Procesamiento de series de tiempo en GRASS GIS

Aplicaciones en Ecología y Ambiente

Dra. Verónica Andreo
CONICET - INMeT

Río Cuarto, 2018

Hands-on to NDVI time series

Overview

Overview

- Data for the exercise

Overview

- Data for the exercise
- Get familiar with the data

Overview

- Data for the exercise
- Get familiar with the data
- Use reliability band

Overview

- Data for the exercise
- Get familiar with the data
- Use reliability band
- Create NDVI time series

Overview

- Data for the exercise
- Get familiar with the data
- Use reliability band
- Create NDVI time series
- Gap-filling: HANTS

Overview

- Data for the exercise
- Get familiar with the data
- Use reliability band
- Create NDVI time series
- Gap-filling: HANTS
- Aggregation

Overview

- Data for the exercise
- Get familiar with the data
- Use reliability band
- Create NDVI time series
- Gap-filling: HANTS
- Aggregation
- Phenology indices

Overview

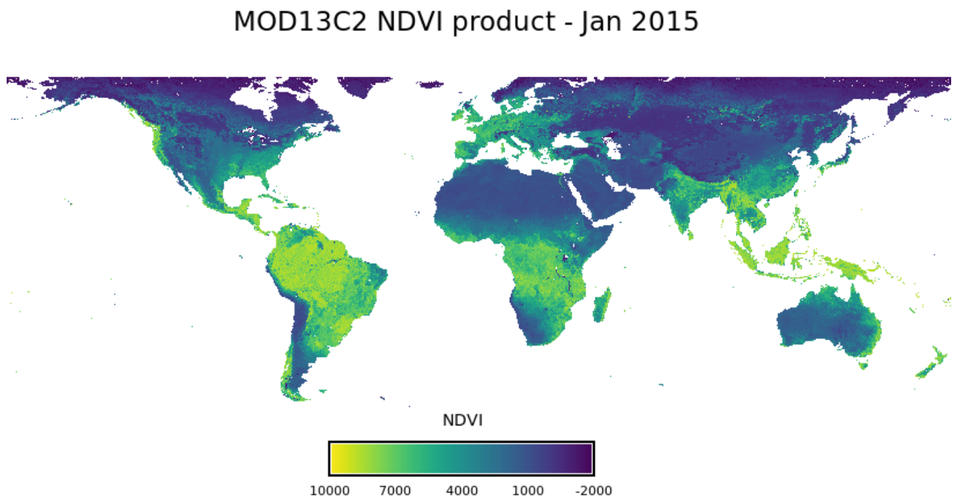
- Data for the exercise
- Get familiar with the data
- Use reliability band
- Create NDVI time series
- Gap-filling: HANTS
- Aggregation
- Phenology indices
- NDWI and flooding frequency

Overview

- Data for the exercise
- Get familiar with the data
- Use reliability band
- Create NDVI time series
- Gap-filling: HANTS
- Aggregation
- Phenology indices
- NDWI and flooding frequency
- Regression between NDVI and NDWI

Data for the exercise

- MODIS product: MOD13C2 Collection 6
- Global monthly composites
- Spatial resolution: 5600m



Data for the exercise

- Download *modis_ndvi* mapset
- Unzip it within North Carolina location
- Ready

Download the code to follow this exercise

Preparation of the dataset

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Preparation of the dataset

```
#
# Data download and preparation
#



### DO NOT RUN ###


# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi

# add modis_lst to path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# set region to an LST map
g.list type=raster mapset=modis_lst
g.region -p raster=MOD11B3.A2015001.h11v05.single_LST_Day_6km@modis_lst
```

Start GRASS GIS in NC location and create a new mapset

Preparation of the dataset

```
#


### DO NOT RUN ###


# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi

# add modis_lst to path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# set region to an LST map
g.list type=raster mapset=modis_lst
g.region -p raster=MOD11B3.A2015001.h11v05.single_LST_Day_6km@modis_lst

# get bounding box in ll
g.region -bg
#~ ll n=40.5924763
```

Add modis_lst mapset to path

Preparation of the dataset

```
# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi

# add modis_lst to path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# set region to an LST map
g.list type=raster mapset=modis_lst
g.region -p raster=MOD11B3.A2015001.h11v05.single_LST_Day_6km@modis_lst

# get bounding box in ll
g.region -bg
#~ ll_n=40.59247652
#~ ll_s=29.48543350
#~ ll_w=-91.37851025
#~ ll_e=-67.97322249
#~ ll_clon=-79.67586637
```

Set region to an LST map

Preparation of the dataset

```
# set region to an LST map
g.list type=raster mapset=modis_lst
g.region -p raster=MOD11B3.A2015001.h11v05.single_LST_Day_6km@modis_lst

# get bounding box in ll
g.region -bg
#~ ll_n=40.59247652
#~ ll_s=29.48543350
#~ ll_w=-91.37851025
#~ ll_e=-67.97322249
#~ ll_clon=-79.67586637
#~ ll_clat=35.03895501

# download MOD13C2 (https://lpdaac.usgs.gov/dataset_discovery/modis/mod
i.modis.download settings=$HOME/gisdata/SETTING \
 product=ndvi_terra_monthly_5600 \
 startday="2015-01-01" \
```

Get bounding box in ll

Preparation of the dataset

```
#~ ll_s=29.48543350
#~ ll_w=-91.37851025
#~ ll_e=-67.97322249
#~ ll_clon=-79.67586637
#~ ll_clat=35.03895501


# download MOD13C2 (https://lpdaac.usgs.gov/dataset_discovery/modis/mod
i.modis.download settings=$HOME/gisdata/SETTING \
 product=ndvi_terra_monthly_5600 \
 startday="2015-01-01" \
 endday="2017-12-31" \
 folder=$HOME/gisdata/mod13

# move to latlong location

# import into latlong location: NDVI, EVI, QA, NIR, SWIR, Pixel reliabi
i.modis.import files=$HOME/gisdata/mod13/listfileMOD13C2.006.txt \
 spectral="( 1 1 1 0 1 0 1 0 0 0 0 0 1 )"
```

Download MOD13C2

Preparation of the dataset

```
i.modis.download settings=$HOME/gisdata/SETTING \
 product=ndvi_terra_monthly_5600 \
 startday="2015-01-01" \
 endday="2017-12-31" \
 folder=$HOME/gisdata/mod13

# move to latlong location

# import into latlong location: NDVI, EVI, QA, NIR, SWIR, Pixel reliabi
i.modis.import files=$HOME/gisdata/mod13/listfileMOD13C2.006.txt \
 spectral="( 1 1 1 0 1 0 1 0 0 0 0 0 1 )"

# set region to bb
g.region -p n=40.59247652 s=29.48543350 w=-91.37851025 e=-67.97322249 \
 align=MOD13C2.A2017335.006.single_CMG_0.05_Deg_Monthly_NDVI

# subset to region and remove global maps
for map in `g.list type=raster pattern="MOD13C2*"` ; do
```

Move into latlong_wgs84 location and import

Preparation of the dataset

```
folder=$HOME/gisdata/mod13

# move to latlong location

# import into latlong location: NDVI, EVI, QA, NIR, SWIR, Pixel reliabi
i.modis.import files=$HOME/gisdata/mod13/listfileMOD13C2.006.txt \
 spectral="( 1 1 1 0 1 0 1 0 0 0 0 0 1 )"

# set region to bb
g.region -p n=40.59247652 s=29.48543350 w=-91.37851025 e=-67.97322249 \
 align=MOD13C2.A2017335.006.single_CMG_0.05_Deg_Monthly_NDVI

# subset to region and remove global maps
for map in `g.list type=raster pattern="MOD13C2*"` ; do
 r.mapcalc expression="$map = $map" --o
done

# get list of maps to reproject
```

Set region to bb obtained from NC

Preparation of the dataset

```
i.modis.import files=$HOME/gisdata/mod13/listfileMOD13C2.006.txt \
 spectral="( 1 1 1 0 1 0 1 0 0 0 0 0 1 )"

# set region to bb
g.region -p n=40.59247652 s=29.48543350 w=-91.37851025 e=-67.97322249 \
 align=MOD13C2.A2017335.006.single_CMG_0.05_Deg_Monthly_NDVI

# subset to region and remove global maps
for map in `g.list type=raster pattern="MOD13C2*"` ; do
 r.mapcalc expression="$map = $map" --o
done

# get list of maps to reproject
g.list type=raster pattern="MOD13C2*" output=list_proj.txt

# move back to NC location

# reproject
```

Subset to region

Preparation of the dataset

```
g.region -p n=40.59247652 s=29.48543350 w=-91.37851025 e=-67.97322249 \
 align=MOD13C2.A2017335.006.single_CMG_0.05_Deg_Monthly_NDVI

# subset to region and remove global maps
for map in `g.list type=raster pattern="MOD13C2*"` ; do
 r.mapcalc expression="$map = $map" --o
done

# get list of maps to reproject
g.list type=raster pattern="MOD13C2*" output=list_proj.txt

# move back to NC location

# reproject
for map in `cat list_proj.txt` ; do
 r.proj input=$map \
  location=latlong_wgs84 \
  mapset=testing \
```

List of maps that will be reprojected

Preparation of the dataset

```
# get list of maps to reproject
g.list type=raster pattern="MOD13C2*" output=list_proj.txt

# move back to NC location


# reproject
for map in `cat list_proj.txt` ; do
 r.proj input=$map \
  location=latlong_wgs84 \
  mapset=testing \
  resolution=5600
done

# check projected data
r.info map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI

### END OF DO NOT RUN ###
```

Reprojection - in target location

Preparation of the dataset

```
# reproject
for map in `cat list_proj.txt` ; do
 r.proj input=$map \
  location=latlong_wgs84 \
  mapset=testing \
  resolution=5600
done

# check projected data
r.info map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI

### END OF DO NOT RUN ###



#
# Get familiar with NDVI data
#
```

Check projected data

Get familiar with NDVI data

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Get familiar with NDVI data

```
#
# Get familiar with NDVI data
#



# download ready to use mapset and unzip in NC location


# start GRASS GIS in `modis_ndvi` mapset
grass74 $HOME/grassdata/nc_spm_08_grass7/modis_ndvi

# add `modis_lst` to accessible mapsets path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# list files and get info and stats
g.list type=raster mapset=.
r.info map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI
```

Start GRASS GIS in modis_ndvi mapset

Get familiar with NDVI data

```
#



# download ready to use mapset and unzip in NC location


# start GRASS GIS in `modis_ndvi` mapset
grass74 $HOME/grassdata/nc_spm_08_grass7/modis_ndvi

# add `modis_lst` to accessible mapsets path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# list files and get info and stats
g.list type=raster mapset=.
r.info map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI
r.univar map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI
```

Add modis_lst to accessible mapsets path

Get familiar with NDVI data

```
# start GRASS GIS in `modis_ndvi` mapset
grass74 $HOME/grassdata/nc_spm_08_grass7/modis_ndvi

# add `modis_lst` to accessible mapsets path
g.mapsets -p
g.mapsets mapset=modis_lst operation=add

# list files and get info and stats
g.list type=raster mapset=.
r.info map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI
r.univar map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI


#
# Use of reliability band
#

# set computational region
```

List files and get info and stats

Task:

- *Display EVI, NIR and QA maps and get information about minimum and maximum values*
- *What do you notice?*

Use of reliability band

***Task*:**

- *Read about this reliability band at the MOD13 User guide (pag 27).*
- *Display one of the pixel reliability bands along with NDVI band of the same date.*
- *Select only pixels with value 0 (Good quality) in the pixel reliability band. What do you notice?*

Use of reliability band

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Use of reliability band

```
r.info map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI
r.univar map=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI



#
# Use of reliability band
#


# set computational region
g.region -p vector=nc_state \
 align=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI

# keep only NDVI most reliable pixels (one map) - *nix
PR=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_pixel_reliability
NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc \
 expression="${NDVI}_filt = if(${PR} != 0, null(), ${NDVI})"
```

Set computational region

Use of reliability band

```
# Use of reliability band
#


# set computational region
g.region -p vector=nc_state \
 align=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI

# keep only NDVI most reliable pixels (one map) - *nix
PR=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_pixel_reliability
NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc \
 expression="${NDVI}_filt = if(${PR} != 0, null(), ${NDVI})"

# keep only NDVI most reliable pixels (one map) - windows
SET PR=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_pixel_reliabili
SET NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc expression="%NDVI%_filt = if(%PR% != 0, null(), %NDVI%)"
```

Keep only best quality pixels - UNIX

Use of reliability band

```
# keep only NDVI most reliable pixels (one map) - *nix
PR=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_pixel_reliability
NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc \
 expression="${NDVI}_filt = if(${PR} != 0, null(), ${NDVI})"

# keep only NDVI most reliable pixels (one map) - windows
SET PR=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_pixel_reliabili
SET NDVI=MOD13C2.A2015274.006.single_CMG_0.05_Deg_Monthly_NDVI

r.mapcalc expression="%NDVI%_filt = if(%PR% != 0, null(), %NDVI%)"

# for all NDVI maps (Windows users run bash.exe and once done, exit)

# list of maps
PR=`g.list type=raster pattern="*_pixel_reliability" separator=" "`
NDVI=`g.list type=raster pattern="*_Monthly_NDVI" separator=" "`
```

Keep only best quality pixels - Windows

Use of reliability band

```
r.mapcalc expression="%NDVI%_filt = if(%PR% != 0, null(), %NDVI%)"

# for all NDVI maps (Windows users run bash.exe and once done, exit)

# list of maps
PR=`g.list type=raster pattern="*_pixel_reliability" separator=" "`
NDVI=`g.list type=raster pattern="*_Monthly_NDVI" separator=" "`
# convert list to array
PR=($PR)
NDVI=($NDVI)

# iterate over the 2 arrays
for ((i=0;i<${#PR[@]};i++)) ; do
 echo ${PR[$i]} ${NDVI[$i]};
 r.mapcalc --o \
  expression="${NDVI[$i]}_filt = if(${PR[$i]} != 0, null(), ${NDVI[$i]}
done
```

Keep only best quality pixels - all maps

***Task**: Compare stats among original and filtered NDVI maps for the same date*

Create time series

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Create time series

```
#
# Create NDVI time series
#



# create STRDS
t.create output=ndvi_monthly \
 type=strds temporaltype=absolute \
 title="Filtered monthly NDVI" \
 description="Filtered monthly NDVI - MOD13C2 - 2015-2017"

# check it was created
t.list type=strds

# list NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt
```

Create the STRDS

Create time series

```
#


# create STRDS
t.create output=ndvi_monthly \
 type=strds temporaltype=absolute \
 title="Filtered monthly NDVI" \
 description="Filtered monthly NDVI - MOD13C2 - 2015-2017"

# check it was created
t.list type=strds

# list NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt

# register maps
t.register -i input=ndvi_monthly \
 type=raster file=ndvi_list.txt \
```

Check STRDS was created

Create time series

```
# create STRDS
t.create output=ndvi_monthly \
 type=strds temporaltype=absolute \
 title="Filtered monthly NDVI" \
 description="Filtered monthly NDVI - MOD13C2 - 2015-2017"

# check it was created
t.list type=strds

# list NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt

# register maps
t.register -i input=ndvi_monthly \
 type=raster file=ndvi_list.txt \
 start="2015-01-01" increment="1 months"

# print time series info
```

Create file with list of maps

Create time series

```
 description="Filtered monthly NDVI - MOD13C2 - 2015-2017"

# check it was created
t.list type=strds

# list NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt

# register maps
t.register -i input=ndvi_monthly \
 type=raster file=ndvi_list.txt \
 start="2015-01-01" increment="1 months"

# print time series info
t.info input=ndvi_monthly

# print list of maps in time series
t.rast.list input=ndvi_monthly
```

Register maps

Create time series

```
# list NDVI filtered files
g.list type=raster pattern="*filt" output=ndvi_list.txt

# register maps
t.register -i input=ndvi_monthly \
 type=raster file=ndvi_list.txt \
 start="2015-01-01" increment="1 months"

# print time series info
t.info input=ndvi_monthly

# print list of maps in time series
t.rast.list input=ndvi_monthly


#
# Estimate percentage of missing data
```

Print time series info

Create time series

```
# register maps
t.register -i input=ndvi_monthly \
 type=raster file=ndvi_list.txt \
 start="2015-01-01" increment="1 months"

# print time series info
t.info input=ndvi_monthly

# print list of maps in time series
t.rast.list input=ndvi_monthly


#
# Estimate percentage of missing data
#
```

Print list of maps in STRDS

***Task**: Visually explore the values of the time series in different points. Use g.gui.tplot and select different points interactively.*

Missing data

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Missing data

```
# print list of maps in time series
t.rast.list input=ndvi_monthly



#
# Estimate percentage of missing data
#



# set mask
r.mask vector=nc_state

# How much missing data we have after filtering for pixel reliability?
t.rast.univar input=ndvi_monthly

# count valid data
t.rast.series input=ndvi_monthly \
 method=count output=ndvi_count_valid
```

Set mask

Missing data

```
#
# Estimate percentage of missing data
#



# set mask
r.mask vector=nc_state

# How much missing data we have after filtering for pixel reliability?
t.rast.univar input=ndvi_monthly

# count valid data
t.rast.series input=ndvi_monthly \
 method=count output=ndvi_count_valid

# estimate percentage of missing data
r.mapcalc \
```

Get time series stats

Missing data

```
#

# set mask
r.mask vector=nc_state

# How much missing data we have after filtering for pixel reliability?
t.rast.univar input=ndvi_monthly

# count valid data
t.rast.series input=ndvi_monthly \
 method=count output=ndvi_count_valid

# estimate percentage of missing data
r.mapcalc \
 expression="ndvi_missing = ((36 - ndvi_count_valid) * 100.0)/36"


#
```

Count valid data

Missing data

```
r.mask vector=nc_state

# How much missing data we have after filtering for pixel reliability?
t.rast.univar input=ndvi_monthly

# count valid data
t.rast.series input=ndvi_monthly \
 method=count output=ndvi_count_valid

# estimate percentage of missing data
r.mapcalc \
 expression="ndvi_missing = ((36 - ndvi_count_valid) * 100.0)/36"


#
# Temporal gap-filling: HANTS
#
```

Estimate percentage of missing data

***Task**:*

- *Display the map representing the percentage of missing data and explore values.*
- *Get univariate statistics of this map.*

Temporal gap-filling: HANTS

- Harmonic Analysis of Time Series (HANTS)
- Implemented in r.hants addon

Temporal gap-filling: HANTS

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Temporal gap-filling: HANTS

```
r.mapcalc \
 expression="ndvi_missing = ((36 - ndvi_count_valid) * 100.0)/36"


#
# Temporal gap-filling: HANTS
#



# install extension
g.extension extension=r.hants

# *nix
# list maps
maplist=`t.rast.list input=ndvi_monthly method=comma`

# gapfill: r.hants
r.hants in=$maplist range=-2000,10000 nf=5 fet=500 base_period=12
```

Install extension

Temporal gap-filling: HANTS

```
#
# Temporal gap-filling: HANTS
#



# install extension
g.extension extension=r.hants

# *nix
# list maps
maplist=`t.rast.list input=ndvi_monthly method=comma`

# gapfill: r.hants
r.hants in=$maplist range=-2000,10000 nf=5 fet=500 base_period=12


# patch original with filled (one map)
NDVI_ORIG=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI_filt
NDVI_HANTS=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI_filt_h
```

List maps

Temporal gap-filling: HANTS

```
# install extension
g.extension extension=r.hants

# *nix
# list maps
maplist=`t.rast.list input=ndvi_monthly method=comma`

# gapfill: r.hants
r.hants in=$maplist range=-2000,10000 nf=5 fet=500 base_period=12

# patch original with filled (one map)
NDVI_ORIG=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI_filt
NDVI_HANTS=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI_filt_h

r.patch input=${NDVI_ORIG},${NDVI_HANTS} \
 output=${NDVI_HANTS}_patch
```

Gap-fill: r.hants

***Task**: Test different parameter settings in r.hants and compare results*

Temporal gap-filling: HANTS

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Temporal gap-filling: HANTS

```
# *nix
# list maps
maplist=`t.rast.list input=ndvi_monthly method=comma`

# gapfill: r.hants
r.hants in=$maplist range=-2000,10000 nf=5 fet=500 base_period=12

# patch original with filled (one map)
NDVI_ORIG=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI_filt
NDVI_HANTS=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI_filt_h

r.patch input=${NDVI_ORIG},${NDVI_HANTS} \
 output=${NDVI_HANTS}_patch

# Windows
# list maps
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly" "method=comma"') D
```

Patch original and gapfilled map

Temporal gap-filling: HANTS

```
r.patch input=${NDVI_ORIG},${NDVI_HANTS} \
 output=${NDVI_HANTS}_patch

# Windows
# list maps
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly" "method=comma"') D

r.hants in=%maplist% range=-2000,10000 nf=5 fet=500 base_period=12

# patch original with filled (one map)
SET NDVI_ORIG=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI_fil
SET NDVI_HANTS=MOD13C2.A2015001.006.single_CMG_0.05_Deg_Monthly_NDVI_fi

r.patch input=%NDVI_ORIG%,%NDVI_HANTS% \
 output=%NDVI_HANTS%_patch
```

Gap-fill and patch maps in Windows

Temporal gap-filling: HANTS

```
r.patch input=%NDVI_ORIG%,%NDVI_HANTS% \
 output=%NDVI_HANTS%_patch

# patch original with filled (all maps, Windows users run bash.exe, onc
# list of maps
ORIG=`g.list type=raster pattern="*_filt" separator=" "`
FILL=`g.list type=raster pattern="*_hants" separator=" "`
# convert list to array
ORIG=($ORIG)
FILL=($FILL)

# iterate over the 2 arrays
for ((i=0;i<${#ORIG[@]};i++)) ; do
 echo ${ORIG[$i]} ${FILL[$i]};
 r.patch input=${ORIG[$i]},${FILL[$i]} output=${FILL[$i]}_patch --o
done
```

Patch original and gapfilled maps

Temporal gap-filling: HANTS

```
FILL=($FILL)

# iterate over the 2 arrays
for ((i=0;i<${#ORIG[@]};i++)) ; do
 echo ${ORIG[$i]} ${FILL[$i]};
 r.patch input=${ORIG[$i]},${FILL[$i]} output=${FILL[$i]}_patch --o
done

# create new time series
t.create output=ndvi_monthly_patch \
 type=strds temporaltype=absolute \
 title="Patched monthly NDVI" \
 description="Filtered, gap-filled and patched monthly NDVI - MOD13C2 -

# list NDVI patched files
g.list type=raster pattern="*patch" output=list_ndvi_patched.txt

# register maps
```

Create time series with patched data

Temporal gap-filling: HANTS

```
# create new time series
t.create output=ndvi_monthly_patch \
 type=strds temporaltype=absolute \
 title="Patched monthly NDVI" \
 description="Filtered, gap-filled and patched monthly NDVI - MOD13C2 -

# list NDVI patched files
g.list type=raster pattern="*patch" output=list_ndvi_patched.txt

# register maps
t.register -i input=ndvi_monthly_patch \
 type=raster file=list_ndvi_patched.txt \
 start="2015-01-01" increment="1 months"

# print time series info
t.info input=ndvi_monthly_patch
```

Register maps in time series

Temporal gap-filling: HANTS

```
 description="Filtered, gap-filled and patched monthly NDVI - MOD13C2"

# list NDVI patched files
g.list type=raster pattern="*patch" output=list_ndvi_patched.txt

# register maps
t.register -i input=ndvi_monthly_patch \
 type=raster file=list_ndvi_patched.txt \
 start="2015-01-01" increment="1 months"

# print time series info
t.info input=ndvi_monthly_patch


#
# Obtain phenological information
#
```

Print time series info

***Task**: Graphically assess the results of HANTS reconstruction in pixels with higher percentage of missing data and obtain univariate statistics for the new time series*

Aggregation with granularity

***Task**:*

- *Obtain average NDVI every two months*
- *Visualize the resulting time series with g.gui.animation*

Phenology

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Phenology

```
# print time series info
t.info input=ndvi_monthly_patch



#
# Obtain phenological information
#



# get month of maximum and month of minimum
t.rast.series input=ndvi_monthly_patch method=minimum output=ndvi_min
t.rast.series input=ndvi_monthly_patch method=maximum output=ndvi_max

# get month of maximum and minimum
t.rast.mapcalc -n inputs=ndvi_monthly_patch output=month_max_ndvi \
  expression="if(ndvi_monthly_patch == ndvi_max, start_month(), null())"
  basename=month_max_ndvi
```

Month of maximum and month of minimum

Phenology

```
#

# get month of maximum and month of minimum
t.rast.series input=ndvi_monthly_patch method=minimum output=ndvi_min
t.rast.series input=ndvi_monthly_patch method=maximum output=ndvi_max

# get month of maximum and minimum
t.rast.mapcalc -n inputs=ndvi_monthly_patch output=month_max_ndvi \
  expression="if(ndvi_monthly_patch == ndvi_max, start_month(), null())"
  basename=month_max_ndvi

t.rast.mapcalc -n inputs=ndvi_monthly_patch output=month_min_ndvi \
  expression="if(ndvi_monthly_patch == ndvi_min, start_month(), null())"
  basename=month_min_ndvi

# get the earliest month in which the maximum and minimum appeared
t.rast.series input=month_max_ndvi method=maximum output=max_ndvi_date
t.rast.series input=month_min_ndvi method=minimum output=min_ndvi_date
```

Replace STRDS values with start_month if they match overall min or max

Phenology

```
# get month of maximum and minimum
t.rast.mapcalc -n inputs=ndvi_monthly_patch output=month_max_ndvi \
  expression="if(ndvi_monthly_patch == ndvi_max, start_month(), null())" \
  basename=month_max_ndvi

t.rast.mapcalc -n inputs=ndvi_monthly_patch output=month_min_ndvi \
  expression="if(ndvi_monthly_patch == ndvi_min, start_month(), null())" \
  basename=month_min_ndvi

# get the earliest month in which the maximum and minimum appeared
t.rast.series input=month_max_ndvi method=maximum output=max_ndvi_date
t.rast.series input=month_min_ndvi method=minimum output=min_ndvi_date

# remove month_max_lst strds
t.remove -rf inputs=month_max_ndvi,month_min_ndvi
#associate max lst with max ndvi, max lst date with max ndvi date


# time series of slopes
```

Get the earliest month in which the maximum and minimum appeared

Phenology

```
  basename=month_max_ndvi

t.rast.mapcalc -n inputs=ndvi_monthly_patch output=month_min_ndvi \
  expression="if(ndvi_monthly_patch == ndvi_min, start_month(), null())" \
  basename=month_min_ndvi

# get the earliest month in which the maximum and minimum appeared
t.rast.series input=month_max_ndvi method=maximum output=max_ndvi_date
t.rast.series input=month_min_ndvi method=minimum output=min_ndvi_date

# remove month_max_lst strds
t.remove -rf inputs=month_max_ndvi,month_min_ndvi
#associate max lst with max ndvi, max lst date with max ndvi date

# time series of slopes
t.rast.algebra \
 expression="slope_ndvi = (ndvi_monthly[1] - ndvi_monthly[0]) / 2.0" \
 basename=slope_ndvi
```

Remove intermediate time series

***Task**: Display the resulting maps with g.gui.mapswipe*

Phenology

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Phenology

```
# get the earliest month in which the maximum and minimum appeared
t.rast.series input=month_max_ndvi method=maximum output=max_ndvi_date
t.rast.series input=month_min_ndvi method=minimum output=min_ndvi_date

# remove month_max_lst strds
t.remove -rf inputs=month_max_ndvi,month_min_ndvi
#associate max lst with max ndvi, max lst date with max ndvi date

# time series of slopes
t.rast.algebra \
 expression="slope_ndvi = (ndvi_monthly[1] - ndvi_monthly[0]) / 2.0" \
 basename=slope_ndvi

# get max slope per year
t.rast.aggregate input=slope_ndvi output=ndvi_slope_yearly \
 basename=NDVI_max_slope_year suffix=gran \
 method=maximum granularity="1 years"
```

Get time series of slopes among consequtive maps

Phenology

```
# remove month_max_lst strds
t.remove -rf inputs=month_max_ndvi,month_min_ndvi
#associate max lst with max ndvi, max lst date with max ndvi date


# time series of slopes
t.rast.algebra \
 expression="slope_ndvi = (ndvi_monthly[1] - ndvi_monthly[0]) / 2.0" \
 basename=slope_ndvi

# get max slope per year
t.rast.aggregate input=slope_ndvi output=ndvi_slope_yearly \
 basename=NDVI_max_slope_year suffix=gran \
 method=maximum granularity="1 years"

# install extension
g.extension extension=r.seasons

# start, end and length of growing season -- *nix
```

Get maximum slope per year

***Task**: Obtain a map with the highest growing rate per pixel in the period 2015-2017 and display it*

Phenology

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Phenology

```
# time series of slopes
t.rast.algebra \
 expression="slope_ndvi = (ndvi_monthly[1] - ndvi_monthly[0]) / 2.0" \
 basename=slope_ndvi

# get max slope per year
t.rast.aggregate input=slope_ndvi output=ndvi_slope_yearly \
 basename=NDVI_max_slope_year suffix=gran \
 method=maximum granularity="1 years"

# install extension
g.extension extension=r.seasons

# start, end and length of growing season - *nix
r.seasons input=`t.rast.list -u input=ndvi_monthly_patch method=comma`
 prefix=ndvi_season n=3 \
 nout=ndvi_season threshold_value=3000 min_length=5
```

Install extension

Phenology

```
# get max slope per year
t.rast.aggregate input=slope_ndvi output=ndvi_slope_yearly \
 basename=NDVI_max_slope_year suffix=gran \
 method=maximum granularity="1 years"

# install extension
g.extension extension=r.seasons

# start, end and length of growing season - *nix
r.seasons input=`t.rast.list -u input=ndvi_monthly_patch method=comma`
 prefix=ndvi_season n=3 \
 nout=ndvi_season threshold_value=3000 min_length=5

# start, end and length of growing season - Windows
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly_patch" "separator=,

r.seasons input=%ndvi_list% prefix=ndvi_season n=3 nout=ndvi_season thr
```

Determine start, end and length of growing season

Phenology

```
# install extension
g.extension extension=r.seasons

# start, end and length of growing season - *nix
r.seasons input=`t.rast.list -u input=ndvi_monthly_patch method=comma`
 prefix=ndvi_season n=3 \
 nout=ndvi_season threshold_value=3000 min_length=5

# start, end and length of growing season - Windows
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly_patch" "separator=,

r.seasons input=%ndvi_list% prefix=ndvi_season n=3 nout=ndvi_season thr

# use threshold map: min ndvi + 0.1*ndvi
r.mapcalc expression="threshold_ndvi = ndvi_min*1.1"
```

Determine start, end and length of growing season - Windows

***Task**: Plot the resulting maps*

Phenology

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Phenology

```
# start, end and length of growing season - *nix
r.seasons input=`t.rast.list -u input=ndvi_monthly_patch method=comma`
 prefix=ndvi_season n=3 \
 nout=ndvi_season threshold_value=3000 min_length=5

# start, end and length of growing season - Windows
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly_patch" "separator=,

r.seasons input=%ndvi_list% prefix=ndvi_season n=3 nout=ndvi_season thr

# use threshold map: min ndvi + 0.1*ndvi
r.mapcalc expression="threshold_ndvi = ndvi_min*1.1"


#
# Estimate NDWI
#
```

Create a threshold map to use in r.seasons

***Task**: Use threshold map in r.seasons and compare output maps with the outputs of using a threshold value only*

Water index time series

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Water index time series

```
#
# Estimate NDWI
#



# create time series of NIR and MIR
t.create output=NIR \
 type=strds temporaltype=absolute \
 title="NIR monthly" \
 description="NIR monthly - MOD13C2 - 2015-2017"

t.create output=MIR \
 type=strds temporaltype=absolute \
 title="MIR monthly" \
 description="MIR monthly - MOD13C2 - 2015-2017"

# list NIR and MIR files
```

Create time series of NIR and MIR

Water index time series

```
t.create output=NIR \
 type=strds temporaltype=absolute \
 title="NIR monthly" \
 description="NIR monthly - MOD13C2 - 2015-2017"

t.create output=MIR \
 type=strds temporaltype=absolute \
 title="MIR monthly" \
 description="MIR monthly - MOD13C2 - 2015-2017"

# list NIR and MIR files
g.list type=raster pattern="*NIR*" output=list_nir.txt
g.list type=raster pattern="*MIR*" output=list_mir.txt

# register maps
t.register -i input=NIR \
 type=raster file=list_nir.txt \
 start="2015-01-01" increment="1 months"
```

List NIR and MIR files

Water index time series

```
 title="MIR monthly" \
 description="MIR monthly - MOD13C2 - 2015-2017"

# list NIR and MIR files
g.list type=raster pattern="*NIR*" output=list_nir.txt
g.list type=raster pattern="*MIR*" output=list_mir.txt

# register maps
t.register -i input=NIR \
 type=raster file=list_nir.txt \
 start="2015-01-01" increment="1 months"

t.register -i input=MIR \
 type=raster file=list_mir.txt \
 start="2015-01-01" increment="1 months"

# print time series info
t.info input=NIR
```

Register maps

Water index time series

```
# register maps
t.register -i input=NIR \
 type=raster file=list_nir.txt \
 start="2015-01-01" increment="1 months"

t.register -i input=MIR \
 type=raster file=list_mir.txt \
 start="2015-01-01" increment="1 months"

# print time series info
t.info input=NIR
t.info input=MIR

# estimate NDWI time series
t.rast.algebra basename=ndwi_monthly \
 expression="ndwi_monthly = if(NIR > 0 && MIR > 0, (float(NIR - MIR) /
```

Print time series info

Water index time series

```
 start="2015-01-01" increment="1 months"

t.register -i input=MIR \
 type=raster file=list_mir.txt \
 start="2015-01-01" increment="1 months"

# print time series info
t.info input=NIR
t.info input=MIR

# estimate NDWI time series
t.rast.algebra basename=ndwi_monthly \
 expression="ndwi_monthly = if(NIR > 0 && MIR > 0, (float(NIR - MIR) /


#
# Frequency of inundation
#
```

Estimate NDWI time series

***Task**: Get maximum and minimum values for each NDWI map and explore the time series plot in different points interactively*

Frequency of flooding

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Frequency of flooding

```
# estimate NDWI time series
t.rast.algebra basename=ndwi_monthly \
 expression="ndwi_monthly = if(NIR > 0 && MIR > 0, (float(NIR - MIR) /



#
# Frequency of inundation
#



# reclassify
t.rast.mapcalc -n input=ndwi_monthly output=flood \
 basename=flood expression="if(ndwi_monthly > 0.8, 1, null())"

# flooding frequency
t.rast.series input=flood output=flood_freq method=sum
```

Reclassify maps according to threshold

Frequency of flooding

```
#
# Frequency of inundation
#



# reclassify
t.rast.mapcalc -n input=ndwi_monthly output=flood \
 basename=flood expression="if(ndwi_monthly > 0.8, 1, null())"

# flooding frequency
t.rast.series input=flood output=flood_freq method=sum


#
# Regression between NDWI and NDVI
#
```

Get flooding frequency

***Task**: Which are the areas that have been flooded most frequently?*

Regression analysis

```
#!/bin/bash
############################################################################
# Commands for NDVI time series exercise
# Author: Veronica Andreo
# Date: October, 2018
############################################################################



#
# Data download and preparation
#


### DO NOT RUN ###

# start GRASS GIS in NC location and create a new mapset
grass74 -c $HOME/grassdata/nc_spm_08_grass7/modis_ndvi
```

Regression analysis

```
basename "flood expression="if(ndwi_monthly > 0.0, 1, null())"

# flooding frequency
t.rast.series input=flood output=flood_freq method=sum


#
# Regression between NDWI and NDVI
#



# install extension
g.extension extension=r.regression.series

# use in *nix
xseries=`t.rast.list input=ndvi_monthly_patch method=comma`
yseries=`t.rast.list input=ndwi_monthly method=comma`

r.regression.series xseries=$xseries yseries=$yseries \
```

Install extension

Regression analysis

```
#
# Regression between NDWI and NDVI
#



# install extension
g.extension extension=r.regression.series

# use in *nix
xseries=`t.rast.list input=ndvi_monthly_patch method=comma`
yseries=`t.rast.list input=ndwi_monthly method=comma`

r.regression.series xseries=$xseries yseries=$yseries \
 output=ndvi_ndwi_rsq method=rsq

# use in Windows
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly_patch" "method=comm
FOR /F %c IN ('t.rast.list "-u" "input=ndwi_monthly" "method=comma"') D
```

Perform regression between NDVI and NDWI time series

Regression analysis

```
#


# install extension
g.extension extension=r.regression.series

# use in *nix
xseries=`t.rast.list input=ndvi_monthly_patch method=comma`
yseries=`t.rast.list input=ndwi_monthly method=comma`

r.regression.series xseries=$xseries yseries=$yseries \
 output=ndvi_ndwi_rsq method=rsq

# use in Windows
FOR /F %c IN ('t.rast.list "-u" "input=ndvi_monthly_patch" "method=comm
FOR /F %c IN ('t.rast.list "-u" "input=ndwi_monthly" "method=comma"') D

r.regression.series xseries=%xseries% yseries=%yseries% \
```

Perform regression between NDVI and NDWI time series - Windows

***Task**: Where is the highest correlation among NDVI and NDWI?*

QUESTIONS?

Thanks for your attention!!

Move on to:
GRASS and R interface

Presentation powered by

GitPitch