BIOMASS L2
Prototype Processor

User Manual

Document REF	ARE-017082
Issue	1.1
Date	23 July 2020
Pages	28
Recipients	Klaus Scipal

	Name	**Signature**
Prepared by	Emanuele Giorgi	
Checked by	Francesco Banda	
Approved by	Davide Giudici	

ARESYS HAS A CERTIFIED QUALITY MANAGEMENT SYSTEM

TÜVRheinland CERTIFIED Management System ISO 9001:2008 www.tuv.com ID 9105081902

ACCREDIA L'ENTE ITALIANO DI ACCREDITAMENTO

SUMMARY

1 Introduction and overview

1.1 Scope

The objective of this document is to describe the package content, the installation procedure and the usage of the BIOMASSL2 prototype processor.

1.2 Applicable documents

[AD1] EOP-SM/3001 Statement of Work issue 0, rev 6

[AD2] ARE-007210 BIOMASS L2 implementation study - ARESYS proposal, issue 1.1, 17 Feb. 2017.

1.3 Reference documents

[RD1] ARE-15507 BIOMASS L2 ICD Document, v2.3, 19-06-2020

[RD2] ARE-11567, BIOMASS L2 ATBD v1.4, 19-06-2020

[RD3] ARE-10739, BIOMASS L2 ARD Algorithm Requirement Document, v2.6, 19-06-2020

[RD4] ARE-15529, BIOMASS L2 Verification Report, v1.0, 19-06-2020

[RD5] ESA (2015), Biomass Mission Requirements Document, EOP-SM/1645

[RD6] Asner et al, A universal airborne LiDAR approach for tropical forest carbon mapping, Oecologia 2012

[RD7] Albinet, C., Nouvellon, S., Frommknecht, B., Rutakaza, R., Daniel, S., and Saüt, C.: MAAP: The Mission Algorithm and Analysis Platform: A New Virtual and Collaborative Environment for the Scientific Community, EGU General Assembly 2020, Online, 4–8 May 2020, EGU2020-19989, https://doi.org/10.5194/egusphere-egu2020-19989, 2020

[RD8] The Equi7Grid – V13, Grid and Tiling Definition Document – Issue 0.6, Bernhard Bauer-Marschallinger, Christoph Paulik, Senmao Cao, May 3, 2019

[RD9] M. Martone, P. Rizzoli, C. Wecklich, C. Gonzlez, J.L. Bueso-Bello, P. Valdo, D. Schulze, M. Zink, G. Krieger, and A. Moreira, “The Global Forest/Non-Forest Map from TanDEM-X Interferometric SAR Data”, Remote Sensing of Environment, vol. 205, pp. 352-373, Feb. 2018.

[RD10] T. Esch, W. Heldens, A. Hirner, M. Keil, M. Marconcini, A. Roth, J. Zeidler, S. Dech, and E. Strano, “Breaking new ground in mapping human settlements from space - the global urban footprint”, ISPRS Journal of Photogrammetry and Remote Sensing, no. 134, pp. 30-42, Dez 2017.

1.4 Definitions

FH	height maps
DEM	Digital Elevation Model
kz	Vertical Wave Number
ASC	Ascending
DSC	Descending
DTM	Digital Terrain Model
ESA	European Space Agency
SCS	Single Look Complex Slant Range Image
Geotiff	Georeferenced Tagged Image File Format
ROI	region-of-interest
ICD	Interface Control Document
FD	disturbance maps
AGB	The above ground biomass
TOMO	tomographic voxels
FNF	Forest/Non-forest map
L2	Level-2
SAR	Synthetic Aperture Radar
TBD	to be defined
XML	eXtended Markup Language
NAN	Not A Number
CAL	Calibration

2 Processor overview

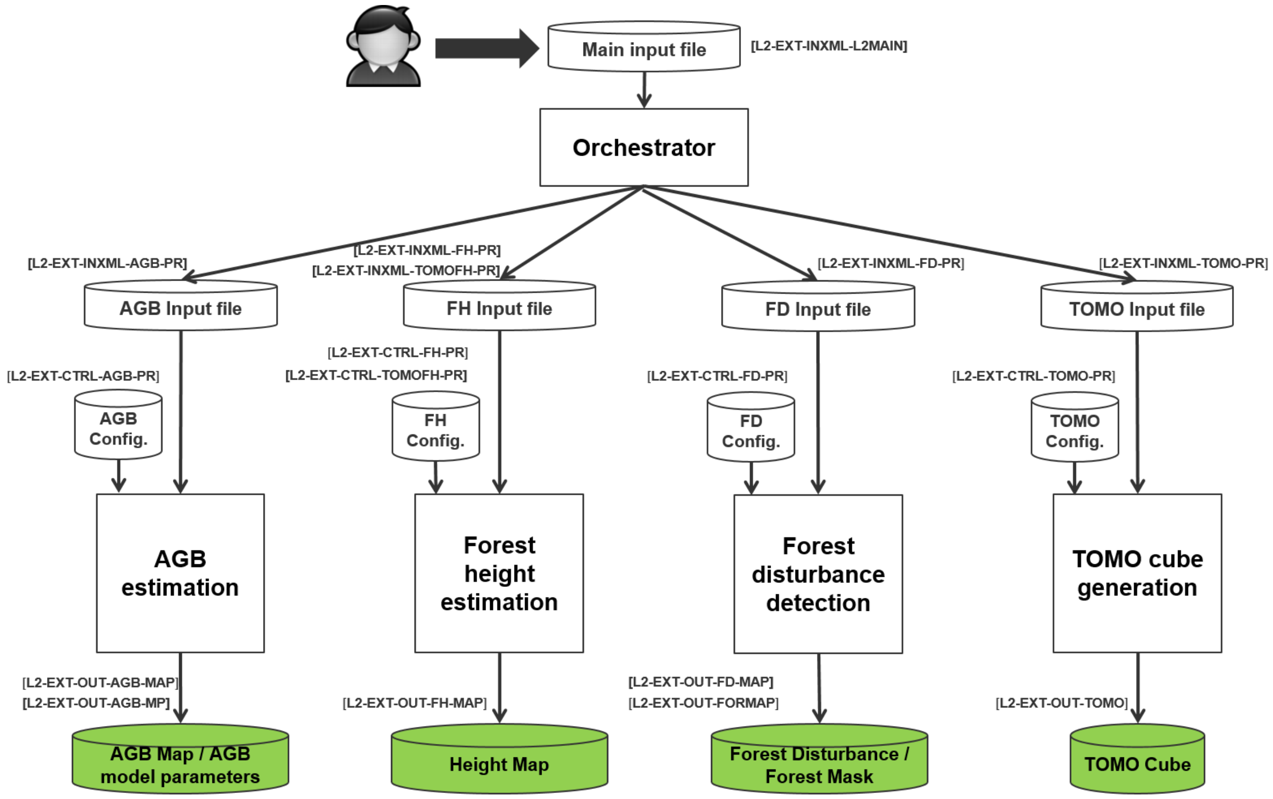


Fig.1 Overall view of the L2 processor software [RD1].

The Biomass L2 Processor is a software suite in charge of processing L1 stacks of coregistered and calibrated SCS data to provide L2 geophysical products (TOMO= Tomographic product, FD= Forest Disturbance, FH= Forest Height, AGB= Above Ground Biomass), as defined in the MRD [RD5]. In particular see Fig.1: the *Orchestrator* receives as input the spatial and temporal Region Of Interest from the User, this is a geographical region defined by a polynomial area and by start/stop timing boundaries according to the interface specified in [RD1]. For instance the geographical region may be a certain biome area or a continent: in particular this software supports only one continental zone at a time, as defined in the Equi7 grid (see [RD8]). The Orchestrator prepares the input file for the four specific processing chains that will retrieve the BIOMASS geophysical products over the full requested ROI.

In the Tomographic Mission Phase, the *TOMO processor* performs tomographic processing to produce 3-D data cubes [RD3].

The *FD processor* receives as input Forest/Non Forest mask and, according to the input stack of SAR data updates it generating an updated FNF map and a disturbance map.

The *FH processor* receives as input a Forest/Non Forest mask (potentially generated by the FD processor) and according to the input stack of SAR data estimates the Forest Height (FH) product. In the TOMO phase a specific tomographic algorithm can be used to process tomographic stacks for retrieving FH as described in [RD1].

The *AGB processor* receives as input a Forest/Non Forest mask (potentially generated by the FD processor) and according to the input stack of SAR data, estimates the Above Ground Biomass (AGB) product. Since AGB and FH are not independent parameters for a certain tree [RD6], FH might also be provided to the AGB processor as an optional input. In this case a consistency map between estimated AGB and FH shall also be generated by the AGB processor. Although the interface is part of the prototype processor the computation of the consistency map is not available being still an open research topic (the algorithm is TBD in future projects).

3 Software overview

3.1 Package description and contents

Two type of package organizations are available and both will be detailed in this manual:

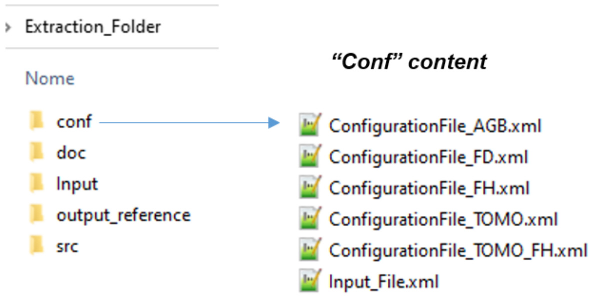
- the default package is a complete suite containing all the processing chains together

- the alternative package is the one specifically organized to be used in the *proto-MAAP* platform [RD7]: its organization consists in five separated and self-consistent suites, each containing only all of the code and inputs needed by each of the five processing chains alone (AGB, FD, FH, TOMO FH, TOMO)

In both cases the software can be delivered in a single compressed tar.gz package, or in multiple tar.gz files package: the content of the tar.gz file / files after extraction in a common folder (see section 4 for the extraction procedure) is described in 3.1.1 for the default package and in 3.1.2 for the alternative *proto-MAAP* ready package.

3.1.1 Package content for the default complete suite organization

The package content after the tar.gz file / files extraction (see section 4) is the following, see Fig.2



Extraction_Folder

Nome

- conf
- doc
- Input
- output_reference
- src

"Conf" content

- ConfigurationFile_AGB.xml
- ConfigurationFile_FD.xml
- ConfigurationFile_FH.xml
- ConfigurationFile_TOMO.xml
- ConfigurationFile_TOMO_FH.xml
- Input_File.xml

Fig.2 Default complete suite package content after tar.gz extraction

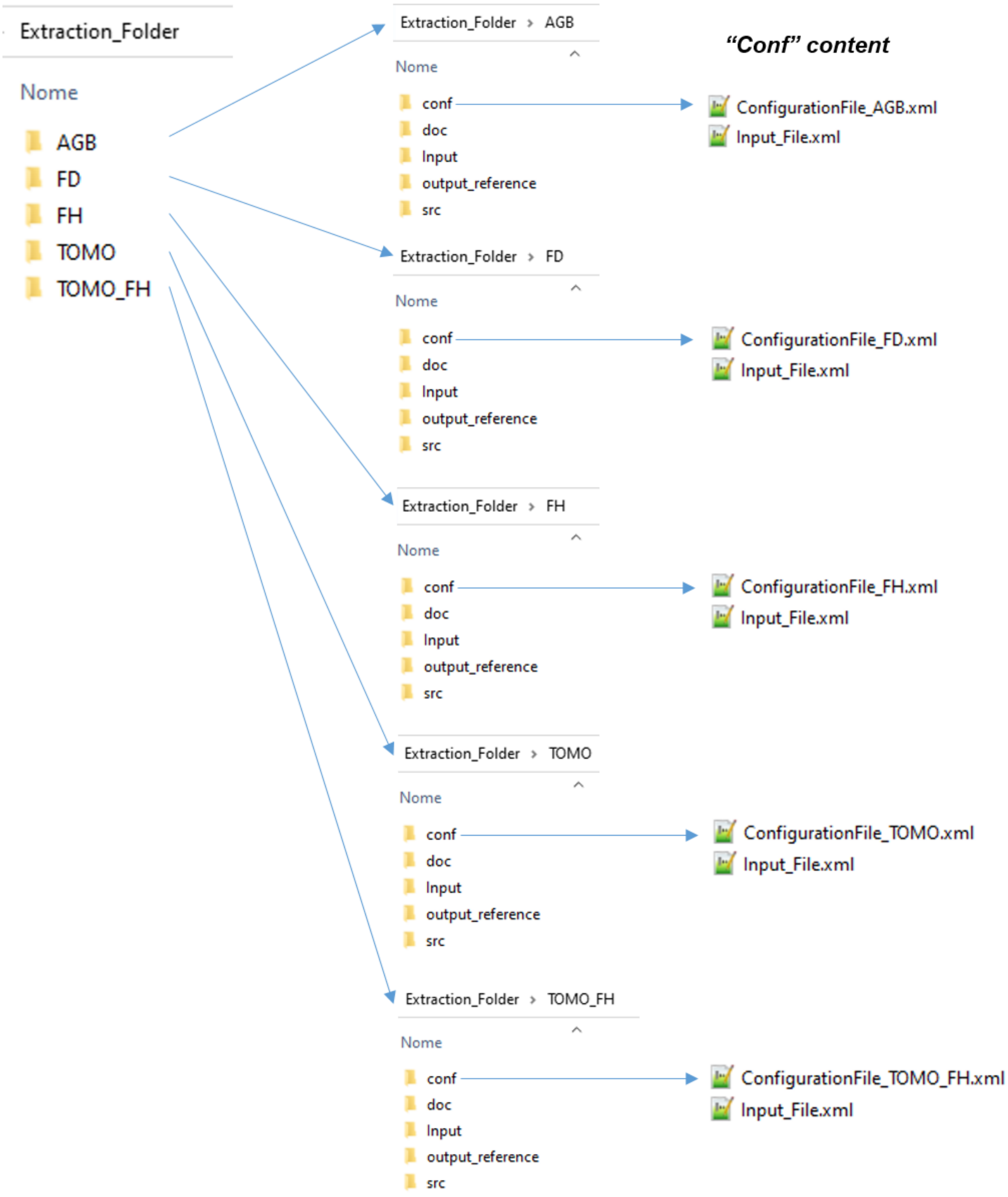
The folders of Fig.2 are described here:

- `conf:` contains (both described in [RD1] and explained in section 5.1)
  - the configuration files for all the chains, `ConfigurationFile_AGB.xml`, `ConfigurationFile_FH.xml`, `ConfigurationFile_FD.xml`, `ConfigurationFile_TOMO.xml` and `ConfigurationFile_TOMO_FH.xml`
  - the BiomassL2 prototype processor input file `Input_File.xml`
- `doc:` contains documentation as a `ReadMe_BiomassL2_Prototype.txt` for package content and quick installation guide and a `documentation.md` file with brief description of the algorithms
- `Input:` contains input data to be processed by the suite (L1 stacks of coregistered and calibrated SCS data), and auxiliary needed data as DEM, Geometry (if the `ComputeGeometry` flag is disabled, see 5.1.2) along with specific chain ones (see section 5)
- `output_reference:` contains output processed reference data; the content is only demonstrative and contains the products which can be generated with the default `conf` and `input` content
- `src:` contains all the source code files and the py script file used to launch the processor.

For L1 demo data list and formats see [RD4] and [RD1].

3.1.2 Package content for the alternative proto-MAAP separated suites organization

The package content after the tar.gz file / files extraction (see section 4) is composed as the sub-folder tree shown in following Fig.3



Extraction_Folder

Nome
- AGB
- FD
- FH
- TOMO
- TOMO_FH

Extraction_Folder › AGB

Nome
- conf
- doc
- Input
- output_reference
- src

Extraction_Folder › FD

Nome
- conf
- doc
- Input
- output_reference
- src

Extraction_Folder › FH

Nome
- conf
- doc
- Input
- output_reference
- src

Extraction_Folder › TOMO

Nome
- conf
- doc
- Input
- output_reference
- src

Extraction_Folder › TOMO_FH

Nome
- conf
- doc
- Input
- output_reference
- src

"Conf" content

- ConfigurationFile_AGB.xml
- Input_File.xml

- ConfigurationFile_FD.xml
- Input_File.xml

- ConfigurationFile_FH.xml
- Input_File.xml

- ConfigurationFile_TOMO.xml
- Input_File.xml

- ConfigurationFile_TOMO_FH.xml
- Input_File.xml

Fig.3 Alternative proto-MAAP package content after tar.gz extraction

The main folder in Fig.3 is composed by five subfolders, one for each processing chain (AGB, FH, TOMO FH, FD and TOMO), each is composed by the same sub-folders structure described here once:

- `conf`: contains (both described in [RD1] and explained in section 5.1)
  - the configuration file for the specific chain, `ConfigurationFile_XXX.xml` (where `XXX` is AGB, FH, TOMO FH, FD or TOMO depending on the chain folder)
  - the BiomassL2 prototype processor input file `Input_File.xml` (the same file is replicated in each chain `conf` folder, with the only difference that only the current chain processing is enabled with the proper flag, see 5.1.1)
- `doc`: contains documentation as a `ReadMe_BiomassL2_Prototype.txt` for package content and quick installation guide and a `documentation.md` file with brief description of the current algorithm
- `Input`: contains input data to be processed by the suite (L1 stacks of coregistered and calibrated SCS data), and auxiliary needed data as DEM, Geometry (if the `ComputeGeometry` flag is disabled, see 5.1.2) along with specific chain ones (see section 5).
- `output_reference`: contains output processed reference data; the content is only demonstrative and contains the products which can be generated with the default `conf` and `input` content, for the specific chain (AGB, FH, TOMO FH, FD or TOMO)
- `src`: contains only the source code needed files and the script files used to launch only the specific chain (AGB, FH, TOMO FH, FD or TOMO)

For L1 demo data list and formats see [RD4] and [RD1].

3.2 Software dependencies

All of the BiomassL2 prototype code is in Python language.

To execute the processor those are the required dependences, meaning that they are supposed to be already part of the Python environment used for processing, before proceeding with the installation described in section 4:

- Python 3.7 environment, with following packages:

Equi7Grid	0.0.10
Pytileproj	0.0.12 (this is a sub-dependence of the above Equi7Grid)
GeoPandas	0.7.0
GDAL	2.3.3
lxmx	4.4.1
Matplotlib	2.2.2
Namedlist	1.7
Numpy	1.15.4
Numpydoc	0.8.0
Packaging	19.0
Pyproj	2.1.1
PyXB	1.2.6
Scipy	1.4.1
Shapely	1.6.4.post2

3.3 Hardware and software requirements

HW requirements

The minimum hardware requirements are:

- CPU: no specific requirement, but consider to use a CPU with good performances in single core / single thread, since the prototype processor does not take advantage of multi core /multi threads architectures.
- Memory (RAM): 40 GB or more
- Storage: about 100GB is the storage requested for the installation and for the provided demonstrative launch

SW requirements

The software requirements are:

Operating System: Linux x64

Processor tested under following Linux environment:

- Operating System: CentOS Linux 7 (Core)
- Kernel: Linux 3.10.0-957.1.3.el7.x86_64
- Architecture: x86-64

4 Installation

4.1 Installation procedure

1) Extract all the tar.gz provided (single or multiple tar.gz files, see 3.1) in a common folder, with a command like the following for each of the tar.gz files, maintaining the same *InstallationFolder*:

```
tar -zxvf current_file.tar.gz -C Out InstallationFolder outputFolder
```

Substituting "*current_file.tar.gz*" with each of the files provided and "*InstallationFolder*" with the desired output installation directory.

The content of the "*InstallationFolder*" after extraction is described in 3.1 and shown in previous chapter at Fig.2 or Fig.3

2)

a. In case of default complete suite organization (see 3.1.1)

Open the xml input file (see [RD1]) stored in the `conf` folder:
InstallationFolder`/conf/Input_File.xml`
(where "*InstallationFolder*" is the folder where the tar.gz have been extracted)

Insert, in the opened xml file, your current directory locations with absolute paths for:

- `L1cRepository`: this should point over the *InstallationFolder*`/input/dataSet` folder
- `AuxiliaryProductsFolder`: this should point over the *InstallationFolder*`/input/auxiliary_data_pf` folder
- `OutputFolder`: is the folder where the output will be generated (note that at each launch a sub-folder with the current date is generated, so any folder will never be overwritten, as better explained in section 5)

Open the py run script stored in the `src` folder:
InstallationFolder`/src/run_biomassL2_processor.py`

Insert, in the opened py file, your current directory locations with absolute paths for:

- `input_file_xml`: is the path of the *InstallationFolder*`/conf/Input_File.xml` file

- gdal_path: this is the folder containing the GDAL executables, usually in the "/bin" sub-folder of GDAL environment (containing e.g. "gdalwarp", "gdal_translate", ...)
- gdal_enviroment_path: this is the GDAL_DATA environment variable path

b. In case of alternative proto-MAAP separated suites organization (see 3.1.2)

Open each of the following xml input files stored in the conf folder of each chain:

```
InstallationFolder/AGB/conf/Input_File.xml
InstallationFolder/FD/conf/Input_File.xml
InstallationFolder/FH/conf/Input_File.xml
InstallationFolder/TOMO/conf/Input_File.xml
InstallationFolder/TOMO_FH/conf/Input_File.xml
```

(where "*InstallationFolder*" is the folder where the tar.gz have been extracted)

Insert, for all the opened xml files, your current directory locations with absolute paths for:

- L1cRepository: this should point over the *InstallationFolder*/XXX/input/dataSet folder of each chain (where "XXX" can be "AGB", "FD", "FH", "TOMO" and "TOMO_FH")
- AuxiliaryProductsFolder: this should point over the *InstallationFolder*/XXX/input/ auxiliary_data_pf folder of each chain (where "XXX" can be "AGB", "FD", "FH", "TOMO" and "TOMO_FH")
- OutputFolder: is the folder where the output will be generated (note that at each launch a sub-folder with the current date is generated, so any folder will never be overwritten, as better explained in section 5)

Open each of the py run script stored in the src folder of each chain:

```
InstallationFolder/AGB/src/run_biomassL2_processor_AGB.py
InstallationFolder/FD/src/run_biomassL2_processor_FD.py
InstallationFolder/FH/src/run_biomassL2_processor_FH.py
InstallationFolder/TOMO/src/run_biomassL2_processor_TOMO.py
InstallationFolder/TOMO_FH/src/run_biomassL2_processor_TOMO_.py
```

For each of the opened py file insert your current directory locations with absolute paths for:

- input_file_xml: is the path of the *InstallationFolder*/XXX/conf/Input_File.xml file of each chain (where "XXX" can be "AGB", "FD", "FH", "TOMO" and "TOMO_FH")
- gdal_path: this is the folder containing the GDAL executables, usually in the "/bin" sub-folder of GDAL environment (containing e.g. "gdalwarp", "gdal_translate", ...)
- gdal_enviroment_path: this is the GDAL_DATA environment variable path

4.2 Installation verification with demo launch

In order to verify the correct installation of the processor, it is possible to launch the demonstrative processing which will execute one demo for each chain (AGB, FD, FH, TOMO and TOMO_FH).

To execute the demo, run the following script in your python environment (in case of complete suite organization):

```
InstallationFolder/src/run_biomassL2_processor.py
```

To execute the demo for the alternative proto-MAAP suite organization, run all of the following scripts in your python environment (the chain order execution is not important for the demo):

```
InstallationFolder/AGB/src/run_biomassL2_processor_AGB.py
InstallationFolder/FD/src/run_biomassL2_processor_FD.py
InstallationFolder/FH/src/run_biomassL2_processor_FH.py
InstallationFolder/TOMO/src/run_biomassL2_processor_TOMO.py
InstallationFolder/TOMO_FH/src/run_biomassL2_processor_TOMO_FH.py
```

If no error occurs in the python environment command window during the demo, the installation can be considered done correctly (it may take some hours to end all of the demonstrative launches, see the python environment command window for messages about processes progress).

For more details on processor configuration, execution and outputs description please follow section 5.

4.3 Uninstall procedure

There is no need of any particular un-installation procedure, simply delete the `InstallationFolder` created during installation step 4.1. Beware that the `OutputFolder` chosen in step 4.1 may be internal to the `InstallationFolder`, in this case the output products will be erased too, so in this case be careful to move products away before un-install, if desired.

5 User Manual

In this section the User is guided in the processor usage, in particular, following topics will be discussed:

- how to configure the processor to run custom processing
- how to launch the processor
- description of the chain output and breakpoint products
- how to concatenate the processing flows to use output products of one as input to another, in order to reproduce "mission like" scenarios

The description will be slightly different in case of default vs alternative proto-MAAP organized package, each difference will be addressed.

5.1 Configure the processor

Inside the `conf` folder two types of xml files are present ([RD1]):

- `Input_File.xml:` this is the main BiomassL2 processor input file and it is used to enable the desired chain/s, to define the spatial and temporal Region Of Interest where to search for input data and to set input/output paths
- `Configuration_File_XXX.xml:` where "`XXX`" can be "AGB", "FD", "FH", "TOMO_FH" and "TOMO" is used to configure many aspects of the processing, different settings for different chains.

They are explained in more details here below.

5.1.1 Input File

The input file, in case of default suite organization, is present in
`InstallationFolder/conf/Input_File.xml`

The input file, in case of alternative proto-MAAP suite organization, is present in
`InstallationFolder/XXX/conf/Input_File.xml`
Where "`XXX`" can be "AGB", "FD", "FH", "TOMO_FH" and "TOMO" (one for each chain)

Fields of the input file are the following:

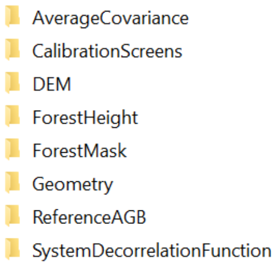
- ***L1cRepository:*** is the full path of the folder where the L1 stacks of coregistered and calibrated SCS data are present (for example the path for the default demo is `InstallationFolder/Input/dataSet`); note that The L1 stacks of data are in a specific "Aresys Product" format, the stacks are tagged inside their binary headers with UTC date and latitude / longitude boundaries used by the orchestrator to select only valid data inside the User defined temporal (see ***L1cDate*** below) and spatial (see ***GeographicBoundary*** below) Region Of Interest.
- ***L2Product:*** contains the Boolean flags of the chain/s to be enabled (AGB, FD, FH, TOMO_FH, TOMO)
  - In case of default suite, one or all the chains can be enabled simultaneously and the processor will execute them one after the other (if enabled), in the following order: AGB, than FD, than FH, than TOMO_FH, than TOMO. This order is arbitrary and each chain is independently launched with its own inputs: enabling simultaneously more than one chain does not permit to concatenate the output of the previous as input to the following in order to reproduce "mission like" scenarios; to do so it is needed to execute each chain separately and manually moving the output products of the previous to the input folders to the following, as explained in section 5.4)
  - In case of alternative proto-MAAP suite this flags section should not be modified, and for each specific chain, only the flag of the same chain should be enabled (see section 5.4 to concatenate outputs of previous chains as input to the following to reproduce "mission like" scenarios)
- ***L1cDate***: start and stop values of the dates, in UTC format: the orchestrator will use all and only the data in the ***L1cRepository*** which are inside this temporal Region Of Interest
- ***GeographicBoundary***: this is a list of a minimum of three Latitude / Longitude coordinates: the orchestrator will use only the data in the ***L1cRepository*** which are at least partially contained this bounding box); this software supports only one continental zone at a time, as defined in the Equi7 grid (see [RD8]).
- ***AuxiliaryProductsFolder***: is the full path of the folder where the auxiliary data are stored (for example the path for the default demo is `InstallationFolder/Input/auxiliary_data_pf`).

This is the list of mandatory and optional auxiliary needed: the names in this description is the name of the folder where they are stored, as shown in Fig.4

- mandatory auxiliary data for AGB: *DEM, ForestMask, ReferenceAGB*
- mandatory auxiliary data for FD: *DEM, ForestMask*
- mandatory auxiliary data for FH: *DEM, ForestMask*
- mandatory auxiliary data for TOMO: *DEM*
- mandatory auxiliary data for TOMO_FH: *DEM*
- optional auxiliary for AGB: *ForestHeigth*
- optional auxiliary data for FD: *AverageCovariance* (first step is used to generate first AverageCovariance if this is not present in input)
- optional auxiliary data for FH: *SystemDecorrelationFunction*
- optional auxiliary data for all the chains: *Geometry* (they are computed if this is not present), *CalibrationScreens* (for the prototype; this should be mandatory in the future for TOMO and TOMO_FH)

Note that the auxiliary FNF (*ForestMask* folder) in case of AGB and FH can be taken from the FD output products of a previous run (see 5.3 and 5.4), or can be a TANDEM-X FNF (see [RD9] and [RD10])

- ***OutputFolder***: : is the full path of the folder where processing will save the products: each run of the processing will generate a new folder here inside, with the processing date time stamp, and so without overwriting any of the already present processing (see also 5.3)
- ***GeographicGridSampling***: is the value in meters of the sampling for the final processed products on geographic map, valid for east and north directions



AverageCovariance
CalibrationScreens
DEM
ForestHeight
ForestMask
Geometry
ReferenceAGB
SystemDecorrelationFunction

Fig.4 `AuxiliaryProductsFolder` content example with all the mandatory and optional folders

5.1.2 Configuration Files

The configuration file is custom for each chain so they will be described separately hereafter (see also [RD1], [RD2] for algorithmic details).

5.1.2.1 Common fields in all the configuration files

Some of the Configuration files fields are the same for all the chains and will be addressed hereafter, valid for all the chains: AGB, FD, FH, TOMO_FH and TOMO.

- **EnableResampling:** Flag, if True (and if needed for the data) up-sampling is done on the used L1 stacks of data to double the samples for meeting Nyquist requirements
- **ComputeGeometry:** Flag, if True the geometry library is called to compute the `Geometry` auxiliaries ECEF grid, kz, off nadir angles, reference height, slant range distances and slope (see [RD1]), otherwise the `Geometry` auxiliaries should be already be present in the `AuxiliaryProductsFolder` path specified in the main input file (5.1.1)
- **ApplyCalibrationScreen:** Flag, if True phase correction and ground steering are applied
- **DEMflattening:** Flag, if True DEM phase is removed from data (it is ignored if `ApplyCalibrationScreen` is True)
- **MultilookHeadingCorrection:** Flag, if True resize of multi-look window is done to account for non-zero heading
- **SaveBreakpoints:** Flag, if True breakpoints are saved to output folder (each chain has specific breakpoints, see [RD1])
- **DeleteTemporaryFiles:** Flag, if True all the processing temporary files are removed from output

5.1.2.2 AGB configuration file

The AGB configuration file is present in

- `InstallationFolder/conf/Configuration_File_AGB.xml` (in case of default suite organization)
- `InstallationFolder/AGB/conf/Configuration_File_AGB.xml` (in case of alternative proto-MAAP suite organization)

Fields of the AGB configuration file are the following:

- **GroundCancellation**:
  - **MultiMaster**: Flag, if True average notch is computed with all possible masters, if False it is computed automatically selecting the best master (in both cases, only if more than two acquisitions are present)
  - **EnhancedForestHeight**: forest height in meters used in the processing to determine the kz value for which the ground cancelled data is generated (in both cases, only if more than two acquisitions are present)
  - **ModelBasedEqualization**: [possible values are “1”, “2” and “3”] backscatter equalization
    - 1: equalization OFF
    - 2: equalization ON
    - 3: if just two acquisitions are present in each of the used stacks (5.1.1) is ON, otherwise is OFF
- **EstimateAGB**:
  - **number_of_tests:** number of tests executed for AGB estimation following a random sampling scheme of estimation and calibration points
  - **fraction_of_roi_per_test:** percentage of estimation points randomly selected for each test
  - **fraction_of_cal_per_test:** percentage of calibration points randomly selected for each test
  - **intermediate_ground_averaging:** ground cancelled averaging window size in meters on ground: it should be less or equal to product_resolution/2 (see below)
  - **product_resolution:** output product resolution in meters on geographic map
  - **distance_sampling_area:** ROIs / sampling areas / estimation points distance in meters on geographic map
  - **parameter_block_size:** size of parameter blocks in meters on geographic map (areas in which model parameters other than AGB are assumed spatially constant)
  - **distance_parameter_block:** distance of parameter blocks in meters on geographic map (areas in which model parameters other than AGB are assumed spatially constant)
  - **min_number_of_rois:** minimum number of estimation points in the selected area for optimization
  - **min_number_of_rois_per_stack:** minimum number of estimation points to be covered by each stack
  - **min_number_of_cal_per_test:** minimum number of calibration points for each test to be valid

- **min_number_of_roi_per_test:** minimum number of estimation points for each test to be valid
- **EstimationValidvaluesLimits:** validity range of the output product in t/ha, invalid values will be set to NAN
- **ModelParameters**
  - **ParameterRanges:** range of values used in non-linear optimization of Biomass model (AGB model scaling in dB, AGB model exponent, AGB cosine exponent and agb estimation values in t/ha)
  - **ChangeAcrossPolarization:** flags to let variation of AGB model scaling, AGB model exponent and AGB cosine exponent across different polarizations; see [RD2] AGB section and [RD4] section on test data set
  - **ChangeAcrossStack:** flags to let variation of AGB model scaling, AGB model exponent and AGB cosine exponent across different stacks; see [RD2] AGB section and [RD4] section on test data set
  - **ChangeAcrossGlobalCycle:** flags to let variation of AGB model scaling, AGB model exponent and AGB cosine exponent across different global cycles; see [RD2] AGB section and [RD4] section on test data set
  - **ChangeAcrossHeading:** flags to let variation of AGB model scaling, AGB model exponent and AGB cosine exponent across different headings; see [RD2] AGB section and [RD4] section on test data set
  - **ChangeAcrossSwath:** flags to let variation of AGB model scaling, AGB model exponent and AGB cosine exponent across different swaths; see [RD2] AGB section and [RD4] section on test data set
  - **ChangeAcrossSubSwath:** flags to let variation of AGB model scaling, AGB model exponent and AGB cosine exponent across different sub-swaths; see [RD2] AGB section and [RD4] section on test data set
  - **ChangeAcrossAzimuth:** flags to let variation of AGB model scaling, AGB model exponent and AGB cosine exponent across different azimuths; see [RD2] AGB section and [RD4] section on test data set

5.1.2.3 FD configuration file

The FD configuration file is present in

- *InstallationFolder*/conf/Configuration_File_FD.xml (in case of default suite organization)
- *InstallationFolder*/FD/conf/Configuration_File_FD.xml (in case of alternative proto-MAAP suite organization)

Fields of the FD configuration file are the following:

- **GroundCancellation**: same description of AGB, see 5.1.2.2
- **ChangeDetection:**
  - **ProductResolution:** covariance estimation window in meters on ground
  - **ConfidenceLevel:** determining the threshold of the test for changes

5.1.2.4 FH configuration file

The FH configuration file is present in

- *InstallationFolder*/conf/Configuration_File_FH.xml (in case of default suite organization)
- *InstallationFolder*/FH/conf/Configuration_File_FH.xml (in case of alternative proto-MAAP suite organization)

Fields of the FH configuration file are the following:

- **EstimateFH**
  - **EstimationValidValuesLimits:** validity range of the output product in meters, invalid values will be set to NAN
  - **SpectralShiftFiltering**: flag to enable / disable filtering
  - **ProductResolution:** determines covariance estimation window in meters on ground
  - **Kz_threshold:** range for kz considered valid for estimation
  - **ModelParameters**
    - **MaximunHeigth:** maximum height to be estimated in meters
    - **NumberOfExtintionValues:** number of values evaluated for model fitting (maximum extinction value determined internally)
    - **NumberOfGroundVolumeRatioValue:** number of values evaluated for model fitting (maximum ground to volume ratio value determined internally)
    - **NumberOfTemporalDecorrelatonValue:** number of values evaluated for model fitting (maximum temporal decorrelation value corresponding to one)
  - **MedianFactor:** filtering the estimates

5.1.2.5 TOMO FH configuration file

The TOMO FH configuration file is present in

- *InstallationFolder*/conf/Configuration_File_TOMO_FH.xml (in case of default suite organization)
- *InstallationFolder*/TOMO_FH/conf/Configuration_File_TOMO_FH.xml (in case of alternative proto-MAAP suite organization)

Fields of the TOMO FH configuration file are the following:

- **ProductResolution:** determines covariance estimation window in meters on ground
- **VerticalRange:** minimum, maximum and sampling in elevation for voxel computation, in meters
- **EstimateFH**
  - **EstimationValidValuesLimits:** validity range of the output product in meters
  - **EnableSuperResolution:** Flag, if True Capon estimator is applied instead of beamforming (if False) for tomography
  - **RegularizationNoiseFactor:** Noise variance for regularization of the inversion for Capon, used only if EnableSuperResolution is True
  - **PowerThreshold:** power decay threshold with respect to the peak value
  - **MedianFactor:** filtering the estimates

5.1.2.6 TOMO configuration file

The TOMO configuration file is present in

- *InstallationFolder*/conf/Configuration_File_TOMO.xml (in case of default suite organization)
- *InstallationFolder*/TOMO/conf/Configuration_File_TOMO.xml (in case of alternative proto-MAAP suite organization)

Fields of the TOMO configuration file are the following:

- **InterpolateStack**
  - **RegularBaselineGrid:** flag, if True re-interpolation on a regular baseline grid is done
- **ProductResolution:** determines covariance estimation window in meters on ground
- **VerticalRange:** minimum, maximum and sampling in elevation for voxel computation in meters

5.2 Launch the processor

To launch the processor, inside the src folders there are examples of python run scripts, in details, the examples scripts are, in case of complete suite organization:

`InstallationFolder/src/run_biomassL2_processor.py`

In case of alternative proto-MAAP suite organization:

```
InstallationFolder/AGB/src/run_biomassL2_processor_AGB.py
InstallationFolder/FD/src/run_biomassL2_processor_FD.py
InstallationFolder/FH/src/run_biomassL2_processor_FH.py
InstallationFolder/TOMO/src/run_biomassL2_processor_TOMO.py
InstallationFolder/TOMO_FH/src/run_biomassL2_processor_TOMO_FH.py
```

It is important to maintain the run scripts as done in the installation process (4.1) without modifying `gdal_path` and `gdal_enviroment_path`, however the user is free to create new scripts with the same structure or modify the present ones and basically make them to point at the desired `Input_File.xml` (5.1.1); it is not recommended to modify any other paths for correct functioning.

Once done that, just execute the script inside your python environment and look at the command window messages: all the messages in command window will be automatically copied in the output log file, see 5.3

5.3 Processing output description

Each run of the processing will generate a new folder inside the ***OutputFolder*** specified in the Input File (see 5.1), with the processing date time stamp, and so without overwriting any of the already present processing, in particular the generated output folder name convention is the following:

```
BIOMASS_L2_YYYYMMDDThhmmss
```

Where

- `YYYY:` year
- `MM:` month
- `DD:` day
- `hh:` hour
- `mm:` minute
- `ss:` second

And this is an example:

```
BIOMASS_L2_20200615T203525
```

Those are the outputs generated by each chain (where “`BIOMASS_L2_YYYYMMDDThhmmss`” is as explained below and “`XXX`” is “AGB”, “FD”, “FH”, “TOMO_FH” or “TOMO”)

- `BIOMASS_L2_YYYYMMDDThhmmss/XXX/ biomassL2.log`

  This is the log of the processing, it can be opened with a text editor (suggested Notepad++)

- `BIOMASS_L2_YYYYMMDDThhmmss/XXX/ConfigurationFile.xml`

  This is a copy of the configuration file used (the one present in the `conf` folder), to keep track of it.

- `BIOMASS_L2_YYYYMMDDThhmmss/XXX/InputFile.xml`

  This is a copy of the BiomassL2 main input file used (the one present in the `conf` folder), to keep track of it.

- `BIOMASS_L2_YYYYMMDDThhmmss/XXX/ Products/ temporary_processing`

  This contains all of the temporary products internal to the processing. The folder is present only if the configuration flag `DeleteTemporaryFiles` is `False` (see 5.1.2)

- `BIOMASS_L2_YYYYMMDDThhmmss/XXX/ Products/ breakpoints`

  This contains all of the breakpoints products, specific of each chain; the breakpoints are described in [RD1] . The folder is present only if the configuration flag `SaveBreakpoints` is `True` (see 5.1.2)

- `BIOMASS_L2_YYYYMMDDThhmmss/XXX/ Products/ global_XXX`

  This contains all of the output products, specific of each chain; the outputs are described in [RD1]

5.4 Concatenating processor flows

This section is contained in [RD1] and is reproduced here tailored to the present manual for the reader's commodity.

The flows that can be enabled from the BiomassL2 main input file (see 5.1.1), which are AGB, FD, FH, TOMO_FH and TOMO are independent: the orchestrator executes each automatically and separately and produces independent outputs for each chain.

This section is needed to explain instead how to manually launch each flow sequentially in order to reproduce "mission like" scenario: in this step-by-step execution the suggested concatenation flow order is FD → FH → AGB; the processor flows are better described in [RD2].
In the following block schemas three different suggested flow executions are presented for different purposes.

- Generation of the products in the interferometric mission phase
  Flow order: FD → FH → AGB.
  In the scheme, the *Forest height estimation* is the interferometric flow ("FH") described in [RD1]).
  Note that each flow should have all the needed inputs ready to run, in particular:
  - when FD flow is ready, in order to execute FH manually, copy the output FD Forest Mask inside `AuxiliaryProductsFolder/ForestMask` sub-folder of the FH
  - when FH flow is ready, in order to execute AGB manually, copy the FH output Height Map inside `AuxiliaryProductsFolder/ForestHeight (*)` sub-folder of the AGB and manually copy the output Forest Mask from FD inside `AuxiliaryProductsFolder/ForestMask` sub-folder of the AGB (see 5.1.1, 5.3 and Fig.4)

  (*) Note that although the interface is part of the prototype processor the computation of the consistency map is not available being still an open research topic (the algorithm is TBD in future projects), so the `AuxiliaryProductsFolder/ForestHeight` is not used in this prototype version

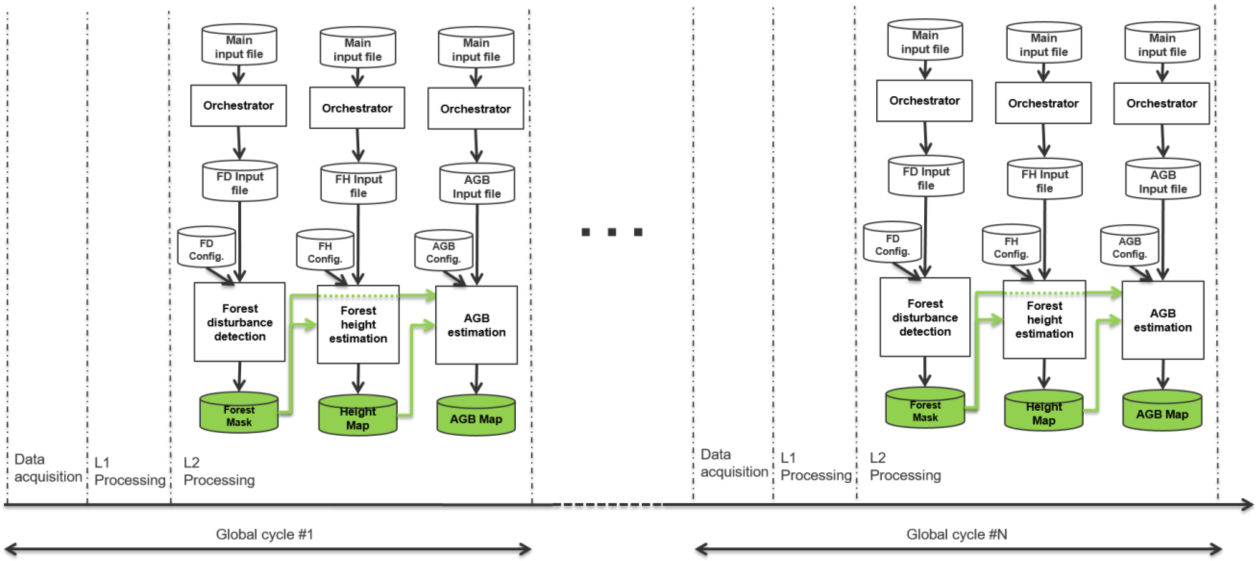


Fig.5 Suggested manual concatenation flow order for the generation of products in the interferometric mission phase

- Generation of the products in the tomographic mission phase.

  Flow order: TOMO_FH → AGB.

  In the scheme, the *Forest height estimation* is the tomographic flow ("TOMO_FH") described in [RD1]).

  When TOMO_FH flow is ready, in order to execute AGB manually, copy the TOMO_FH output Height Map inside `AuxiliaryProductsFolder/ForestHeight` (*) sub-folder of AGB (see 5.1.1, 5.3 and Fig.4)

  (*) Note that although the interface is part of the prototype processor the computation of the consistency map is not available being still an open research topic (the algorithm is TBD in future projects), so the `AuxiliaryProductsFolder/ForestHeight` is not used in this prototype version

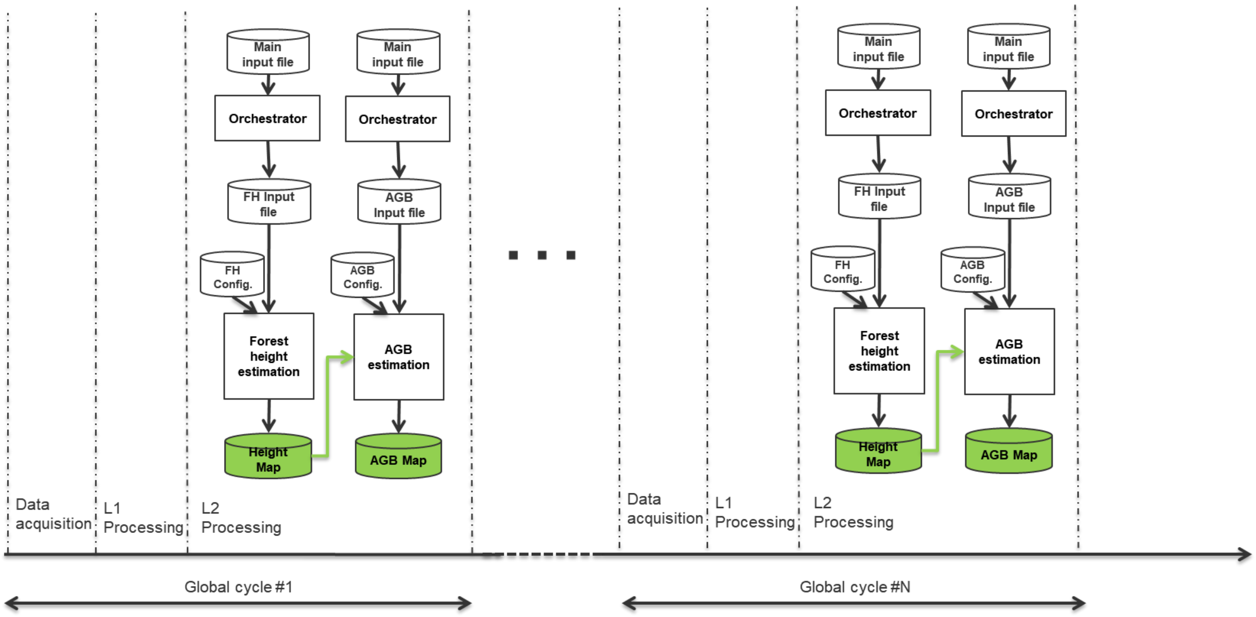


Fig.6 Suggested manual concatenation flow order for the generation of products in the tomographic mission phase

- Generation of tomographic voxels for the tomographic mission phase (TOMO)
  The TOMO chain is self-consistent and does not use any of the output from other chains, neither its outputs are used in any of the other chains

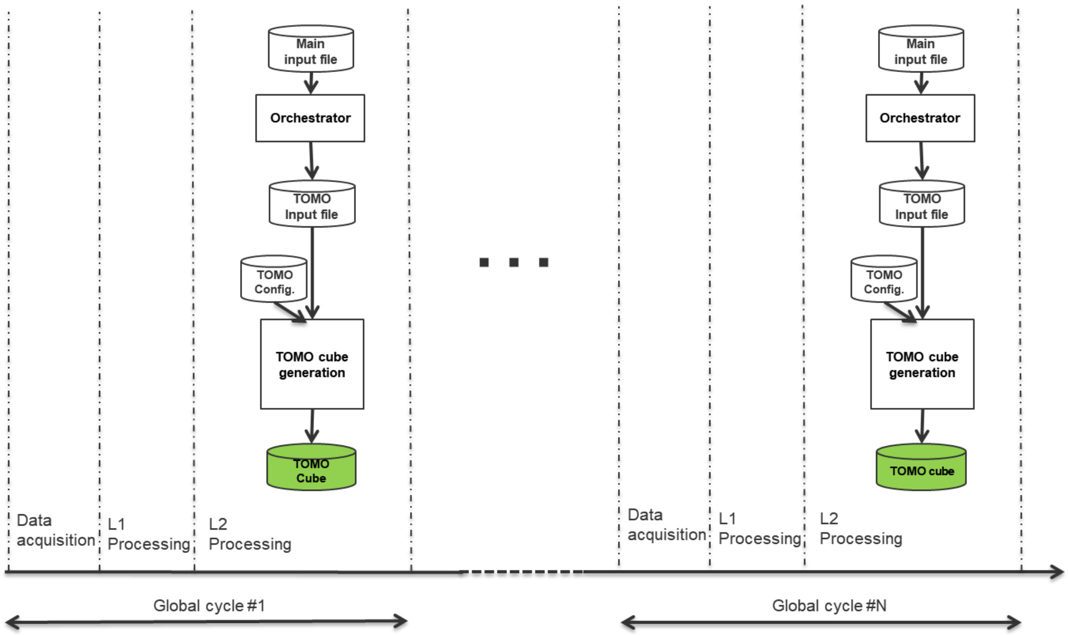


Fig.7 Flow execution for the generation of tomographic voxels (tomographic mission phase)