Procesamiento de series de tiempo en GRASS GIS

Aplicaciones en Ecología y Ambiente

Dra. Verónica Andreo
CONICET - INMeT

Río Cuarto, 2018

Working with Copernicus Sentinel 2 images in GRASS GIS

Overview

Overview

- List available scenes and download

Overview

- List available scenes and download
- Import Sentinel 2 data

Overview

- List available scenes and download
- Import Sentinel 2 data
- Color autobalance

Overview

- List available scenes and download
- Import Sentinel 2 data
- Color autobalance
- Pre-processing

Overview

- List available scenes and download
- Import Sentinel 2 data
- Color autobalance
- Pre-processing
- Cloud and cloud shadow masking

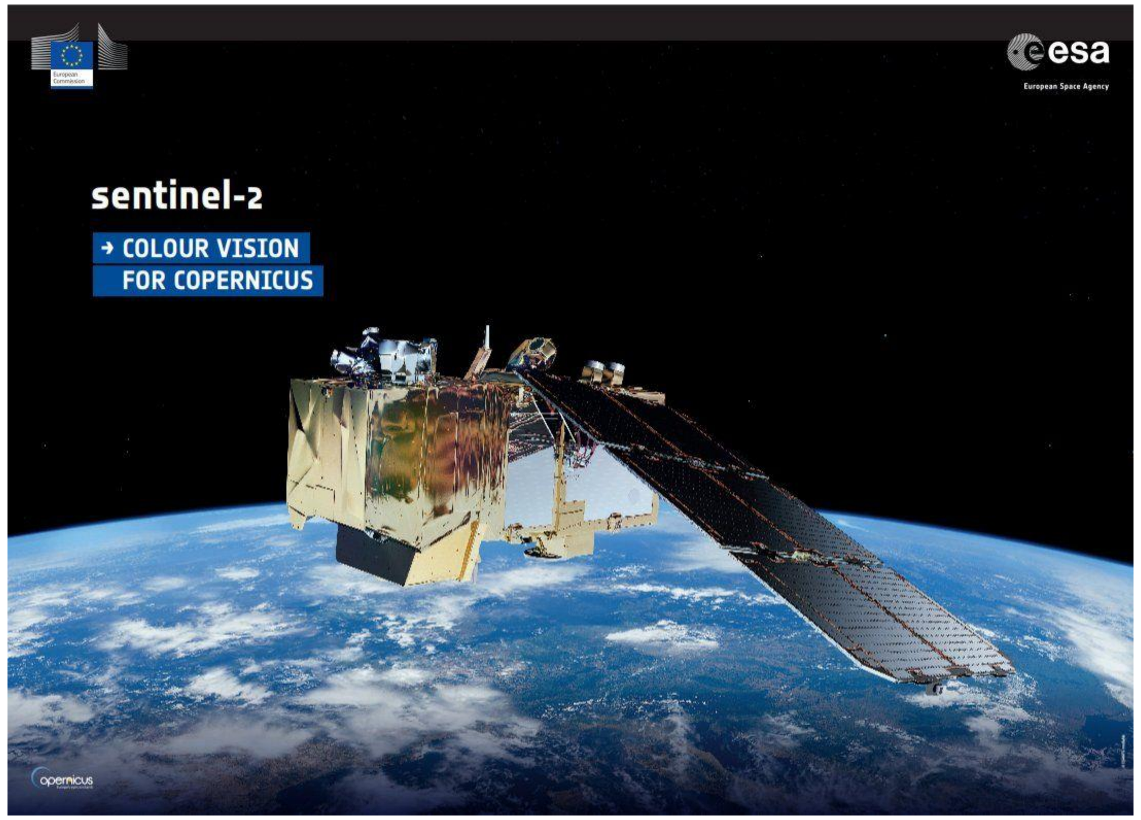
Overview

- List available scenes and download
- Import Sentinel 2 data
- Color autobalance
- Pre-processing
- Cloud and cloud shadow masking
- Vegetation and water indices

Overview

- List available scenes and download
- Import Sentinel 2 data
- Color autobalance
- Pre-processing
- Cloud and cloud shadow masking
- Vegetation and water indices
- Image segmentation

Sentinel 2 data



- Launch: Sentinel-2A in spring 2015, Sentinel-2B in 2017
- Five days revisit time
- Systematic coverage of land and coastal areas between 84°N and 56°S
- 13 spectral bands with spatial resolutions of 10 m (4 VIS and NIR bands), 20 m (6 red-edge/SWIR bands) and 60 m

Set of GRASS GIS extensions to manage Sentinel 2 data:

- i.sentinel.download: downloads Copernicus Sentinel products from Copernicus Open Access Hub
- i.sentinel.import: imports Sentinel satellite data downloaded from Copernicus Open Access Hub
- i.sentinel.preproc: imports and performs atmospheric correction of Sentinel-2 images
- i.sentinel.mask: creates clouds and shadows masks for Sentinel-2 images

See Sentinel wiki for further info

- i.sentinel.download allows downloading Sentinel-2 products from Copernicus Open Access Hub
- To connect to Copernicus Open Access Hub, you need to be registered
- Create the `SETTING_SENTINEL` file with the following content in the `$HOME/gisdata/` directory:

```
myusername
mypassword
```

Download Sentinel 2 data

```
#!/bin/bash
#######################################################################
# Worflow for Sentinel 2 data processing in GRASS GIS
# GRASS GIS postgraduate course in Rio Cuarto
# Author: Veronica Andreo
# October, 2018
#######################################################################


# Create an account in copernicus-hub
https://scihub.copernicus.eu/dhus/#/self-registration

# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
```

Download Sentinel 2 data

```
# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
pip install pandas

# start grass and create a new mapset in NC location
grass74 -c $HOME/grassdata/nc_spm_08_grass7/sentinel2

# install i.sentinel extension
g.extension extension=i.sentinel

# set region to elevation map
g.region -p raster=elevation

# explore list of scenes for a certain date range
```

Start GRASS GIS and create a new mapset

Download Sentinel 2 data

```
# install dependencies (as root user)
pip install sentinelsat
pip install pandas

# start grass and create a new mapset in NC location
grass74 -c $HOME/grassdata/nc_spm_08_grass7/sentinel2

# install i.sentinel extension
g.extension extension=i.sentinel

# set region to elevation map
g.region -p raster=elevation

# explore list of scenes for a certain date range
i.sentinel.download -l settings=$HOME/gisdata/SETTING_SENTINEL \
 start="2018-08-19" end="2018-08-26"
#~ 5 Sentinel product(s) found
```

Install i.sentinel extension

Download Sentinel 2 data

```
pip install pandas

# start grass and create a new mapset in NC location
grass74 -c $HOME/grassdata/nc_spm_08_grass7/sentinel2

# install i.sentinel extension
g.extension extension=i.sentinel

# set region to elevation map
g.region -p raster=elevation

# explore list of scenes for a certain date range
i.sentinel.download -l settings=$HOME/gisdata/SETTING_SENTINEL \
 start="2018-08-19" end="2018-08-26"
#~ 5 Sentinel product(s) found
#~ a559365f-8fc4-4399-8d1c-9123f72cc7a2 2018-08-24T15:48:09Z  1% S2MSI1
#~ 780697f6-0071-4675-b7eb-662d174776b 2018-08-24T15:48:09Z  5% S2MSI1
#~ f188af8c-c7f6-47a6-aca2-4925e2cb2404 2018-08-22T15:59:01Z  6% S2MSI1
```

Set computational region

Download Sentinel 2 data

```
g.extension extension=i.sentinel

# set region to elevation map
g.region -p raster=elevation

# explore list of scenes for a certain date range
i.sentinel.download -l settings=$HOME/gisdata/SETTING_SENTINEL \
 start="2018-08-19" end="2018-08-26"
#~ 5 Sentinel product(s) found
#~ a559365f-8fc4-4399-8d1c-9123f72cc7a2 2018-08-24T15:48:09Z  1% S2MSI1
#~ 780697f6-0071-4675-b7eb-662d1747776b 2018-08-24T15:48:09Z  5% S2MSI1
#~ f188af8c-c7f6-47a6-aca2-4925e2cb2404 2018-08-22T15:59:01Z  6% S2MSI1
#~ c326f43f-5b1f-46e0-8ecc-c37e819425fc 2018-08-22T15:59:01Z  9% S2MSI1
#~ 74f27482-145d-42ea-a628-57a2bd9ca095 2018-08-19T15:49:01Z 16% S2MSI1

# explore list of scenes for a certain date range
i.sentinel.download -l settings=$HOME/gisdata/SETTING_SENTINEL \
 start="2018-08-19" end="2018-08-26" area_relation=Contains
```

List available scenes intersecting computational region

Download Sentinel 2 data

```
#~ 5 Sentinel product(s) found
#~ a559365f-8fc4-4399-8d1c-9123f72cc7a2 2018-08-24T15:48:09Z  1% S2MSI1
#~ 780697f6-0071-4675-b7eb-662d174776b 2018-08-24T15:48:09Z  5% S2MSI1
#~ f188af8c-c7f6-47a6-aca2-4925e2cb2404 2018-08-22T15:59:01Z  6% S2MSI1
#~ c326f43f-5b1f-46e0-8ecc-c37e819425fc 2018-08-22T15:59:01Z  9% S2MSI1
#~ 74f27482-145d-42ea-a628-57a2bd9ca095 2018-08-19T15:49:01Z 16% S2MSI1

# explore list of scenes for a certain date range
i.sentinel.download -l settings=$HOME/gisdata/SETTING_SENTINEL \
 start="2018-08-19" end="2018-08-26" area_relation=Contains
#~ 1 Sentinel product(s) found
#~ c326f43f-5b1f-46e0-8ecc-c37e819425fc 2018-08-22T15:59:01Z  9% S2MSI1

# download the scene that fully contains our region
i.sentinel.download settings=$HOME/gisdata/SETTING_SENTINEL \
 uuid=c326f43f-5b1f-46e0-8ecc-c37e819425fc output=$HOME/gisdata \
 footprints=sentinel_2018_08
```

List available scenes containing computational region

Download Sentinel 2 data

```
# explore list of scenes for a certain date range
i.sentinel.download -l settings=$HOME/gisdata/SETTING_SENTINEL \
 start="2018-08-19" end="2018-08-26" area_relation=Contains
#~ 1 Sentinel product(s) found
#~ c326f43f-5b1f-46e0-8ecc-c37e819425fc 2018-08-22T15:59:01Z  9% S2MSI1

# download the scene that fully contains our region
i.sentinel.download settings=$HOME/gisdata/SETTING_SENTINEL \
 uuid=c326f43f-5b1f-46e0-8ecc-c37e819425fc output=$HOME/gisdata \
 footprints=sentinel_2018_08

# print bands info before importing (1 -proj match, 0- no proj match)
i.sentinel.import -p input=$HOME/gisdata/
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
```

Download selected scene

Since downloading takes a while, we'll skip it.

Download the pre-downloaded Sentinel 2 scene we'll use and move it to `$HOME/gisdata`

Download also the file with the code to follow the exercise.

Import Sentinel 2 data

```
#!/bin/bash
#######################################################################
# Worflow for Sentinel 2 data processing in GRASS GIS
# GRASS GIS postgraduate course in Rio Cuarto
# Author: Veronica Andreo
# October, 2018
#######################################################################


# Create an account in copernicus-hub
https://scihub.copernicus.eu/dhus/#/self-registration

# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
```

Import Sentinel 2 data

```
 footprints=sentinel_2018_08

# print bands info before importing (1 -proj match, 0- no proj match)
i.sentinel.import -p input=$HOME/gisdata/
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
```

Print info about bands before importing

Import Sentinel 2 data

```
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ
#~ /home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQ

# import the downloaded data
# -r flag is used to reproject the data during import
# -c flag allows to import also the cloud mask
i.sentinel.import -rc input=$HOME/gisdata/

# display an RGB combination
d.mon wx0
d.rgb -n red=T17SQV_20180822T155901_B04 \
 green=T17SQV_20180822T155901_B03 \
 blue=T17SQV_20180822T155901_B02
```

Import the data

***Task**: Display an RGB 432 combination of the original data and zoom to computational region. How does it look like?*

Color enhancement

```
#!/bin/bash
########################################################################
# Worflow for Sentinel 2 data processing in GRASS GIS
# GRASS GIS postgraduate course in Rio Cuarto
# Author: Veronica Andreo
# October, 2018
########################################################################


# Create an account in copernicus-hub
https://scihub.copernicus.eu/dhus/#/self-registration

# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
```

Color enhancement

```
i.sentinel.import -rc input=$HOME/gisdata/

# display an RGB combination
d.mon wx0
d.rgb -n red=T17SQV_20180822T155901_B04 \
 green=T17SQV_20180822T155901_B03 \
 blue=T17SQV_20180822T155901_B02

# perform color auto-balancing for RGB bands
i.colors.enhance red=T17SQV_20180822T155901_B04 \
 green=T17SQV_20180822T155901_B03 \
 blue=T17SQV_20180822T155901_B02

# display the enhanced RGB combination
d.mon wx0
d.rgb -n red=T17SQV_20180822T155901_B04 \
 green=T17SQV_20180822T155901_B03 \
 blue=T17SQV_20180822T155901_B02
```

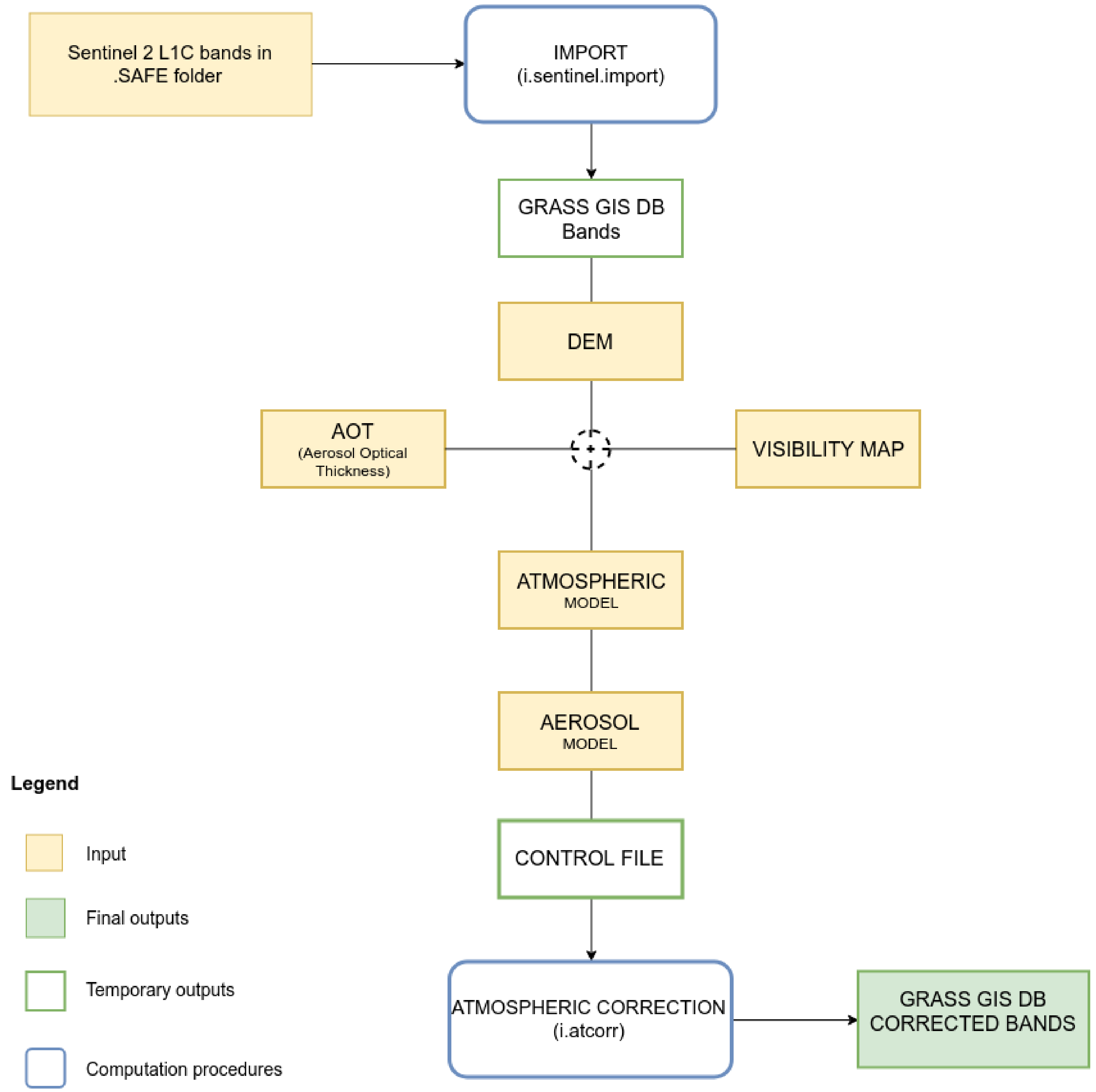
Color enhancement

```
 blue=T17SQV_20180822T155901_B02

# perform color auto-balancing for RGB bands
i.colors.enhance red=T17SQV_20180822T155901_B04 \
 green=T17SQV_20180822T155901_B03 \
 blue=T17SQV_20180822T155901_B02

# display the enhanced RGB combination
d.mon wx0
d.rgb -n red=T17SQV_20180822T155901_B04 \
 green=T17SQV_20180822T155901_B03 \
 blue=T17SQV_20180822T155901_B02



#
# Pre-processing of Sentinel 2 data
#
```

Display an RGB combination of the enhanced bands

Auto-balanced Sentinel scene RGB

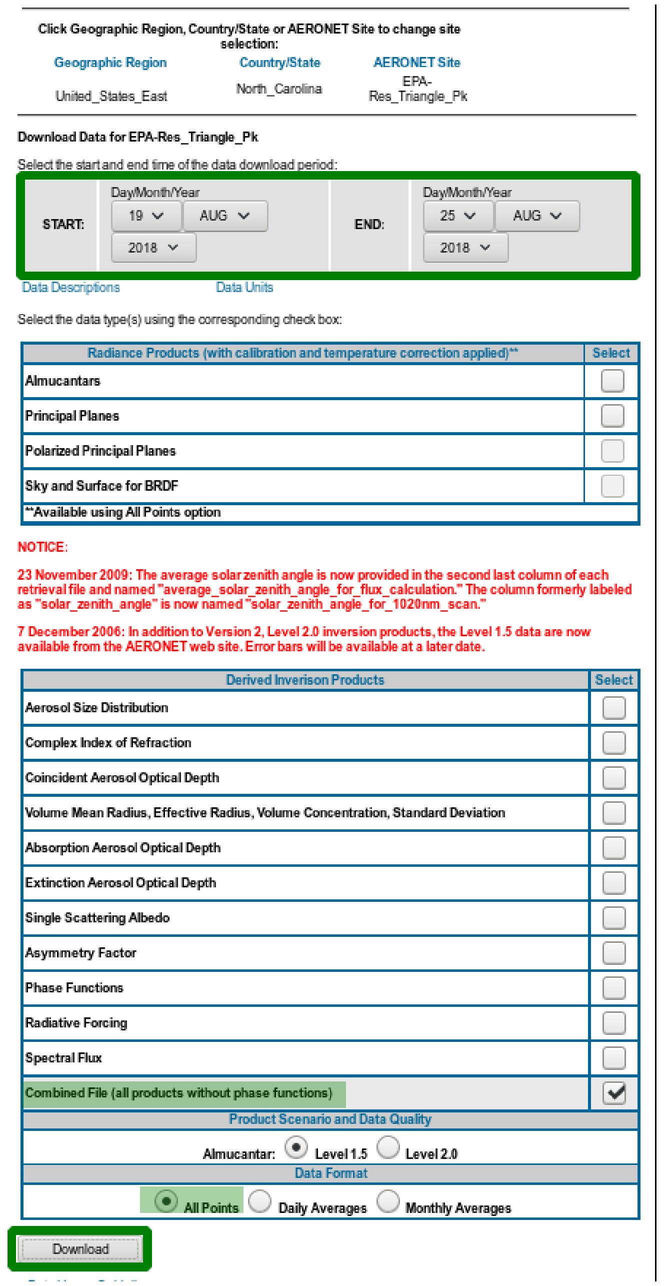
Import with Atmospheric correction: i.sentinel.preproc



We need:

- unzip S2 file
- visibility map or AOD (Aerosol Optic Depth)
- elevation map

Obtain AOD from http://aeronet.gsfc.nasa.gov



- EPA-Res_Triangle_Pk station
- Select start and end date
- Choose Combined file and All points
- Download and unzip in `$HOME/gisdata` (the final file has a .dubovik extension)

If that does not work, here is the AOD file

Elevation map

For now, we'll use the `elevation` map present in NC location

... but only the region covered by `elevation` map will be atmospherically corrected

Import plus atmospheric correction

```
#!/bin/bash
######################################################################
# Worflow for Sentinel 2 data processing in GRASS GIS
# GRASS GIS postgraduate course in Rio Cuarto
# Author: Veronica Andreo
# October, 2018
######################################################################


# Create an account in copernicus-hub
https://scihub.copernicus.eu/dhus/#/self-registration

# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
```

Import plus atmospheric correction

```
 blue=T17SQV_20180822T155901_B02



#
# Pre-processing of Sentinel 2 data
#



# enter directory with Sentinel scene and unzip file
cd $HOME/gisdata/
unzip $HOME/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQV_201808

# get AOD from http://aeronet.gsfc.nasa.gov
# https://aeronet.gsfc.nasa.gov/cgi-bin/webtool_opera_v2_inv?stage=3&re
# select start and end date
# tick the box labelled as 'Combined file (all products without phase f
# choose 'All Points' under Data Format
# download and unzip the file into `$HOME/gisdata/` folder (the final f
```

Enter directory with Sentinel scene and unzip file

Import plus atmospheric correction

```
# https://aeronet.gsfc.nasa.gov/cgi-bin/webtool_opera_v2_inv?stage=3&re
# select start and end date
# tick the box labelled as 'Combined file (all products without phase f
# choose 'All Points' under Data Format
# download and unzip the file into `$HOME/gisdata/` folder (the final f

# run i.sentinel.preproc using elevation map in NC location
i.sentinel.preproc -atr \
 input_dir=$HOME/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQV_2
 elevation=elevation \
 aeronet_file=$HOME/gisdata/180819_180825_EPA-Res_Triangle_Pk.dubovik \
 suffix=corr \
 text_file=$HOME/gisdata/sentinel_mask

# perform color auto-balancing for RGB bands
i.colors.enhance red=T17SQV_20180822T155901_B04_corr \
 green=T17SQV_20180822T155901_B03_corr \
 blue=T17SQV_20180822T155901_B02_corr
```

Run i.sentinel.preproc using elevation map in NC location

Import plus atmospheric correction

```
# run i.sentinel.preproc using elevation map in NC location
i.sentinel.preproc -atr \
 input_dir=$HOME/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQV_2
 elevation=elevation \
 aeronet_file=$HOME/gisdata/180819_180825_EPA-Res_Triangle_Pk.dubovik \
 suffix=corr \
 text_file=$HOME/gisdata/sentinel_mask

# perform color auto-balancing for RGB bands
i.colors.enhance red=T17SQV_20180822T155901_B04_corr \
 green=T17SQV_20180822T155901_B03_corr \
 blue=T17SQV_20180822T155901_B02_corr

# display RBG combination of atmospherically corrected image
d.mon wx0
d.rgb -n red=T17SQV_20180822T155901_B04_corr \
 green=T17SQV_20180822T155901_B03_corr \
 blue=T17SQV_20180822T155901_B02_corr
```

Color enhancement

Import plus atmospheric correction

```
 suffix=corr \
 text_file=$HOME/gisdata/sentinel_mask

# perform color auto-balancing for RGB bands
i.colors.enhance red=T17SQV_20180822T155901_B04_corr \
 green=T17SQV_20180822T155901_B03_corr \
 blue=T17SQV_20180822T155901_B02_corr

# display RBG combination of atmospherically corrected image
d.mon wx0
d.rgb -n red=T17SQV_20180822T155901_B04_corr \
 green=T17SQV_20180822T155901_B03_corr \
 blue=T17SQV_20180822T155901_B02_corr

#
# Replace elevation map by SRTM DEM: r.in.srtm.region addon
#
```

Display atmospherically corrected map

Let's now use a different elevation map: SRTM

- Shuttle Radar Topography Mission (SRTM) is a worldwide Digital Elevation Model with a resolution of 30 or 90 meters.
- r.in.srtm.region downloads and imports SRTM data for the current computational region.

Obtain SRTM digital elevation model

```
#!/bin/bash
#####################################################################
# Worflow for Sentinel 2 data processing in GRASS GIS
# GRASS GIS postgraduate course in Rio Cuarto
# Author: Veronica Andreo
# October, 2018
#####################################################################


# Create an account in copernicus-hub
https://scihub.copernicus.eu/dhus/#/self-registration

# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
```

Obtain SRTM digital elevation model

```
#
# Replace elevation map by SRTM DEM: r.in.srtm.region addon
#



# in the NC location get the bounding box of the full S2 scene
g.region raster=T17SQV_20180822T155901_B04 -b
#~ north latitude:   36:07:27.469949N
#~ south latitude:   35:06:24.823887N
#~ west longitude:   78:48:18.996088W
#~ east longitude:   77:33:33.210097W

# open a new grass session in the latlong location
grass74 -c $HOME/grassdata/latlong/srtm

# In the lat-long location, set the region using the values obtained in
```

Get bounding box of the full S2 scene

Obtain SRTM digital elevation model

```
# in the NC location get the bounding box of the full S2 scene
g.region raster=T17SQV_20180822T155901_B04 -b
#~ north latitude:   36:07:27.469949N
#~ south latitude:   35:06:24.823887N
#~ west longitude:   78:48:18.996088W
#~ east longitude:   77:33:33.210097W

# open a new grass session in the latlong location
grass74 -c $HOME/grassdata/latlong/srtm

# In the lat-long location, set the region using the values obtained in
g.region -p n=36:07:27.469949N s=35:06:24.823887N e=77:33:33.210097W w=

# install r.in.srtm.region extension
g.extension extension=r.in.srtm.region
```

Open a new grass session in a lat-long location

Obtain SRTM digital elevation model

```
g.region raster=T17SQV_20180822T155901_B04 -b
#~ north latitude:   36:07:27.469949N
#~ south latitude:   35:06:24.823887N
#~ west longitude:   78:48:18.996088W
#~ east longitude:   77:33:33.210097W


# open a new grass session in the latlong location
grass74 -c $HOME/grassdata/latlong/srtm

# In the lat-long location, set the region using the values obtained in
g.region -p n=36:07:27.469949N s=35:06:24.823887N e=77:33:33.210097W w=

# install r.in.srtm.region extension
g.extension extension=r.in.srtm.region


# downloading and import SRTM data
r.in.srtm.region output=srtm user=your_NASA_user pass=your_NASA_passwor
#~ Importing 4 SRTM tiles...
```

Set the region using the values obtained in NC location

Obtain SRTM digital elevation model

```
#~ west longitude:    78:48:18.996088W
#~ east longitude:    77:33:33.210097W

# open a new grass session in the latlong location
grass74 -c $HOME/grassdata/latlong/srtm

# In the lat-long location, set the region using the values obtained in
g.region -p n=36:07:27.469949N s=35:06:24.823887N e=77:33:33.210097W w=

# install r.in.srtm.region extension
g.extension extension=r.in.srtm.region

# downloading and import SRTM data
r.in.srtm.region output=srtm user=your_NASA_user pass=your_NASA_passwor
#~ Importing 4 SRTM tiles...
  #~ 25%

# display srtm map and get info
```

Install r.in.srtm.region extension

Obtain SRTM digital elevation model

```
grass74 -c $HOME/grassdata/latlong/srtm

# In the lat-long location, set the region using the values obtained in
g.region -p n=36:07:27.469949N s=35:06:24.823887N e=77:33:33.210097W w=

# install r.in.srtm.region extension
g.extension extension=r.in.srtm.region

# downloading and import SRTM data
r.in.srtm.region output=srtm user=your_NASA_user pass=your_NASA_passwor
#~ Importing 4 SRTM tiles...
  #~ 25%

# display srtm map and get info
d.mon wx0
d.rast srtm
r.info srtm
```

Download and import SRTM data for the region

***Task**: Display the imported SRTM map and get basic info*

Reproject and run i.sentinel.preproc again

```
#!/bin/bash
#######################################################################
# Worflow for Sentinel 2 data processing in GRASS GIS
# GRASS GIS postgraduate course in Rio Cuarto
# Author: Veronica Andreo
# October, 2018
#######################################################################


# Create an account in copernicus-hub
https://scihub.copernicus.eu/dhus/#/self-registration

# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
```

Reproject and run i.sentinel.preproc again

```
r.in.srtm.region output=srtm user=your_NASA_user pass=your_NASA_passwor
#~ Importing 4 SRTM tiles...
  #~ 25%

# display srtm map and get info
d.mon wx0
d.rast srtm
r.info srtm

# change back to NC location and sentinel2 mapset
# reproject the SRTM map
r.proj location=latlong mapset=srtm input=srtm resolution=30

# use `srtm` map in i.sentinel.preproc
i.sentinel.preproc -atr \
 input_dir=$HOME/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQV_2
 elevation=srtm \
 aeronet_file=$HOME/gisdata/180819_180825_EPA-Res_Triangle_Pk.dubovik \
 suffix=corr full \
```

Change back to NC location and reproject the SRTM map

Reproject and run i.sentinel.preproc again

```
d.rast srtm
r.info srtm


# change back to NC location and sentinel2 mapset
# reproject the SRTM map
r.proj location=latlong mapset=srtm input=srtm resolution=30

# use `srtm` map in i.sentinel.preproc
i.sentinel.preproc -atr \
 input_dir=$HOME/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17SQV_2
 elevation=srtm \
 aeronet_file=$HOME/gisdata/180819_180825_EPA-Res_Triangle_Pk.dubovik \
 suffix=corr_full \
 text_file=$HOME/gisdata/sentinel_mask_full


# enhance colors
i.colors.enhance red=T17SQV_20180822T155901_B04_corr_full \
 green=T17SQV_20180822T155901_B03_corr_full \
 blue=T17SQV_20180822T155901_B02_corr_full
```

Use `srtm` map in i.sentinel.preproc

***Task**: Enhance colors and display an RGB combination of the S2 full scene*

Clouds and clouds' shadows masks

```
#!/bin/bash
#######################################################################
# Worflow for Sentinel 2 data processing in GRASS GIS
# GRASS GIS postgraduate course in Rio Cuarto
# Author: Veronica Andreo
# October, 2018
#######################################################################


# Create an account in copernicus-hub
https://scihub.copernicus.eu/dhus/#/self-registration

# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
```

Clouds and clouds' shadows masks

```
 blue=T17SQV_20180822T155901_B02_corr_full

#
# Identify and mask clouds
#

# identify and mask clouds and clouds shadows: i.sentinel.mask
i.sentinel.mask --o input_file=$HOME/gisdata/sentinel_mask_full \
 cloud_mask=T17SQV_20180822T15590_cloud \
 shadow_mask=T17SQV_20180822T15590_shadow \
 mtd=/home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17

# display output
d.mon wx0
d.rgb -n red=T17SQV_20180822T155901_B04_corr_full \
 green=T17SQV_20180822T155901_B03_corr_full \
 blue=T17SQV_20180822T155901_B02_corr_full
```

Identify and mask clouds and clouds shadows

Clouds and clouds' shadows masks

```
# identify and mask clouds and clouds shadows: i.sentinel.mask
i.sentinel.mask --o input_file=$HOME/gisdata/sentinel_mask_full \
 cloud_mask=T17SQV_20180822T15590_cloud \
 shadow_mask=T17SQV_20180822T15590_shadow \
 mtd=/home/veroandreo/gisdata/S2A_MSIL1C_20180822T155901_N0206_R097_T17

# display output
d.mon wx0
d.rgb -n red=T17SQV_20180822T155901_B04_corr_full \
 green=T17SQV_20180822T155901_B03_corr_full \
 blue=T17SQV_20180822T155901_B02_corr_full
d.vect T17SQV_20180822T15590_cloud fill_color=red
d.vect T17SQV_20180822T15590_shadow fill_color=blue



#
# Estimate vegetation and water indices
```

Display output

Clouds and cloud shadows identified by *i.sentinel.mask*

Vegetation and water indices

```
#!/bin/bash
########################################################################
# Worflow for Sentinel 2 data processing in GRASS GIS
# GRASS GIS postgraduate course in Rio Cuarto
# Author: Veronica Andreo
# October, 2018
########################################################################


# Create an account in copernicus-hub
https://scihub.copernicus.eu/dhus/#/self-registration

# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
```

Vegetation and water indices

```
#



# set region
g.region -p raster=elevation align=T17SQV_20180822T155901_B04_corr_full
#~ projection: 99 (Lambert Conformal Conic)
#~ zone:       0
#~ datum:      nad83
#~ ellipsoid:  a=6378137 es=0.006694380022900787
#~ north:      228500
#~ south:      215000
#~ west:       630000
#~ east:       645000
#~ nsres:      10
#~ ewres:      10
#~ rows:       1350
#~ cols:       1500
#~ cells:      2025000
```

Set computational region

Vegetation and water indices

```
#~ west:        630000
#~ east:        645000
#~ nsres:       10
#~ ewres:       10
#~ rows:        1350
#~ cols:        1500
#~ cells:       2025000

# set clouds mask
v.patch input=T17SQV_20180822T15590_cloud,T17SQV_20180822T15590_shadow
 output=cloud_shadow_mask
v.to.rast input=cloud_shadow_mask output=cloud_shadow_mask use=val valu
r.mask -i raster=cloud_shadow_mask

# estimate vegetation indices
i.vi red=T17SQV_20180822T155901_B04_corr_full \
 nir=T17SQV_20180822T155901_B08_corr_full \
 output=T17SQV_20180822T155901_NDVI viname=ndvi
```

Set clouds mask

Vegetation and water indices

```
# set clouds mask
v.patch input=T17SQV_20180822T15590_cloud,T17SQV_20180822T15590_shadow
 output=cloud_shadow_mask
v.to.rast input=cloud_shadow_mask output=cloud_shadow_mask use=val valu
r.mask -i raster=cloud_shadow_mask

# estimate vegetation indices
i.vi red=T17SQV_20180822T155901_B04_corr_full \
 nir=T17SQV_20180822T155901_B08_corr_full \
 output=T17SQV_20180822T155901_NDVI viname=ndvi

i.vi red=T17SQV_20180822T155901_B04_corr_full \
 nir=T17SQV_20180822T155901_B08_corr_full \
 blue=T17SQV_20180822T155901_B02_corr_full \
 output=T17SQV_20180822T155901_EVI viname=evi

# install extension
g.extension extension=i.wi
```

Estimate vegetation indices

Vegetation and water indices

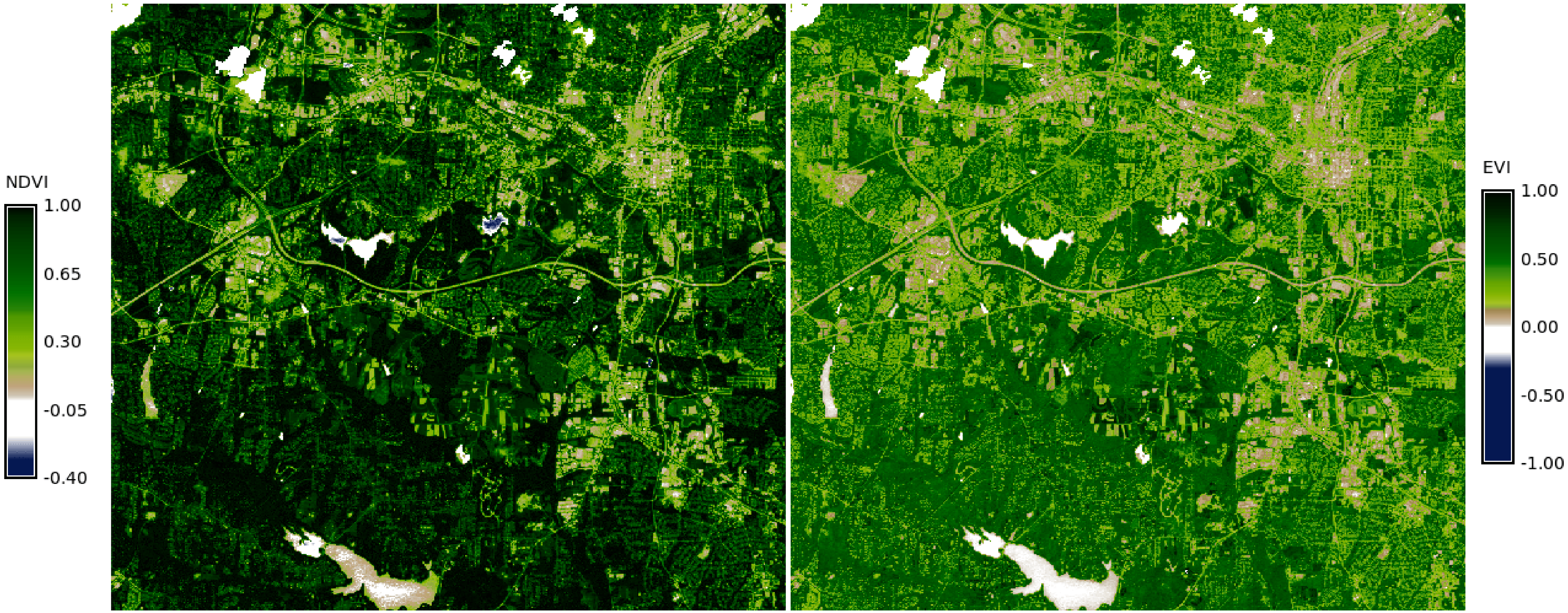
```
# estimate vegetation indices
i.vi red=T17SQV_20180822T155901_B04_corr_full \
 nir=T17SQV_20180822T155901_B08_corr_full \
 output=T17SQV_20180822T155901_NDVI viname=ndvi

i.vi red=T17SQV_20180822T155901_B04_corr_full \
 nir=T17SQV_20180822T155901_B08_corr_full \
 blue=T17SQV_20180822T155901_B02_corr_full \
 output=T17SQV_20180822T155901_EVI viname=evi

# install extension
g.extension extension=i.wi

# estimate water indices
i.wi green=T17SQV_20180822T155901_B03_corr_full \
 nir=T17SQV_20180822T155901_B08_corr_full \
 output=T17SQV_20180822T155901_NDWI winame=ndwi_mf
```

Install i.wi extension

Vegetation and water indices

```
i.vi red=T17SQV_20180822T155901_B04_corr_full \
 nir=T17SQV_20180822T155901_B08_corr_full \
 blue=T17SQV_20180822T155901_B02_corr_full \
 output=T17SQV_20180822T155901_EVI viname=evi

# install extension
g.extension extension=i.wi

# estimate water indices
i.wi green=T17SQV_20180822T155901_B03_corr_full \
 nir=T17SQV_20180822T155901_B08_corr_full \
 output=T17SQV_20180822T155901_NDWI winame=ndwi_mf

#
# Image segmentation
#
```

Estimate water indices



NDVI and EVI from Sentinel 2

NDWI from Sentinel 2

Segmentation

```
#!/bin/bash
#######################################################################
# Worflow for Sentinel 2 data processing in GRASS GIS
# GRASS GIS postgraduate course in Rio Cuarto
# Author: Veronica Andreo
# October, 2018
#######################################################################


# Create an account in copernicus-hub
https://scihub.copernicus.eu/dhus/#/self-registration

# Create a text file in $HOME/gisdata called SENTINEL_SETTING with
your_username
your_password

# install dependencies (as root user)
pip install sentinelsat
```

Segmentation

```
i.wi green=T17SQV_20180822T155901_B03_corr_full \
 nir=T17SQV_20180822T155901_B08_corr_full \
 output=T17SQV_20180822T155901_NDWI winame=ndwi_mf



#
# Image segmentation
#



# install extension
g.extension extension=i.superpixels.slic

# list maps and create groups and subgroups
g.list type=raster pattern="*corr_full" \
 mapset=sentinel2 output=list
i.group group=sentinel subgroup=sentinel file=list

# run i.superpixels.slic and convert the resulting raster to vector
```

Install extension

Segmentation

```
#
# Image segmentation
#



# install extension
g.extension extension=i.superpixels.slic

# list maps and create groups and subgroups
g.list type=raster pattern="*corr_full" \
 mapset=sentinel2 output=list
i.group group=sentinel subgroup=sentinel file=list

# run i.superpixels.slic and convert the resulting raster to vector
i.superpixels.slic input=sentinel \
 output=superpixels num_pixels=2000
r.to.vect input=superpixels output=superpixels type=area
```

List maps and create groups and subgroups

Segmentation

```
# install extension
g.extension extension=i.superpixels.slic

# list maps and create groups and subgroups
g.list type=raster pattern="*corr_full" \
 mapset=sentinel2 output=list
i.group group=sentinel subgroup=sentinel file=list

# run i.superpixels.slic and convert the resulting raster to vector
i.superpixels.slic input=sentinel \
 output=superpixels num_pixels=2000
r.to.vect input=superpixels output=superpixels type=area

# run i.segment and convert the resulting raster to vector
i.segment group=sentinel output=segments \
 threshold=0.5 minsize=100 memory=500
r.to.vect input=segments output=segments type=area
```

Run i.superpixels.slic

Segmentation

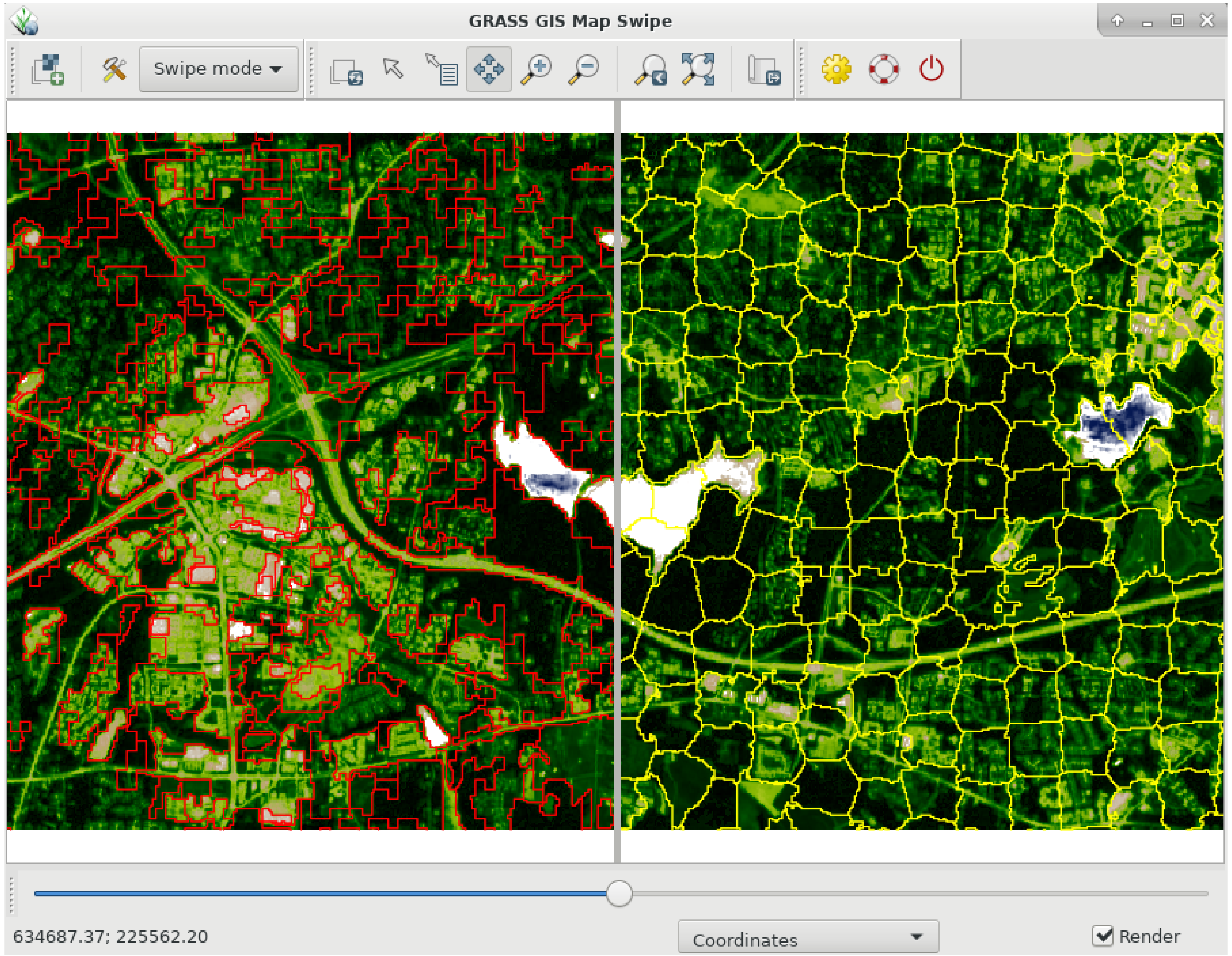
```
g.list type=raster pattern="*corr_full" \
 mapset=sentinel2 output=list
i.group group=sentinel subgroup=sentinel file=list

# run i.superpixels.slic and convert the resulting raster to vector
i.superpixels.slic input=sentinel \
 output=superpixels num_pixels=2000
r.to.vect input=superpixels output=superpixels type=area

# run i.segment and convert the resulting raster to vector
i.segment group=sentinel output=segments \
 threshold=0.5 minsize=100 memory=500
r.to.vect input=segments output=segments type=area

# display NDVI along with the 2 segmentation outputs
d.mon wx0
d.rast map=T17SQV_20180822T155901_NDVI
d.vect map=superpixels fill_color=none
d.vect map=segments fill_color=none
```

Run i.segment

Segmentation

```
g.list type=raster pattern="*corr_full" \
 mapset=sentinel2 output=list
i.group group=sentinel subgroup=sentinel file=list

# run i.superpixels.slic and convert the resulting raster to vector
i.superpixels.slic input=sentinel \
 output=superpixels num_pixels=2000
r.to.vect input=superpixels output=superpixels type=area

# run i.segment and convert the resulting raster to vector
i.segment group=sentinel output=segments \
 threshold=0.5 minsize=100 memory=500
r.to.vect input=segments output=segments type=area

# display NDVI along with the 2 segmentation outputs
d.mon wx0
d.rast map=T17SQV_20180822T155901_NDVI
d.vect map=superpixels fill_color=none
d.vect map=segments fill_color=none
```

Display NDVI along with the 2 segmentation outputs



Segmentation results

QUESTIONS?

Thanks for your attention!!

Move on to:

Temporal data processing and visualization

Presentation powered by

GitPitch