sensing, v3.2 30-Apr-2018 awi/clw/ts/cm

Parameters

- **GeoTIFF** – (input) image data file in GeoTIFF format (enter - for none, *.tiff)
- **annotation_XML** – (input) Sentinel-1 L1 XML annotation file
- **calibration_XML** – (input) Sentinel-1 L1 radiometric calibration XML file (enter - for no radiometric calibration)
- **noise_XML** – (input) Sentinel-1 L1 noise XML file (enter - to not subtract thermal noise power level)
- **MLI_par** – (output) MLI parameter file (example: yyyyymmdd_pp.mli.par)
- **MLI** – (output) MLI data file in slant range geometry (example: yyyyymmdd_pp.mli, enter - for none)
- **GRD_par** – (output) GRD parameter file (example: yyyyymmdd_pp.grd.par, enter - for none)
- **GRD** – (output) GRD data file (example: yyyyymmdd_pp.grd, enter - for none)
- **eflg** –

GR-SR grid extrapolation flag:

- 0: no extrapolation of the GR-SR grid beyond the grid boundaries
- 1: permit extrapolation of the GR-SR grid to cover the entire image (default)
- NOTE: extrapolation of the GR-SR grid may introduce geocoding errors
- **rps** – slant range pixel spacing (m) (enter - for default: calculated from ground-range parameters)
- **noise_pwr** –
noise intensity for each MLI sample in slant range using data from noise_XML
 - NOTE: when the noise_pwr file is specified, noise power correction will NOT be applied to the MLI data values

- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_S1_SLC (GeoTIFF, annotation_XML, calibration_XML,
                                         noise_XML, SLC_par, SLC, TOPS_par='- ', dtype='- ',
                                         sc_dB='- ', noise_pwr='- ', logpath=None, out-
                                         dir=None, shellscript=None)
```

Generate SLC parameter and image files for Sentinel-1 SLC data

Copyright 2018, Gamma Remote Sensing, v4.0 30-Apr-2018 awi/clw/cm

Parameters

- **GeoTIFF** – (input) image data file in GeoTIFF format (enter - for none, *.tiff)
- **annotation_XML** – (input) Sentinel-1 L1 XML annotation file
- **calibration_XML** – (input) Sentinel-1 L1 radiometric calibration XML file (enter - for no radiometric calibration)
- **noise_XML** – (input) Sentinel-1 L1 noise XML file (enter - to not subtract thermal noise power level)
- **SLC_par** – (output) ISP SLC parameter file (example: yyyyymmdd_iw1_vv.slc.par)
- **SLC** – (output) SLC data file (enter - for none, example: yyyyymmdd_iw1_vv.slc)
- **TOPS_par** – (output) SLC burst annotation file, TOPS and EW SLC data only (enter - for none, example: yyyyymmdd_iw1_vv.tops_par)
- **dtype** –
output data type:
 - 0: FCOMPLEX (default)
 - 1: SCOMPLEX
- **sc_dB** – scale factor for FCOMPLEX -> SCOMPLEX, (enter - for default: HH,VV (dB): 60.0000, VH,HV: 70.0000)
- **noise_pwr** –
noise intensity for each SLC sample in slant range using data from noise_XML
 - NOTE: when the noise_pwr file is specified, noise power will NOT be subtracted from the image data values
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_SIRC (CEOS_leader, SLC_par, UTC_MET='- ', logpath=None,
                                       outdir=None, shellscript=None)
```

ISP SLC parameter file from SIR-C CEOS leader file

Copyright 2009, Gamma Remote Sensing, v2.5 30-Oct-2009 clw/uw

Parameters

- **CEOS_leader** – (input) JPL SIR-C CEOS leader file
- **SLC_par** – (output) ISP SLC parameter file
- **UTC_MET** –
 time reference for state vectors: MET (Mission Elapsed Time) or UTC (default=UTC)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_TX_GRD (annotation_XML, GeoTIFF, GRD_par, GRD, pol='-',  
                                         logpath=None, outdir=None, shellscript=None)
```

Generate ground range image and image parameter file for Terrasar-X MGD data in GeoTIFF format
Copyright 2014, Gamma Remote Sensing, v1.3 17-Oct awi/clw

Parameters

- **annotation_XML** – (input) Terrasar-X product annotation XML file
- **GeoTIFF** –
 (input) image data file in geotiff format
 – NOTE: make sure the data set contains the selected polarisation)
- **GRD_par** – ISP ground range image parameter file (example: `yyyymmdd.grd.par`)
- **GRD** – (output) calibrated ground range data file (example: `yyyymmdd.grd`)
- **pol** – polarisation: HH, HV, VH, VV (default: first polarisation found in the `annotation_XML`)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_TX_SLC (annotation_XML, COSAR, SLC_par, SLC, pol='-', log-  
                                         path=None, outdir=None, shellscript=None)
```

Generate SLC parameter file and SLC image from a Terrasar-X SSC data set
Copyright 2017, Gamma Remote Sensing, v2.3 7-Mar-2017 awi/clw

Parameters

- **annotation_XML** – (input) TerraSAR-X product annotation XML file
- **COSAR** – (input) COSAR SSC stripmap or spotlight mode SLC data file
- **SLC_par** – (output) ISP SLC parameter file (example: `yyyymmdd.slc.par`)
- **SLC** – (output) SLC data file, example: `yyyymmdd.slc` (enter - for none, SLC output will not be produced)
- **pol** – polarisation HH, HV, VH, VV (default: first polarisation found in the `annotation_XML`)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in

- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_TX_ScanSAR(annot_XML, swath, SLC_par, SLC, TOPS_par,
                                           bwflg='-', logpath=None, outdir=None,
                                           shellscript=None)
```

Generate SLC, SLC_par and TOPS_par from a Terrasar-X ScanSAR data set

Copyright 2018, Gamma Remote Sensing, v1.6 12-Feb-2018 clw/cm/awi

Parameters

- **annot_XML** –
(input) TerraSAR-X ScanSAR product annotation XML file including path
 - NOTE: The path to the image products is determined from the path to the XML annotation
- **swath** –
number specifying the desired ScanSAR swath (1 -> maximum number of swaths (4 or 6))
 - NOTE: The image product name is specified in the XML file
- **SLC_par** – (output) ISP SLC parameter file (example: yyyyymmdd.slc.par)
- **SLC** – (output) SLC ScanSAR data file, example: yyyyymmdd.slc (enter - for none, SLC output will not be produced)
- **TOPS_par** – (output) SLC ScanSAR burst annotation file (example: yyyyymmdd_sl.tops_par)
- **bwflg** –
burst window flag:
 - 0: use first and last annotation line values specified in the annot_XML
 - 1: extend first and last valid line to include all data lines (default)
 - NOTE: While TSX ScanSAR data are not acquired in TOPS mode, the same data structure can be used for burst annotation
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.par_UAVSAR_SLC(ann, SLC_MLI_par, image_type, image_format,
                                           logpath=None, outdir=None, shellscript=None)
```

ISP image parameter file from UAVSAR annotation file (ann) for SLC and MLC products

Copyright 2014, Gamma Remote Sensing, v1.3 20-Aug-2014 clw

Parameters

- **ann** – (input) UAVSAR annotation file (*ann.txt)
- **SLC_MLI_par** – (output) ISP image parameter file
- **image_type** –
image type flag

- 0: SLC (slc) in slant range coordinates
- 1: MLC (mlc) in slant range coordinates HHHH*, VVVV*, HVHV* are FLOAT format HHHV*, HHVV*, HVVV* are FCOMPLEX format
- **image_format** –
image data format flag
 - 0: FCOMPLEX (pairs of 4-byte float (re,im))
 - 2: FLOAT (4-bytes/value)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ph_slope_base(int_in, SLC_par, OFF_par, base, int_out,  
                                         int_type='-', inverse='-', logpath=None, out-  
                                         dir=None, shellscript=None)
```

Subtract/add interferogram flat-Earth phase trend as estimated from initial baseline

Copyright 2006, Gamma Remote Sensing, v4.4 3-Nov-2006 clw

Parameters

- **int_in** – (input) interferogram (FCOMPLEX) or unwrapped phase (FLOAT) (unflattened)
- **SLC_par** – (input) ISP parameter file for the reference SLC
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **base** – (input) baseline file
- **int_out** – (output) interferogram (FCOMPLEX) or unwrapped phase (FLOAT) with phase trend subtracted/added
- **int_type** – interferogram type: 0=unwrapped phase, 1=complex interf. (default=1)
- **inverse** – subtract/add inversion flag (0=subtract phase ramp, 1=add phase ramp (default=0))
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.phase_slope(interf, slopes, width, win_sz='-', thres='-', xmin='-',  
                                       xmax='-', ymin='-', ymax='-', logpath=None, out-  
                                       dir=None, shellscript=None)
```

Calculate interferogram phase slopes in range and azimuth

Copyright 2011, Gamma Remote Sensing, v1.3 19-Apr-2011 clw/uw

Parameters

- **interf** – (input) interferogram (fcomplex)
- **slopes** – (output) range and azimuth phase slopes (fcomplex)
- **width** – number of samples/row

- **win_sz** – size of region used for slopes determination (default = 5)
- **thres** – correlation threshold for accepting slope estimates 0.0 -> 1.0 (default=.4)
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row offset (default = 0)
- **ymax** – last azimuth row offset (default = nlines-1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ptarg_SLC(SLC_par, SLC, r_samp, az_samp, ptr_image, r_plot,
                                     az_plot, ptr_par='-', osf='-', win='-', pltflg='-', log-
                                     path=None, outdir=None, shellscript=None)
```

Point target response analysis and interpolation for SLC images
 Copyright 2016, Gamma Remote Sensing, v1.9 19-Feb-2016 clw

Parameters

- **SLC_par** – (input) SLC image parameter file
- **SLC** – (input) SLC image in FCOMPLEX or SCOMPLEX format
- **r_samp** – point target range sample number
- **az_samp** – point target azimuth line number
- **ptr_image** – (output) oversampled point target image (fcomplex, 1024x1024 samples), with and without phase gradient
- **r_plot** – (output) range point target response plot data (text format)
- **az_plot** – (output) azimuth point target response plot data (text format)
- **ptr_par** – (output) measured point target parameters (text format)
- **osf** – image over-sampling factor, 2, 4, 8, 16, 32, 64 (enter - for default: 16)
- **win** – maximum search window offset (samples) (enter - for default: 1)
- **pltflg** –
plotting mode flag:
 - 0: none
 - 1: output plots in PNG format (default)
 - 2: screen output
 - 3: output plots in PDF format
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ptarg_cal_MLI(MLI_par, MLI, r_samp, az_samp, psigma,
                                          c_r_samp, c_az_samp, ptr_image, r_plot, az_plot,
                                          pcal, osf='-', win='-', pltflg='-', psz='-', csz='-',
                                          theta_inc='-', logpath=None, outdir=None,
                                          shellscript=None)
```

Point target analysis and radiometric calibration of slant-range and ground-range (GRD) images

Copyright 2016, Gamma Remote Sensing, v2.6 19-Feb-2016 clw

Parameters

- **MLI_par** – (input) slant-range or ground-range image parameter file for detected intensity data
- **MLI** – (input) ground-range or slant range detected image in FLOAT format
- **r_samp** – point target range sample number, target region size is 16x16
- **az_samp** – point target azimuth line number, target region size is 16x16
- **psigma** – radar cross-section of the calibration target in m**2
- **c_r_samp** – clutter region center range sample number, clutter region size is 16x16
- **c_az_samp** – clutter region center azimuth line number, clutter region size is 16x16
- **ptr_image** – (output) oversampled point target image, with and without phase gradient, nominal width: 256
- **r_plot** – (output) range point target response plot data (text format)
- **az_plot** – (output) azimuth point target response plot data (text format)
- **pcal** – (output) measured point target parameters and radiometric calibration factor (text format)
- **osf** – image over-sampling factor, 2, 4, 8, 16, 32, 64 (enter - for default: 16)
- **win** – maximum search window offset (samples) (enter - for default: 1)
- **pltflg** –
 plotting mode flag:
 - 0: none
 - 1: output plots in PNG format (default)
 - 2: screen output
 - 3: output plots in PDF format
- **psz** – point target region size (samples) (enter - for default: 16)
- **csz** – clutter region size (samples) (enter - for default: 16)
- **theta_inc** – incidence angle required for calibration of terrain corrected RISAT-1 images
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.ptarg_cal_SLC(SLC_par, SLC, r_samp, az_samp, psigma,
                                         c_r_samp, c_az_samp, ptr_image, r_plot, az_plot,
                                         pcal, osf='-', win='-', pltflg='-', psz='-', csz='-',
                                         c_image='-', logpath=None, outdir=None,
                                         shellscript=None)
```

Point target analysis and radiometric calibration of SLC images

Copyright 2016, Gamma Remote Sensing, v2.4 19-Feb-2016 clw

Parameters

- **SLC_par** – (input) SLC image parameter file
- **SLC** – (input) SLC image in FCOMPLEX or SCOMPLEX format
- **r_samp** – point target range sample number, target region size is 16x16
- **az_samp** – point target azimuth line number, target region size is 16x16
- **psigma** – radar cross-section of the calibration target in m**2
- **c_r_samp** – clutter region center range sample number, clutter region size is 16x16
- **c_az_samp** – clutter region center azimuth line number, clutter region size is 16x16
- **ptr_image** – (output) oversampled point target image, with and without phase gradient, nominal width: 256
- **r_plot** – (output) range point target response plot data (text format)
- **az_plot** – (output) azimuth point target response plot data (text format)
- **pcal** – (output) measured point target parameters and radiometric calibration factor (text format)
- **osf** – image over-sampling factor, 2, 4, 8, 16, 32, 64 (enter - for default: 16)
- **win** – maximum search window offset (samples) (enter - for default: 1)
- **pltflg** –
plotting mode flag:
 - 0: none
 - 1: output plots in PNG format (default)
 - 2: screen output
 - 3: output plots in PDF format
- **psz** – point target region size (samples) (enter - for default: 16)
- **csz** – clutter region size (samples) (enter - for default: 16)
- **c_image** – (output) clutter region image (FCOMPLEX format)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.pwr2ras(MLI_tab, width, scale='-', exp='-', logpath=None, out-
                                   dir=None, shellscript=None)
```

Generate raster images of MLI image files

Copyright 2012, Gamma Remote Sensing, v1.2 27-Nov-2012 clw

Parameters

- **MLI_tab** – (input) list of mli intensity images (float)
- **width** – image width of input data files
- **scale** – image display scale factor (enter - for default, default: 0.9)
- **exp** – image display exponent (default: 0.35)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.radcal_MLI (MLI, MLI_par, OFF_par, CMLI, antenna='-  
, rloss_flag='- ', ant_flag='- ', refarea_flag='- ',  
sc_dB='- ', K_dB='- ', pix_area='- ', logpath=None,  
outdir=None, shellscript=None)
```

Radiometric calibration for multi-look intensity (MLI) data

Copyright 2016, Gamma Remote Sensing, v2.0 9-Nov-2016 uw/clw/of

Parameters

- **MLI** – (input) MLI image (float)
- **MLI_par** – (input) SLC parameter file of input MLI image
- **OFF_par** – (input) ISP offset/interferogram parameter file (enter - for images in MLI geometry)
- **CMLI** – (output) radiometrically calibrated output MLI (float)
- **antenna** – (input) 1-way antenna gain pattern file or - if not provided
- **rloss_flag** –
 range spreading loss correction:
 - 0: no correction (default)
 - 1: apply r^3 correction (all modes except ASAR APS)
 - 2: apply r^4 correction (used only for ASAR APS mode)
 - -1: undo r^3 correction
 - -2: undo r^4 correction)
- **ant_flag** –
 antenna pattern correction:
 - 0: no correction (default)
 - 1: apply antenna pattern correction
 - -1: undo antenna pattern correction)
- **refarea_flag** –
 reference pixel area correction:

- 0: no pixel area correction (default)
- 1: calculate sigma0, scale area by $\sin(\text{inc_ang})/\sin(\text{ref_inc_ang})$
- 2: calculate gamma0, scale area by $\sin(\text{inc_ang})/(\cos(\text{inc_ang})*\sin(\text{ref_inc_ang}))$
- -1: undo sigma0 area scaling factor
- -2: undo gamma0 area scaling factor
- **sc_dB** – scale factor in dB (default: 0.0)
- **K_dB** – calibration factor in dB (default: -(value from MLI_par))
- **pix_area** – (output) ellipsoid-based ground range sigma0 or gamma0 pixel reference area (float)
- **refarea_flag** – 1 or -1: sigma0 ref. area
- **refarea_flag** – 2 or -2: gamma0 ref. area
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.radcal_PRI (PRI, PRI_par, GRD, GRD_par, K_dB='-', inc_ref='-',
                                       roff='-', nr='-', loff='-', nl='-', logpath=None, out-
                                       dir=None, shellscript=None)
```

Convert ESA processed short integer format PRI to radiometrically calibrated GRD image (float)

Copyright 2016, Gamma Remote Sensing, v1.5 5-Mar-2016 uw/clw

Parameters

- **PRI** – (input) PRI ground-range image (short integer, sqrt(backscat. intensity))
- **PRI_par** – (input) SLC parameter file of input PRI ground-range image (yyyym-mdd.pri.par)
- **GRD** – (output) calibrated ground-range image (float, backscat. intensity)
- **GRD_par** – (output) ISP image parameter file of output calibrated ground-range image (yyyymmdd.grd.par)
- **K_dB** –
calibration factor in decibels (default: 59.75 dB) ERS1 (D-Paf,ESRIN): 58.24 dB,
ERS2 (D-Paf,ESRIN,I-Paf,UK-Paf after 1997): 59.75 dB ENVISAT ASAR: 55.0 dB (all
modes) for details see product specifications and ESA publications.
- **inc_ref** –
reference incidence angle in deg. (default: 23.0 deg.) ENVISAT ASAR: 90.0 deg. (all
modes)
- **roff** – offset to starting range sample (default: 0)
- **nr** – number of range samples (default: 0, to end of line)
- **loff** – offset to starting line (default: 0, 1 header line in the input file is assumed for ERS)
- **nl** – number of lines to copy (default: 0, to end of file)
- **logpath** (*str* or *None*) – a directory to write command logfiles to

- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.radcal_SLC(SLC, SLC_par, CSLC, CSLC_par, fcase='-'  
, antenna='-', rloss_flag='-', ant_flag='-',  
refarea_flag='-', sc_dB='-', K_dB='-', pix_area='-',  
logpath=None, outdir=None, shellscript=None)
```

Radiometric calibration of SLC data

Copyright 2016, Gamma Remote Sensing, v2.4 30-Dec-2017 uw/clw/of

Parameters

- **SLC** – (input) SLC (fcomplex or scomplex)
- **SLC_par** – (input) SLC parameter file of input SLC
- **CSLC** – (output) radiometrically calibrated SLC (fcomplex or scomplex)
- **CSLC_par** – (output) SLC parameter file of output calibrated SLC
- **fcase** –
format case (default = 1)
 - 1: fcomplex → fcomplex (pairs of float)
 - 2: fcomplex → scomplex (pairs of short integer)
 - 3: scomplex → fcomplex
 - 4: scomplex → scomplex
- **antenna** – 1-way antenna gain pattern file or - (if not provided)
- **rloss_flag** –
range spreading loss correction:
 - 0: no correction (default)
 - 1: apply r^3 correction (all modes except ASAR APS)
 - 2: apply r^4 correction (used only for ASAR APS mode)
 - -1: undo r^3 correction
 - -2: undo r^4 correction)
- **ant_flag** –
antenna pattern correction:
 - 0: no correction (default)
 - 1: apply antenna pattern correction
 - -1: undo antenna pattern correction)
- **refarea_flag** –
reference pixel area correction:
 - 0: no pixel area correction (default)
 - 1: calculate σ_0 , scale area by $\sin(\text{inc_ang})/\sin(\text{ref_inc_ang})$

- 2: calculate gamma0, scale area by $\sin(\text{inc_ang})/(\cos(\text{inc_ang})*\sin(\text{ref_inc_ang}))$
- -1: undo sigma0 area scaling factor
- -2: undo gamma0 area scaling factor
- **sc_dB** – scale factor in dB (default: 0.0)
- **K_dB** – calibration factor in dB (default: -(value from SLC_par))
- **pix_area** – (output) ellipsoid-based ground range sigma0 or gamma0 pixel reference area (float)
- **refarea_flag** – 1 or -1: sigma0 ref. area
- **refarea_flag** – 2 or -2: gamma0 ref. area
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.radcal_pwr_stat(SLC_tab, SLC_tab_cal, plist, MSR_cal,
                                           PWR_cal, roff='-', loff='-', nr='-', nl='-',
                                           plist_out='-', logpath=None, outdir=None,
                                           shellscript=None)
```

Generate calibrated SLC image files using point targets determined from the Mean/Sigma Ratio and Intensity
 Copyright 2018, Gamma Remote Sensing, v1.4 25-Apr-2018 clw/uw/cm

Parameters

- **SLC_tab** – (input) two column list of the SLC filenames and SLC parameter filenames of the uncalibrated SLC images
- **SLC_tab_cal** – (input) two column list of the SLC filenames and SLC parameter filenames of the calibrated SLC images (enter - for none)
- **plist** – (input) point list for the point to use for calibration (int, enter - to use the data to determine the calibration points)
- **MSR_cal** – mean/sigma ratio for point target selection for relative calibration between scenes: 1.500
- **PWR_cal** – intensity threshold ratio for point target selection for relative calibration between scenes: 1.000
- **roff** – offset to starting range of section to analyze (default -: 0)
- **loff** – offset to starting line of section to analyze (default -: 0)
- **nr** – number of range pixels to analyze (default -: to end of line)
- **nl** – number of azimuth lines to analyze (default -: to end of file)
- **plist_out** – point list of points used to determine calibration using MSR_cal and PWR_cal thresholds
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.rascc_mask(cc, pwr, width, start_cc='-', start_pwr='-', nlines='-',
                                       pixavr='-', pixavaz='-', cc_thres='-', pwr_thres='-',
                                       cc_min='-', cc_max='-', scale='-', exp='-',
                                       LR='-', rasf='-', logpath=None, outdir=None,
                                       shellscript=None)
```

Generate phase unwrapping validity mask using correlation and intensity

Copyright 2016, Gamma Remote Sensing, v2.0 12-Sep-2016 clw/uw

Parameters

- **cc** – (input) interferometric correlation image (float)
- **pwr** – (input) intensity image (float, enter - if not available)
- **width** – number of samples/row
- **start_cc** – starting line of coherence image (default: 1)
- **start_pwr** – starting line of intensity image (default: 1)
- **nlines** – number of lines to display (default=0: to end of file)
- **pixavr** – number of pixels to average in range (default: 1)
- **pixavaz** – number of pixels to average in azimuth (default: 1)
- **cc_thres** – coherence threshold for masking, pixels with $cc < cc_thres$ are set to 0 (default: 0.0)
- **pwr_thres** – relative intensity threshold for masking, pixels with $intensity < pwr_thres * average\ intensity$ are set to 0 (default: 0)
- **cc_min** – minimum coherence value used for color display (default: 0.1)
- **cc_max** – maximum coherence value used for color display (default: 0.9)
- **scale** – intensity display scale factor (default: 1.)
- **exp** – intensity display exponent (default: .35)
- **LR** – left/right mirror image flag, (1: normal (default), -1: mirror image)
- **rasf** –
(output) image filename, extension determines the format, enter - for default: *.ras
*.bmp BMP format *.ras Sun raster format *.tif TIFF format
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.rascc_mask_thinning(ras_in, in_file, width, ras_out, nmax='-',
                                                thresholds='-', logpath=None, outdir=None,
                                                shellscript=None)
```

Adaptive sampling reduction for phase unwrapping validity mask

Copyright 2015, Gamma Remote Sensing, v1.5 5-Dec-2015 uw/clw

Parameters

- **ras_in** – (input) validity mask (SUN/BMP/TIFF raster format 8-bit image)

- **in_file** – (input) file used for adaptive sampling reduction, e.g. correlation coefficient (float)
- **width** – number of samples/row of in_file
- **ras_out** – (output) validity mask with reduced sampling (8-bit SUN rasterfile or BMP format image)
- **nmax** – number of sampling reduction runs (default: 3)
- **thresholds** – a list of thresholds sorted from smallest to largest scale sampling reduction
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.res_map(hgt, gr, data, SLC_par, OFF_par, res_hgt, res_data, nr='-',
                                   naz='-', azps_res='-', loff='-', nlines='-', logpath=None,
                                   outdir=None, shellscript=None)
```

Slant range to ground range transformation based on interferometric ground-range

Copyright 2008, Gamma Remote Sensing, v2.3 5-Sep-2008 clw/uw

Parameters

- **hgt** – (input) height file in slant range geometry
- **gr** – (input) ground range file in slant range geometry
- **data** – (input) data file in slant range geometry (float) (intensity *.pwr or correlation *.cc)
- **SLC_par** – (input) ISP parameter file of reference SLC
- **OFF_par** – (input) offset/interferogram processing parameters
- **res_hgt** – (output) resampled height file in ground range geometry
- **res_data** – (output) resampled data file in ground range geometry
- **nr** – number of range samples for L.S. estimate (default=7, must be odd)
- **naz** – number of azimuth samples for L.S. estimate (default=7, must be odd)
- **azps_res** – azimuth output map sample spacing in meters (default=azimuth spacing)
- **loff** – offset to starting line for height calculations (default=0)
- **nlines** – number of lines to calculate (default=to end of file)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.residue(int, flag, width, xmin='-', xmax='-', ymin='-', ymax='-',
                                   logpath=None, outdir=None, shellscript=None)
```

Determine interferometric phase unwrapping residues

Copyright 2014, Gamma Remote Sensing, v2.6 14-Jan-2014 clw/uw

Parameters

- **int** – (input) interferogram (fcomplex)

- **flag** – (input) flag file (unsigned char)
- **width** – number of samples/row
- **xmin** – offset to starting range pixel (default = 0)
- **xmax** – offset last range pixel (default = width-1)
- **ymin** – offset to starting azimuth row (default = 0)
- **ymax** – offset to last azimuth row (default = nlines-1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.residue_cc (int, flag, width, xmin='-', xmax='-', ymin='-', ymax='-',  
                                       logpath=None, outdir=None, shellscript=None)
```

Determine interferometric phase unwrapping residues considering low coherence regions
Copyright 2014, Gamma Remote Sensing, v2.6 20-Jan-2014 clw/uw/ts

Parameters

- **int** – (input) interferogram (fcomplex)
- **flag** – (input) flag file (unsigned char)
- **width** – number of samples/row
- **xmin** – offset to starting range pixel (default = 0)
- **xmax** – offset last range pixel (default = width-1)
- **ymin** – offset to starting azimuth row (default = 0)
- **ymax** – offset to last azimuth row (default = nlines-1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.run_all (list, command, logpath=None, outdir=None,  
                                   shellscript=None)
```

Parameters

- **list** – (input) list file (ascii)
- **command** –
command to use (example 1: 'ls -l \$1.slc') (example 2: 'JERS_PROC JERS-1_720.par \$1
2 6 ... 8192')
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.sbi_filt (SLC_1, SLC1_par, SLC2R_par, SLCf, SLCf_par, SLCb,
                                     SLCb_par, norm_sq, iwflg='-', logpath=None, out-
                                     dir=None, shellscript=None)
```

Azimuth filtering of SLC data to support split-beam interferometry to measure azimuth offsets

Copyright 2016, Gamma Remote Sensing, v1.2 clw 5-Mar-2016

Parameters

- **SLC_1** – (input) SLC image (SCOMPLEX or FCOMPLEX format)
- **SLC1_par** – (input) SLC image parameter file
- **SLC2R_par** –
(input) SLC2 ISP image parameter file for the co-registered image of the interferometric pair,
used to determine azimuth common-band for each output SLC (enter - for none)
- **SLCf** – (output) SLC image (forward-looking, FCOMPLEX format)
- **SLCf_par** – (output) SLC parameter file (forward-looking)
- **SLCb** – (output) SLC image (backward-looking, FCOMPLEX format)
- **SLCb_par** – (output) SLC parameter file (backward-looking)
- **norm_sq** – squint between beams as a fraction of the azimuth spectrum width (default: 0.5)
- **iwflg** –
inverse weighting flag:
 - 0: no compensation for azimuth spectrum weighting
 - 1: compensate for the azimuth spectrum weighting (default)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.sbi_offset (sbi_unw, SLCf_par, SLCb_par, OFF_par, az_offset,
                                       logpath=None, outdir=None, shellscript=None)
```

Calculate azimuth offsets from unwrapped split-beam interferogram

Copyright 2011, Gamma Remote Sensing, v1.0 25-Nov-2011

Parameters

- **sbi_unw** – (input) unwrapped phase of split-beam interferogram (float)
- **SLCf_par** – (input) reference SLC parameter file (forward-looking)
- **SLCb_par** – (input) reference SLC parameter file (backward-looking)
- **OFF_par** – (input) offset parameter file
- **az_offset** – (output) azimuth offsets (m)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.slant_range(SLC_par, slr, logpath=None, outdir=None,  
                                       shellscript=None)
```

Calculate slant range for every range sample

Copyright 2013, Gamma Remote Sensing v1.1 28-Aug-2013

Parameters

- **SLC_par** – (input) SLC or MLI image parameter file
- **slr** – (output) slant range for every sample in the image (float)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.split_WB(data_in, data_par_in, data_tab, dtype, logpath=None, out-  
                                   dir=None, shellscript=None)
```

ISP: Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/split_WB.c

Copyright 2018, Gamma Remote Sensing, v1.3 25-Apr-2018 clw/cm

Split WB mosaic image into individual beams using ISP parameter files

Parameters

- **data_in** – (input) input mosaicked data in slant-range geometry (e.g. DEM data)
- **data_par_in** – (input) ISP image parameter file for data in the input mosaic
- **data_tab** – (input) 2 column list of output data filenames and ISP image parameter files for each beam in the mosaic (text)
- **dtype** –
(input) input data type:
 - 0: FLOAT
 - 1: FCOMPLEX
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.subtract_phase(interf_in, phase_file, interf_out, width, factor='-',  
                                         logpath=None, outdir=None, shellscript=None)
```

Land Application Tools: Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/subtract_phase.c

Copyright 2001, Gamma Remote Sensing, v3.1 23-Jan-2001 uw/clw

subtract scaled phase image from a complex interferogram

Parameters

- **interf_in** – (input) input interferogram (fcomplex format)
- **phase_file** – (input) unwrapped interferometric phase (float)
- **interf_out** – (output) output interferogram (input interferogram - scaled phase) (fcomplex)

- **width** – number of samples/line
- **factor** – constant scale factor for input phase data [default=1.0]
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.tree_cc(flag, width, mbl='- ', xmin='- ', xmax='- ', ymin='- ', ymax='- ',
                                   logpath=None, outdir=None, shellscript=None)
```

Phase unwrapping tree generation with low correlation search (modified ARW algorithm)

Copyright 2014, Gamma Remote Sensing, v2.9 20-Jan-2014 clw/uw

Parameters

- **flag** – (input) phase unwrapping flag file
- **width** – number of samples/row
- **mbl** – maximum branch length (default=32, maximum=64)
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row, relative to start (default = 0)
- **ymax** – last azimuth row, relative to start (default = nlines-1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.tree_gzw(flag, width, mbl='- ', xmin='- ', xmax='- ', ymin='- ',
                                     ymax='- ', logpath=None, outdir=None,
                                     shellscript=None)
```

Phase unwrapping tree generation (GZW algorithm)

Copyright 2008, Gamma Remote Sensing, v3.6 5-Sep-2008 clw/uw

Parameters

- **flag** – (input) phase unwrapping flag file
- **width** – number of samples/row
- **mbl** – maximum branch length (default=32)
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row, relative to start (default = 0)
- **ymax** – last azimuth row, relative to start (default = nlines-1)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.unw_correction_filt(unw_in, unw_out, width, fsize='',  
                                              thresh1='', thresh2='', iterations='',  
                                              cleaning='', logpath=None, out-  
                                              dir=None, shellscript=None)
```

Parameters

- **unw_in** – (input) unwrapped phase file to correct (float)
- **unw_out** – (output) corrected unwrapped phase file (float)
- **width** – number of range samples per line
- **fsize** – maximum filter radius in pixels (default = 5)
- **thresh1** – upper threshold for negative phase differences (default = -3.0)
- **thresh2** – lower threshold for positive phase differences (default = 3.0)
- **iterations** – number of iterations to run (default = 1)
- **cleaning** –
cleaning flag indicating if intermediary files are deleted (default = 1: yes, 0: no) The difference between the unfiltered and spatially filtered phase (using fspf) is used to determine an correct phase unwrapping ambiguity errors
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.unw_correction_poly(unw_in, unw_out, width, poly, flag,  
                                              max_iter='', logpath=None, out-  
                                              dir=None, shellscript=None)
```

Parameters

- **unw_in** – (input) unwrapped phase file to correct (float)
- **unw_out** – (output) corrected unwrapped phase file (float)
- **width** – number of range samples per line
- **poly** – (input) polygon file (text)
- **flag** – ambiguity corrected flag (1: add 2PI; -1: subtract 2PI)
- **max_iter** –
maximum number of iterations done (default = 1) (iterations are used (a) if the ambiguity to correct is not 2PI but a multiple of 2PI and (b) if the ambiguity error is in an area with a significant phase slope)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format

```
pyroSAR.gamma.parser_demo.unw_model(interf, unw_model, unw, width, xinit='-', yinit='-',
                                     ref_ph='-', width_model='-', logpath=None, out-
                                     dir=None, shellscript=None)
```

Phase unwrapping using a model of the unwrapped phase

Copyright 2008, Gamma Remote Sensing, v1.6 5-Sep-2008 clw/uw

Parameters

- **interf** – (input) complex interferogram
- **unw_model** – (input) approximate unwrapped phase model (float)
- **unw** – (output) unwrapped phase (float)
- **width** – number of samples/row of the interferogram
- **xinit** – offset to phase reference location in range (col)
- **yinit** – offset to phase reference location in azimuth (row)
- **ref_ph** – reference point phase (radians) (enter - for phase at the reference point)
- **width_model** – number of samples/row of the unwrapped phase model (default: interferogram width)
- **logpath** (*str* or *None*) – a directory to write command logfiles to
- **outdir** (*str* or *None*) – the directory to execute the command in
- **shellscript** (*str* or *None*) – a file to write the Gamma commands to in shell format


```
class pyroSAR.S1.OSV(osvdir)
```

Bases: `object`

interface for management of S1 Orbit State Vector (OSV) files

input is a directory which is supposed to contain, or already contains, OSV files. Two subdirectories are expected and created otherwise: one for Precise Orbit Ephemerides (POE) named POEORB and one for Restituted Orbit (RES) files named RESORB

Using method `match()` the corresponding POE (priority) or RES file is returned for a timestamp. Timestamps are always handled in the format YYYYmmddTHHMMSS.

Parameters `osvdir` (*str*) – the directory to write the orbit files to

catch (*osvtype*='POE', *start*=None, *stop*=None)
check a server for files

Parameters

- **osvtype** (`{ 'POE', 'RES' }`) – the type of orbit files required
- **start** (*str*) – the date to start searching for files
- **stop** (*str*) – the date to stop searching for files

Returns the URLs of the remote OSV files

Return type `list`

clean_res ()
delete all RES files for whose date a POE file exists

date (*file*, *datatype*)
extract a date from an OSV file name

Parameters `datatype` (`{ 'publish', 'start', 'stop' }`) – one of three possible date types contained in the OSV filename

Returns a time stamp in the format YYYYmmddTHHMMSS

Return type `str`

getLocals (*osvtype*='POE')

get a list of local files of specific type

Parameters **osvtype** (`{ 'POE', 'RES' }`) – the type of orbit files required

Returns a selection of local OSV files

Return type `list`

match (*timestamp*, *osvtype*='POE')

return the corresponding OSV file for the provided time stamp. The file returned is one which covers the acquisition time and, if multiple exist, the one which was published last. In case a list of options is provided as osvtype, the file of higher accuracy (i.e. POE over RES) is returned.

Parameters

- **timestamp** (*str*) – the time stamp in the format 'YYYYmmddTHHMMSS'
- **osvtype** (`{ 'POE', 'RES' }` or *list*) – the type of orbit files required; either 'POE', 'RES' or a list of both

Returns the best matching orbit file (overlapping time plus latest publication date)

Return type `str`

maxdate (*osvtype*='POE', *datetype*='stop')

return the latest date of locally existing POE/RES files

Parameters

- **osvtype** (`{ 'POE', 'RES' }`) – the type of orbit files required
- **datetype** (`{ 'publish', 'start', 'stop' }`) – one of three possible date types contained in the OSV filename

Returns a timestamp in format YYYYmmddTHHMMSS

Return type `str`

mindate (*osvtype*='POE', *datetype*='start')

return the earliest date of locally existing POE/RES files

Parameters

- **osvtype** (`{ 'POE', 'RES' }`) – the type of orbit files required
- **datetype** (`{ 'publish', 'start', 'stop' }`) – one of three possible date types contained in the OSV filename

Returns a timestamp in format YYYYmmddTHHMMSS

Return type `str`

retrieve (*files*)

download a list of remote files into the respective subdirectories, i.e. POEORB or RESORB

Parameters **files** (*list*) – a list of remotely existing OSV files as returned by method `catch()`

sortByDate (*files*, *datetype*='start')

sort a list of OSV files by a specific date type

Parameters

- **files** (*list*) – some OSV files

- **datatype** (`{'publish', 'start', 'stop'}`) – one of three possible date types contained in the OSV filename

Returns the input OSV files sorted by the defined date

Return type `list`

update (`update_res=True`)

Caution! This method is intended for downloading all available POE files and all RES files for whose time span no POE file yet exists locally. This will be a data volume of several GB and is particularly suited for multi-node SAR processing where not all nodes might have internet access and thus all files have to be downloaded before starting the processing.

If you want to download the OSV file for a single scene either use the respective methods of the SAR drivers (e.g. `pyroSAR.drivers.SAFE.getOSV()`) or methods `catch()` and `retrieve()` in combination.

Perform creating/updating operations for POE and RES files: download newest POE and RES files, delete RES files which can be replaced by newly downloaded POE files.

actions performed:

- the ESA Quality Control (QC) server is checked for any POE files not in the local directory
- POE files on the server and not in the local directory are downloaded
- RES files newer than the latest POE file are downloaded; POE files are approximately 18 days behind the actual date, thus RES files can be used instead
- delete all RES files for whose date a POE file now exists locally

Parameters `update_res` (`bool`) – should the RES files also be updated (or just the POE files)

`pyroSAR.S1.removeGRDBorderNoise` (`scene`)

mask out Sentinel-1 image border noise

Reference: Masking “No-value” Pixels on GRD Products generated by the Sentinel-1 ESA IPF’ (issue 1, June 2015; issue 2.1 Jan 29 2018) available online under <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/document-library>

Parameters `scene` (`SAFE`) – the Sentinel-1 scene object

Auxiliary Data Tools

```
pyroSAR.auxdata.dem_autoload(geometries, demType, vrt=None, buffer=None, username=None,  
                             password=None)
```

obtain all relevant DEM tiles for selected geometries

Parameters

- **geometries** (*list*) – a list of `spatialist.vector.Vector` geometries to obtain DEM data for; CRS must be WGS84 LatLon (EPSG 4326)
- **demType** (*str*) – the type of DEM to be used; current options:
 - 'AW3D30' (ALOS Global Digital Surface Model "ALOS World 3D - 30m (AW3D30)")
 - * url: ftp://ftp.eorc.jaxa.jp/pub/ALOS/ext1/AW3D30/release_v1804
 - 'SRTM 1Sec HGT'
 - * url: <https://step.esa.int/auxdata/dem/SRTMGL1>
 - 'SRTM 3sec'
 - * url: http://srtm.csi.cgiar.org/wp-content/uploads/files/srtm_5x5/TIFF
 - 'TDX90m'
 - * registration: <https://geoservice.dlr.de/web/dataguide/tdm90>
 - * url: <ftpes://tandemx-90m.dlr.de>
- **vrt** (*str* or *None*) – an optional GDAL VRT file created from the obtained DEM tiles
- **buffer** (*int* or *float*) – a buffer in degrees to add around the individual geometries
- **username** (*str* or *None*) – (optional) the user name for services requiring registration
- **password** (*str* or *None*) – (optional) the password for the registration account

Returns the names of the obtained files or the name of the VRT file

Return type `list` or `str`

Examples

download all SRTM 1 arcsec DEMs overlapping with a Sentinel-1 scene and mosaic them to a single GeoTiff file

```
>>> from pyroSAR import identify
>>> from pyroSAR.auxdata import dem_autoload
>>> from spatialist import gdalwarp
# identify the SAR scene
>>> scene = identify('S1A_IW_GRDH_1SDV_20141012T162337_20141012T162402_002799_
↳00326F_8197.zip')
# extract the bounding box as spatialist.Vector object
>>> bbox = scene.bbox()
# download the tiles and virtually combine them in an in-memory VRT file,
↳subsetting to the extent of the SAR scene
>>> vrt = dem_autoload(geometries=[bbox], demType='SRTM 1Sec HGT', vrt='/vsimem/
↳srtm1.vrt', buffer=0.01)
# write the final GeoTiff file
>>> outname = scene.outname_base() + 'srtm1.tif'
>>> gdalwarp(src=vrt, dst='', options={'format': 'GTiff'})
```

This (still experimental) module is intended to easily prepare SAR scenes processed by pyroSAR for ingestion into an Open Data Cube.

```
class pyroSAR.datacube_util.Dataset (filename, units='DN')
```

Bases: `object`

```
__add__ (dataset)
```

override the + operator. This is intended to easily combine two Dataset objects, which were created from different files belonging to the same measurement, e.g. two GeoTiffs with one polarization each.

Parameters `dataset` (`Dataset`) – the dataset to add to the current one

Returns the combination of the two

Return type `Dataset`

```
__radd__ (dataset)
```

similar to Dataset.__add__ but for function sum, e.g. sum([Dataset1, Dataset2])

Parameters `dataset` (`Dataset`) – the dataset to add to the current one

Returns the combination of the two

Return type `Dataset`

```
close ()
```

```
filenames
```

Returns all file names registered in the dataset

Return type `dict`

```
identifier
```

Returns a unique dataset identifier

Return type `str`

```
units
```

Returns all measurement unit names registered in the dataset

Return type `dict`

class `pyroSAR.datacube_util.Product` (*definition=None, name=None, product_type=None, description=None*)

Bases: `object`

add (*dataset*)

Add a dataset to the abstracted product description. This first performs a check whether the dataset is compatible with the product and its already existing measurements. If a measurement in the dataset does not yet exist in the product description it is added.

Parameters **dataset** (`Dataset`) – the dataset whose description is to be added

check_integrity (*dataset, allow_new_measurements=False*)

check if a dataset is compatible with the product definition.

Parameters

- **dataset** (`Dataset`) – the dataset to be checked
- **allow_new_measurements** (`bool`) – allow new measurements to be added to the product definition? If not and the dataset contains measurements, which are not defined in the product, an error is raised.

Raises `RuntimeError`

close ()

export_indexing_yaml (*dataset, outdir*)

export_ingestion_yaml (*outname, product_name, ingest_location*)

measurements

Returns a dictionary with measurement names as keys

Return type `dict of dict`

write (*yamlfile*)

write the product definition to a YAML file

Parameters **yamlfile** (`str`) – the file to write to

Ancillary Functions

This script gathers central functions and classes for general pyroSAR applications.

<i>find_datasets</i>	find pyroSAR datasets in a directory based on their metadata
<i>groupby</i>	group a list of images by a metadata attribute
<i>groupbyTime</i>	function to group images by their acquisition time difference
<i>parse_datasetname</i>	Parse the name of a pyroSAR processing product and extract its metadata components as dictionary
<i>seconds</i>	function to extract time in seconds from a file name.

`pyroSAR.ancillary.find_datasets(directory, recursive=False, **kwargs)`
find pyroSAR datasets in a directory based on their metadata

Parameters

- **directory** (*str*) – the name of the directory to be searched
- **recursive** (*bool*) – search the directory recursively into subdirectories?
- **kwargs** – Metadata attributes for filtering the scene list supplied as *key=value*. e.g. *sensor='SIA'*. Multiple allowed options can be provided in tuples, e.g. *sensor=('SIA', 'SIB')*. Any types other than tuples require an exact match, e.g. *proc_steps=['grd', 'mli', 'geo', 'norm', 'db']* will be matched if only these processing steps are contained in the product name in this exact order. See function `parse_productname()` for options.

Returns the file names found in the directory and filtered by metadata attributes

Return type list of str

Examples

```
>>> selection = find_datasets('path/to/files', sensor=('S1A', 'S1B'),  
↪ polarization='VV')
```

`pyroSAR.ancillary.groupby(images, attribute)`
group a list of images by a metadata attribute

Parameters

- **images** (*list of str*) – the names of the images to be sorted
- **attribute** (*str*) – the name of the attribute used for sorting; see `parse_datasetname()` for options

Returns a list containing a list with image names ofr each group

Return type list of lists

`pyroSAR.ancillary.groupbyTime(images, function, time)`
function to group images by their acquisition time difference

Parameters

- **images** (*list of str*) – a list of image names
- **function** (*function*) – a function to derive the time from the image names; see e.g. `seconds()`
- **time** (*int or float*) – a time difference in seconds by which to group the images

Returns a list of sub-lists containing the grouped images

Return type list

`pyroSAR.ancillary.parse_datasetname(name, parse_date=False)`
Parse the name of a pyroSAR processing product and extract its metadata components as dictionary

Parameters

- **name** (*str*) – the name of the file to be parsed
- **parse_date** (*bool*) – parse the start date to a `datetime` object or just return the string?

Returns the metadata attributes

Return type dict

Examples

```
>>> meta = parse_datasetname('S1A_IW__A_20150309T173017_VV_grd_mli_geo_norm_db.  
↪tif')  
>>> print(list(meta.keys()))  
['sensor', 'acquisition_mode', 'orbit', 'start', 'extensions', 'polarization',  
↪'proc_steps']
```

`pyroSAR.ancillary.seconds(filename)`
function to extract time in seconds from a file name. the format must follow a fixed pattern: YYYYmmddTHH-MMSS Images processed with pyroSAR functionalities via module snap or gamma will contain this information.

Parameters **filename** (*str*) – the name of a file from which to extract the time from

Returns the difference between the time stamp in filename and Jan 01 1900 in seconds

Return type `float`

8.1 0.6 / 2018-11-20

8.1.1 SAR metadata

- new standardized metadata fields *orbitNumber_abs*, *orbitNumber_rel*, *cycleNumber* and *frameNumber* for all SAR formats
- customization of output file names with additional metadata fields (e.g. orbit numbers)

8.1.2 software configuration

- pyroSAR configuration file handling: the paths to the SNAP and Gamma installation as well as relevant metadata directories are now registered in a configuration file *config.ini*, which is stored in a directory *.pyrosar* in the user home directory
- improved SNAP installation verification: pyroSAR now performs a deeper check of the SNAP installation to make sure it is not mistaken with e.g. the Ubuntu package manager snap; relevant installation executables and directories are stored in the configuration file

8.1.3 general functionality

- deeper integration of package *spatialist*: all the spatial file handling functionality that was part of pyroSAR is now part of package *spatialist*; now all the functionality is imported from *spatialist* and removed from pyroSAR
- improved search for datasets processed by pyroSAR: new helper functions exist, which make it easier to search for datasets by metadata fields, which are internally searched for in the respective file names
- introduced gamma function parser: these new tools search for a *GAMMA_HOME* environment variable and, if found, parse Python functions from the docstring of respective command line tools; for this, new Python scripts are created, which are stored alongside the configuration file in the user home directory; this way users can easily

use Python functions with named parameters instead of the positional arguments of the Gamma command line tools

- improved documentation

8.1.4 Open Data Cube Export

functionality to export processed datasets directly to an Open Data Cube: it is now possible to create Open Data Cube product YML files as well as YML files for data indexing and ingestion into this product; pyroSAR also internally checks for compatibility of a particular dataset with the target product; this way, the resulting files can easily be passed to the Open Data Cube command line tools several bug fixes

8.1.5 SNAP API

improved SNAP processing workflow node linking: it is now possible to add a node also before an existing one, instead of just after it

8.1.6 Python package integrity

- add trove classifiers for supported operating systems and MIT license for easier online search
- exchange http with https for all URLs that support it

8.2 0.7 / 2019-01-03

several changes to the functioning of the Gamma command API

8.2.1 GAMMA API

processing

- `pyroSAR.gamma.geocode()`:
 - optionally write all Gamma commands to shellscript
 - newly introduced choice of normalization method
 - changed normalization default approach
- `pyroSAR.gamma.process()`:
 - new parameter *logfile* to specify a logfile instead of just a directory with automated file naming
 - new parameter *shellscript* to write the executed command to a shell script protocol

command parser

- add parameters *outdir* and *shellscript* to parsed functions
- extensive improvement to accurately parse more commands
- add parameter *inlist* to some commands, which require interactive input via *stdin*

general

- several bug fixes
- extended documentation
- make use of parsed command functions internally
- enable passing *logpath*, *outdir* and *shellscript* to all parsed functions via additional parameters for other convenience functions

8.3 0.8 / 2019-02-11

8.3.1 Auxiliary Data Handling

- new module `auxdata` with function `pyroSAR.auxdata.dem_autoload()` to automatically download tiles of different DEM types overlapping with given geometries
- class `pyroSAR.S1.OSV`: reduced search time for new RES orbit state vector files; included more meaningful status messages

8.3.2 GAMMA API

- new function `pyroSAR.gamma.srtm.dem_autocreate()` to automatically create DEMs in Gamma format from the output of function `pyroSAR.auxdata.dem_autoload()`
- improved writing of ENVI HDR files from class `pyroSAR.gamma.ISPPar`
- class `pyroSAR.gamma.UTM`: improved to work with newer Gamma versions
- function `pyroSAR.gamma.geocode()`:
 - improved documentation
 - clarified code for better readability
 - more consistent naming scheme for all temporarily written files
 - export temporarily written files (e.g. local incidence angle) via new parameter `export_extra`
 - additional parametrization tests to ensure best processing result
 - changed default of parameter `func_interp` to 2 to work best with default of parameter `normalization_method` (see documentation of Gamma command `pixel_area`)

8.3.3 SNAP API

- function `pyroSAR.snap.util.geocode()`:
 - export temporarily written files (e.g. local incidence angle) via new parameter `export_extra`

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `pyroSAR.ancillary`, [133](#)
- `pyroSAR.auxdata`, [129](#)
- `pyroSAR.datacube_util`, [131](#)
- `pyroSAR.drivers`, [1](#)
- `pyroSAR.gamma`, [17](#)
- `pyroSAR.gamma.parser`, [26](#)
- `pyroSAR.gamma.parser_demo`, [27](#)
- `pyroSAR.gamma.srtm`, [23](#)
- `pyroSAR.S1`, [125](#)
- `pyroSAR.snap.util`, [15](#)

Symbols

`__add__()` (*pyroSAR.datacube_util.Dataset* method), 131
`__radd__()` (*pyroSAR.datacube_util.Dataset* method), 131

A

`adapt_filt()` (in module *pyroSAR.gamma.parser_demo*), 53
`add()` (*pyroSAR.datacube_util.Product* method), 132
`adf()` (in module *pyroSAR.gamma.parser_demo*), 54
`af_SLC()` (in module *pyroSAR.gamma.parser_demo*), 55
`Archive` (class in *pyroSAR.drivers*), 2
`ASAR_LO_phase_drift()` (in module *pyroSAR.gamma.parser_demo*), 27
`ASAR_XCA()` (in module *pyroSAR.gamma.parser_demo*), 27
`autoparse()` (in module *pyroSAR.gamma.parser*), 26
`ave_image()` (in module *pyroSAR.gamma.parser_demo*), 55
`az_integrate()` (in module *pyroSAR.gamma.parser_demo*), 56
`az_spec_SLC()` (in module *pyroSAR.gamma.parser_demo*), 56

B

`base_copy()` (in module *pyroSAR.gamma.parser_demo*), 57
`base_est_fft()` (in module *pyroSAR.gamma.parser_demo*), 57
`base_init()` (in module *pyroSAR.gamma.parser_demo*), 58
`base_ls()` (in module *pyroSAR.gamma.parser_demo*), 59
`base_orbit()` (in module *pyroSAR.gamma.parser_demo*), 59
`base_perp()` (in module *pyroSAR.gamma.parser_demo*), 60

`bbox()` (*pyroSAR.drivers.ID* method), 7
`bpf()` (in module *pyroSAR.gamma.parser_demo*), 60
`bpf_ssi()` (in module *pyroSAR.gamma.parser_demo*), 61
`bridge()` (in module *pyroSAR.gamma.parser_demo*), 61

C

`catch()` (*pyroSAR.S1.OSV* method), 125
`cc_wave()` (in module *pyroSAR.gamma.parser_demo*), 62
`CEOS_ERS` (class in *pyroSAR.drivers*), 4
`CEOS_PSR` (class in *pyroSAR.drivers*), 5
`check_integrity()` (*pyroSAR.datacube_util.Product* method), 132
`clean_res()` (*pyroSAR.S1.OSV* method), 125
`clear_flag()` (in module *pyroSAR.gamma.parser_demo*), 62
`close()` (*pyroSAR.datacube_util.Dataset* method), 131
`close()` (*pyroSAR.datacube_util.Product* method), 132
`close()` (*pyroSAR.drivers.Archive* method), 2
`compression` (*pyroSAR.drivers.ID* attribute), 7
`convert2gamma()` (in module *pyroSAR.gamma*), 20
`corr_flag()` (in module *pyroSAR.gamma.parser_demo*), 63
`correctOSV()` (in module *pyroSAR.gamma*), 22
`create_offset()` (in module *pyroSAR.gamma.parser_demo*), 63

D

`Dataset` (class in *pyroSAR.datacube_util*), 131
`date()` (*pyroSAR.S1.OSV* method), 125
`dcomp_sirc()` (in module *pyroSAR.gamma.parser_demo*), 64
`dcomp_sirc_quad()` (in module *pyroSAR.gamma.parser_demo*), 64
`DELFT_vec2()` (in module *pyroSAR.gamma.parser_demo*), 28
`dem_autocreate()` (in module *pyroSAR.gamma.srtm*), 23

`dem_autoload()` (in module `pyroSAR.auxdata`), 129
`dempar()` (in module `pyroSAR.gamma.srtm`), 24
`DORIS_vec()` (in module `pyroSAR.gamma.parser_demo`), 28

E

`envidict()` (`pyroSAR.gamma.ISPPar` method), 20
`error_stat()` (in module `pyroSAR.gamma.parser_demo`), 65
`ERS_ASF_SLC()` (in module `pyroSAR.gamma.parser_demo`), 29
`ERS_ESA_PRI()` (in module `pyroSAR.gamma.parser_demo`), 29
`ERS_ESA_SLC()` (in module `pyroSAR.gamma.parser_demo`), 29
`ESA` (class in `pyroSAR.drivers`), 7
`examine()` (`pyroSAR.drivers.ID` method), 7
`export2dict()` (`pyroSAR.drivers.ID` method), 8
`export2shp()` (`pyroSAR.drivers.Archive` method), 2
`export2sqlite()` (`pyroSAR.drivers.ID` method), 8
`export_indexing_yaml()` (`pyroSAR.datacube_util.Product` method), 132
`export_ingestion_yaml()` (`pyroSAR.datacube_util.Product` method), 132

F

`filenames` (`pyroSAR.datacube_util.Dataset` attribute), 131
`fill()` (in module `pyroSAR.gamma.srtm`), 24
`fill_gaps()` (in module `pyroSAR.gamma.parser_demo`), 66
`filter_processed()` (in module `pyroSAR.drivers`), 11
`filter_scenelist()` (`pyroSAR.drivers.Archive` method), 3
`find_datasets()` (in module `pyroSAR.ancillary`), 133
`findfiles()` (in module `pyroSAR.drivers`), 12
`findfiles()` (`pyroSAR.drivers.ID` method), 8
`fspf()` (in module `pyroSAR.gamma.parser_demo`), 67

G

`gcp_phase()` (in module `pyroSAR.gamma.parser_demo`), 68
`gdalinfo()` (`pyroSAR.drivers.ID` method), 8
`geocode()` (in module `pyroSAR.gamma`), 17
`geocode()` (in module `pyroSAR.snap.util`), 15
`get_colnames()` (`pyroSAR.drivers.Archive` method), 3
`get_GAMMA_RASTER()` (in module `pyroSAR.gamma.parser_demo`), 68
`get_tablenames()` (`pyroSAR.drivers.Archive` method), 3

`get_unique_directories()` (`pyroSAR.drivers.Archive` method), 3
`getCorners()` (`pyroSAR.drivers.CEOS_ERS` method), 5
`getCorners()` (`pyroSAR.drivers.CEOS_PSR` method), 6
`getCorners()` (`pyroSAR.drivers.ESA` method), 7
`getCorners()` (`pyroSAR.drivers.ID` method), 8
`getCorners()` (`pyroSAR.drivers.SAFE` method), 10
`getCorners()` (`pyroSAR.drivers.TSX` method), 11
`getFileObj()` (in module `pyroSAR.drivers`), 12
`getFileObj()` (`pyroSAR.drivers.ID` method), 8
`getGammaImages()` (`pyroSAR.drivers.ID` method), 8
`getHGT()` (`pyroSAR.drivers.ID` method), 8
`getLocals()` (`pyroSAR.SI.OSV` method), 126
`getOSV()` (`pyroSAR.drivers.SAFE` method), 10
`grasses()` (in module `pyroSAR.gamma.parser_demo`), 69
`GRD_to_SR()` (in module `pyroSAR.gamma.parser_demo`), 29
`groupby()` (in module `pyroSAR.ancillary`), 134
`groupbyTime()` (in module `pyroSAR.ancillary`), 134

H

`hgt()` (in module `pyroSAR.gamma.srtm`), 24
`hgt_collect()` (in module `pyroSAR.gamma.srtm`), 25
`hgt_map()` (in module `pyroSAR.gamma.parser_demo`), 69

I

`ID` (class in `pyroSAR.drivers`), 7
`identifier` (`pyroSAR.datacube_util.Dataset` attribute), 131
`identify()` (in module `pyroSAR.drivers`), 12
`identify_many()` (in module `pyroSAR.drivers`), 12
`image_stat()` (in module `pyroSAR.gamma.parser_demo`), 70
`import_outdated()` (`pyroSAR.drivers.Archive` method), 3
`init_offset()` (in module `pyroSAR.gamma.parser_demo`), 70
`init_offset_orbit()` (in module `pyroSAR.gamma.parser_demo`), 71
`insert()` (`pyroSAR.drivers.Archive` method), 3
`interf_SLC()` (in module `pyroSAR.gamma.parser_demo`), 72
`interp_ad()` (in module `pyroSAR.gamma.parser_demo`), 73
`INTF_SLC()` (in module `pyroSAR.gamma.parser_demo`), 30
`ionosphere_check()` (in module `pyroSAR.gamma.parser_demo`), 73
`is_processed()` (`pyroSAR.drivers.ID` method), 9

`is_registered()` (*pyroSAR.drivers.Archive method*), 3
`ISPPar` (class in *pyroSAR.gamma*), 20

K

`keys` (*pyroSAR.gamma.ISPPar attribute*), 20

L

`led_filename` (*pyroSAR.drivers.CEOS_PSR attribute*), 6

M

`make_tab()` (in module *pyroSAR.gamma.parser_demo*), 74
`makeSRTM()` (in module *pyroSAR.gamma.srtm*), 25
`mask_data()` (in module *pyroSAR.gamma.parser_demo*), 74
`match()` (*pyroSAR.S1.OSV method*), 126
`maxdate()` (*pyroSAR.S1.OSV method*), 126
`mcf()` (in module *pyroSAR.gamma.parser_demo*), 75
`measurements` (*pyroSAR.datacube_util.Product attribute*), 132
`mindate()` (*pyroSAR.S1.OSV method*), 126
`mk_ptarg()` (in module *pyroSAR.gamma.parser_demo*), 76
`mk_ptarg_cal()` (in module *pyroSAR.gamma.parser_demo*), 76
`mk_tab3()` (in module *pyroSAR.gamma.parser_demo*), 77
`MLI_cat()` (in module *pyroSAR.gamma.parser_demo*), 31
`MLI_copy()` (in module *pyroSAR.gamma.parser_demo*), 31
`mosaic()` (in module *pyroSAR.gamma.srtm*), 25
`mosaic_WB()` (in module *pyroSAR.gamma.parser_demo*), 77
`move()` (*pyroSAR.drivers.Archive method*), 3
`multi_cpx()` (in module *pyroSAR.gamma.parser_demo*), 79
`multi_look()` (in module *pyroSAR.gamma.parser_demo*), 80
`multi_look2()` (in module *pyroSAR.gamma.parser_demo*), 80
`multi_look_MLI()` (in module *pyroSAR.gamma.parser_demo*), 81
`multi_real()` (in module *pyroSAR.gamma.parser_demo*), 81
`multi_S1_TOPS()` (in module *pyroSAR.gamma.parser_demo*), 78
`multi_SLC_WSS()` (in module *pyroSAR.gamma.parser_demo*), 79
`multilook()` (in module *pyroSAR.gamma*), 22

N

`neutron()` (in module *pyroSAR.gamma.parser_demo*), 82

O

`offset_add()` (in module *pyroSAR.gamma.parser_demo*), 84
`offset_fit()` (in module *pyroSAR.gamma.parser_demo*), 84
`offset_plot_az()` (in module *pyroSAR.gamma.parser_demo*), 85
`offset_plot_r()` (in module *pyroSAR.gamma.parser_demo*), 85
`offset_pwr()` (in module *pyroSAR.gamma.parser_demo*), 85
`offset_pwr_tracking()` (in module *pyroSAR.gamma.parser_demo*), 87
`offset_pwr_tracking2()` (in module *pyroSAR.gamma.parser_demo*), 88
`offset_SLC()` (in module *pyroSAR.gamma.parser_demo*), 82
`offset_SLC_tracking()` (in module *pyroSAR.gamma.parser_demo*), 83
`offset_sub()` (in module *pyroSAR.gamma.parser_demo*), 90
`offset_tracking()` (in module *pyroSAR.gamma.parser_demo*), 90
`OPOD_vec()` (in module *pyroSAR.gamma.parser_demo*), 32
`ORB_filt()` (in module *pyroSAR.gamma.parser_demo*), 32
`ORB_prop_SLC()` (in module *pyroSAR.gamma.parser_demo*), 32
`ORRM_vec()` (in module *pyroSAR.gamma.parser_demo*), 33
`OSV` (class in *pyroSAR.S1*), 125
`outname_base()` (*pyroSAR.drivers.ID method*), 9
`ovs()` (in module *pyroSAR.gamma*), 21

P

`par2hdr()` (in module *pyroSAR.gamma*), 22
`par_ACS_ERS()` (in module *pyroSAR.gamma.parser_demo*), 91
`par_ASAR()` (in module *pyroSAR.gamma.parser_demo*), 92
`par_ASF_91()` (in module *pyroSAR.gamma.parser_demo*), 92
`par_ASF_96()` (in module *pyroSAR.gamma.parser_demo*), 92
`par_ASF_PRI()` (in module *pyroSAR.gamma.parser_demo*), 93
`par_ASF_RSAT_SS()` (in module *pyroSAR.gamma.parser_demo*), 93

<code>par_ASF_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 93	<code>py- par_SIRC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 105	<code>py-</code>
<code>par_ATLSCI_ERS()</code> (in module <code>roSAR.gamma.parser_demo</code>), 94	<code>py- par_TX_GRD()</code> (in module <code>roSAR.gamma.parser_demo</code>), 106	<code>py-</code>
<code>par_CS_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 94	<code>py- par_TX_ScanSAR()</code> (in module <code>roSAR.gamma.parser_demo</code>), 107	<code>py-</code>
<code>par_CS_SLC_TIF()</code> (in module <code>roSAR.gamma.parser_demo</code>), 94	<code>py- par_TX_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 106	<code>py-</code>
<code>par_EORC_JERS_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 95	<code>py- par_UAVSAR_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 107	<code>py-</code>
<code>par_EORC_PALSAR()</code> (in module <code>roSAR.gamma.parser_demo</code>), 95	<code>py- parse_command()</code> (in module <code>roSAR.gamma.parser</code>), 26	<code>py-</code>
<code>par_ERSDAC_PALSAR()</code> (in module <code>roSAR.gamma.parser_demo</code>), 95	<code>py- parse_datasetname()</code> (in module <code>roSAR.ancillary</code>), 134	<code>py-</code>
<code>par_ESA_ERS()</code> (in module <code>roSAR.gamma.parser_demo</code>), 96	<code>py- parse_date()</code> (in module <code>pyroSAR.drivers</code>), 12	
<code>par_GF3_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 96	<code>py- parse_date()</code> (<code>pyroSAR.drivers.ID</code> static method), 9	
<code>par_IECAS_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 96	<code>py- parse_module()</code> (in module <code>roSAR.gamma.parser</code>), 27	<code>py-</code>
<code>par_KC_PALSAR_slr()</code> (in module <code>roSAR.gamma.parser_demo</code>), 97	<code>py- ph_slope_base()</code> (in module <code>roSAR.gamma.parser_demo</code>), 108	<code>py-</code>
<code>par_KS_DGM()</code> (in module <code>roSAR.gamma.parser_demo</code>), 97	<code>py- phase_slope()</code> (in module <code>roSAR.gamma.parser_demo</code>), 108	<code>py-</code>
<code>par_KS_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 98	<code>py- PRC_vec()</code> (in module <code>pyroSAR.gamma.parser_demo</code>), 33	
<code>par_MSP()</code> (in module <code>pyroSAR.gamma.parser_demo</code>), 98	<code>py- process()</code> (in module <code>pyroSAR.gamma</code>), 21	
<code>par_PRI()</code> (in module <code>pyroSAR.gamma.parser_demo</code>), 99	<code>Product</code> (class in <code>pyroSAR.datacube_util</code>), 132	
<code>par_PRI_ESRIN_JERS()</code> (in module <code>roSAR.gamma.parser_demo</code>), 99	<code>ptarg_cal_MLI()</code> (in module <code>roSAR.gamma.parser_demo</code>), 109	<code>py-</code>
<code>par_PulsAR()</code> (in module <code>roSAR.gamma.parser_demo</code>), 99	<code>ptarg_cal_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 110	<code>py-</code>
<code>par_RISAT_GRD()</code> (in module <code>roSAR.gamma.parser_demo</code>), 100	<code>py- ptarg_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 109	<code>py-</code>
<code>par_RISAT_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 100	<code>py- pwr2ras()</code> (in module <code>pyroSAR.gamma.parser_demo</code>), 111	
<code>par_RSAT2_SG()</code> (in module <code>roSAR.gamma.parser_demo</code>), 101	<code>py- pyroSAR.ancillary</code> (module), 133	
<code>par_RSAT2_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 102	<code>pyroSAR.auxdata</code> (module), 129	
<code>par_RSAT_SCW()</code> (in module <code>roSAR.gamma.parser_demo</code>), 102	<code>py- pyroSAR.datacube_util</code> (module), 131	
<code>par_RSAT_SGF()</code> (in module <code>roSAR.gamma.parser_demo</code>), 103	<code>pyroSAR.drivers</code> (module), 1	
<code>par_RSAT_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 103	<code>py- pyroSAR.gamma</code> (module), 17	
<code>par_RSI_ERS()</code> (in module <code>roSAR.gamma.parser_demo</code>), 103	<code>pyroSAR.gamma.parser</code> (module), 26	
<code>par_S1_GRD()</code> (in module <code>roSAR.gamma.parser_demo</code>), 104	<code>py- pyroSAR.gamma.parser_demo</code> (module), 27	
<code>par_S1_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 105	<code>pyroSAR.gamma.srtm</code> (module), 23	
	<code>py- pyroSAR.S1</code> (module), 125	
	<code>pyroSAR.snap.util</code> (module), 15	
	<code>py-</code>	
	R	
	<code>py- radcal_MLI()</code> (in module <code>roSAR.gamma.parser_demo</code>), 112	<code>py-</code>
	<code>py- radcal_PRI()</code> (in module <code>roSAR.gamma.parser_demo</code>), 113	<code>py-</code>
	<code>py- radcal_pwr_stat()</code> (in module <code>roSAR.gamma.parser_demo</code>), 115	<code>py-</code>
	<code>py- radcal_SLC()</code> (in module <code>roSAR.gamma.parser_demo</code>), 114	<code>py-</code>

`rascc_mask()` (in module `pyroSAR.gamma.parser_demo`), 115
`rascc_mask_thinning()` (in module `pyroSAR.gamma.parser_demo`), 116
`removeGRDBorderNoise()` (in module `pyroSAR.S1`), 127
`removeGRDBorderNoise()` (`pyroSAR.drivers.SAFE` method), 10
`res_map()` (in module `pyroSAR.gamma.parser_demo`), 117
`residue()` (in module `pyroSAR.gamma.parser_demo`), 117
`residue_cc()` (in module `pyroSAR.gamma.parser_demo`), 118
`retrieve()` (`pyroSAR.S1.OSV` method), 126
`RSAT2_vec()` (in module `pyroSAR.gamma.parser_demo`), 34
`run_all()` (in module `pyroSAR.gamma.parser_demo`), 118

S

`S1_BURST_tab()` (in module `pyroSAR.gamma.parser_demo`), 34
`S1_BURST_tab_from_zipfile()` (in module `pyroSAR.gamma.parser_demo`), 34
`S1_burstloc()` (in module `pyroSAR.gamma.parser_demo`), 37
`S1_deburst()` (in module `pyroSAR.gamma`), 21
`S1_extract_png()` (in module `pyroSAR.gamma.parser_demo`), 37
`S1_GRD_preproc()` (in module `pyroSAR.gamma.parser_demo`), 35
`S1_import_SLC_from_zipfiles()` (in module `pyroSAR.gamma.parser_demo`), 37
`S1_OPOD_vec()` (in module `pyroSAR.gamma.parser_demo`), 35
`S1_TOPS_preproc()` (in module `pyroSAR.gamma.parser_demo`), 36
`SAFE` (class in `pyroSAR.drivers`), 9
`sbi_filt()` (in module `pyroSAR.gamma.parser_demo`), 118
`SBI_INT()` (in module `pyroSAR.gamma.parser_demo`), 38
`sbi_offset()` (in module `pyroSAR.gamma.parser_demo`), 119
`scanMetadata()` (`pyroSAR.drivers.CEOS_ERS` method), 5
`scanMetadata()` (`pyroSAR.drivers.CEOS_PSR` method), 6
`scanMetadata()` (`pyroSAR.drivers.ESA` method), 7
`scanMetadata()` (`pyroSAR.drivers.ID` method), 9
`scanMetadata()` (`pyroSAR.drivers.SAFE` method), 10
`scanMetadata()` (`pyroSAR.drivers.TSX` method), 11
`seconds()` (in module `pyroSAR.ancillary`), 134
`select()` (`pyroSAR.drivers.Archive` method), 4
`select_duplicates()` (`pyroSAR.drivers.Archive` method), 4
`size` (`pyroSAR.drivers.Archive` attribute), 4
`slant_range()` (in module `pyroSAR.gamma.parser_demo`), 120
`SLC_adf()` (in module `pyroSAR.gamma.parser_demo`), 39
`SLC_burst_copy()` (in module `pyroSAR.gamma.parser_demo`), 39
`SLC_burst_corners()` (in module `pyroSAR.gamma.parser_demo`), 40
`SLC_cat()` (in module `pyroSAR.gamma.parser_demo`), 41
`SLC_cat_S1_TOPS()` (in module `pyroSAR.gamma.parser_demo`), 41
`SLC_copy()` (in module `pyroSAR.gamma.parser_demo`), 42
`SLC_copy_S1_TOPS()` (in module `pyroSAR.gamma.parser_demo`), 43
`SLC_copy_WB()` (in module `pyroSAR.gamma.parser_demo`), 43
`SLC_corners()` (in module `pyroSAR.gamma.parser_demo`), 44
`SLC_deramp()` (in module `pyroSAR.gamma.parser_demo`), 44
`SLC_deramp_S1_TOPS()` (in module `pyroSAR.gamma.parser_demo`), 45
`SLC_freq_shift()` (in module `pyroSAR.gamma.parser_demo`), 45
`SLC_interp()` (in module `pyroSAR.gamma.parser_demo`), 46
`SLC_interp_map()` (in module `pyroSAR.gamma.parser_demo`), 47
`SLC_interp_S1_TOPS()` (in module `pyroSAR.gamma.parser_demo`), 46
`SLC_intf()` (in module `pyroSAR.gamma.parser_demo`), 48
`SLC_mosaic_S1_TOPS()` (in module `pyroSAR.gamma.parser_demo`), 49
`SLC_ovr()` (in module `pyroSAR.gamma.parser_demo`), 50
`SLC_ovr2()` (in module `pyroSAR.gamma.parser_demo`), 50
`SLC_phase_shift()` (in module `pyroSAR.gamma.parser_demo`), 51
`sortByDate()` (`pyroSAR.S1.OSV` method), 126
`split_WB()` (in module `pyroSAR.gamma.parser_demo`), 120
`SR_to_GRD()` (in module `pyroSAR.gamma.parser_demo`), 51
`subtract_phase()` (in module `pyroSAR.gamma.parser_demo`), 120

`summary()` (*pyroSAR.drivers.ID method*), 9
`swap()` (*in module pyroSAR.gamma.srtm*), 25

T

`tree_cc()` (*in module pyroSAR.gamma.parser_demo*), 121
`tree_gzw()` (*in module pyroSAR.gamma.parser_demo*), 121
`TSX` (*class in pyroSAR.drivers*), 10
`TX_SLC_preproc()` (*in module pyroSAR.gamma.parser_demo*), 52

U

`units` (*pyroSAR.datacube_util.Dataset attribute*), 131
`unpack()` (*pyroSAR.drivers.CEOS_ERS method*), 5
`unpack()` (*pyroSAR.drivers.CEOS_PSR method*), 6
`unpack()` (*pyroSAR.drivers.ESA method*), 7
`unpack()` (*pyroSAR.drivers.ID method*), 9
`unpack()` (*pyroSAR.drivers.SAFE method*), 10
`unpack()` (*pyroSAR.drivers.TSX method*), 11
`unw_correction_filt()` (*in module pyroSAR.gamma.parser_demo*), 122
`unw_correction_poly()` (*in module pyroSAR.gamma.parser_demo*), 122
`unw_model()` (*in module pyroSAR.gamma.parser_demo*), 122
`UNWRAP()` (*in module pyroSAR.gamma.parser_demo*), 52
`UNWRAP_PAR()` (*in module pyroSAR.gamma.parser_demo*), 53
`update()` (*pyroSAR.SI.OSV method*), 127
`UTM` (*class in pyroSAR.gamma*), 23

W

`write()` (*pyroSAR.datacube_util.Product method*), 132