GCP/GLO/440/FRA - Component 1

Cartographic material on the evolution of forest cover, land uses and climatic conditions of the pilot sites.

Show case of the Maarmora site, Morocco

October 2013

VITO NV
Boeretang 200 – 2400 MOL – BELGIUM
Tel. + 32 14 33 55 11 – Fax + 32 14 33 55 99
vito@vito.be – www.vito.be

VAT BE-0244.195.916 RPR (Turnhout)
Bank 435-4508191-02 KBC (Brussel)
BE32 4354 5081 9102 (IBAN) KREDBEBB (BIC)

Signatures

Author(s) : Pieter Kempeneers, Antoine Royer

Issuing authority : VITO

Change Record

Release	Date	Description of Change	Editor(s)/Reviewers(s)
V1.0	28 October 2013		

Contents

1 Executive summary

This report summarises the the main results of a study performed by VITO, subcontracted by the food and agriculture organisation (FAO) of the united nations. The study is part of the project GCP/GLO/440/FRA, entitled "optimising the production of goods and services from Mediterranean forest ecosystems within a context of global change". The objective of the study was to intensify and improve these goods and services in a sustainable way. The work has been split in two parts: climatic conditions and forest cover mapping. Whereas the first also uses data collected by ground stations in the field, the second is mainly based on remote sensing data. Nevertheless, ground reference data are an indispensable source of of information for calibrating (training) the methods and validating the results.

To support the vulnerability study FAO needed to investigate what are the climatic and vegetation information available, for which period and how to display these various sources of information to facilitate further analysis. For this component three conditions have driven the investigation:

- Data should be freely available,
- The preliminary data preparation should be reproduced by the national experts without software investment,
- The show case of Maarmonara could be extended to the other Mediterranean forest sites.

The main activities of this study were:

- Review of available data for the production of climatic conditions (temperature and rain fall)
- Production of forest land cover maps for the periods 1990, 2000 and 2010. Following a special request from FAO, the reference year 2005 was also added.
- Production of additional maps that could be useful for the analysis of the vulnerability of the Mamora forest site due to climate change (e.g., change maps of the normalised difference vegetation index)

On top of these activities, VITO participated to a workshop in Solsona, where preliminary results were presented. Prior to that, VITO visited the Haut Commissariat aux Eaux et Forêts et à la Lutte Contre la Désertification in Rabat (Morocco) as preparatory mission to the workshop. This one week mission included a field trip to the Mamora forest, the site under study for this project. VITO received full support of the Moroccan authorities and was provided with the latest field data information available.

Although the data and results presented here are focused on the Mamora forest, the followed approach can be extrapolated to other sites. This was also

confirmed by the positive comments expressed by the delegates of other countries such as Lebanon, Tunisia, Turkey who are also involved in the GCP/-GLO/440/FRA project. However, the success of such an extrapolation will heavily depend on their respective support and the quality of the available ground reference data. In the case of the Mamora forest, detailed information was available on the current forest status (types). Unfortunately, no historical data for the reference years 1990, 2000 and 2005 were available, which hampered the assessment of the forest dynamics in the site. It can be foreseen that for some other regions, field reference data is of poor quality or entirely missing. Although some alternative approaches are given in this report, this will have a serious impact on the quality of the final products.

This study has been performed with sustainability and capacity building in mind. All tools used in this project are open source and available for free. The goal is that countries will be capable to monitor their own forests and manage in a sustainable way using the same techniques that are presented here. Nevertheless, training and capacity building are essential in this respect.

2 Introducing the Mamora forest site

The Mamora site has been divided in administrative units. There are five cantons, labeled from A to E (see Fig. 1). The cantons are sub-divided in groups, labeled with roman numbers (I, II, III,...), counting from North to South and East to West. Each group is again divided in parcels, labeled with Arabic numbers (not shown in Fig. 1). Each parcel was finally divided in forest stands. The reference data from the Moroccan authorities of the national forest inventory (NFI) service provided information at forest stand level. The local projection to which the images have been projected is Merchich, valid for North of Morocco (epsg:26191).

3 Climatic cata

One of the thematic to analyse within the project concerns the evolution of the climatic conditions (rainfall and temperature) over the Mamora forest. A preliminary research of the available sources of information was made and various free data set were identified (see Table 3)

To identify possible climatic variation the longest available time series set was finally selected. This data set is based on the interpolated grid fom meteo stations and produced by JRC/MARS with a spatial resolution of 25 km. Over Mamora forest the data set covers all the other sites of the project as shown in the following tables and maps

The specific reference grid 34048/34049/34050/33048/33049/33050 covering the Mamora forest were identified (see Fig. 3) and various available climatic indicators were extracted.

More than 20 indicators were calculated for each of the six 25km grids :

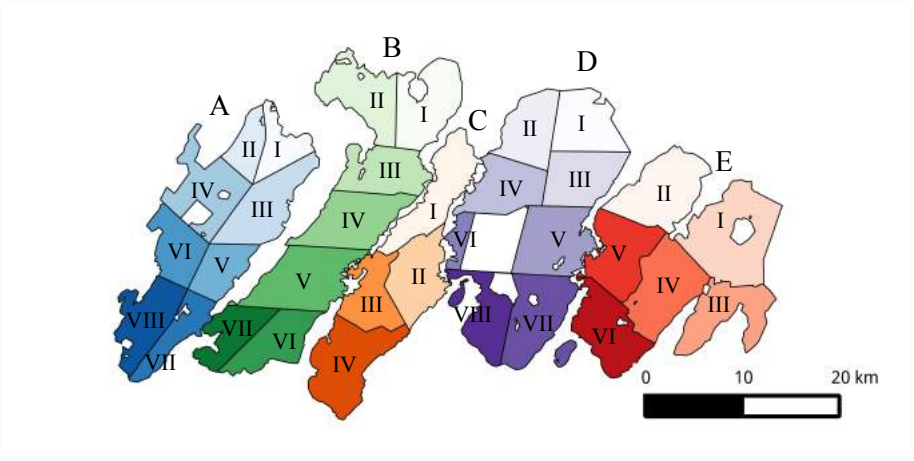


Figure 1: Cantons and groups of Mamora site

Indicators	Source	Time series	Time step	Extension	Resolution
Rainfall estimate RFE v2.0 from MSG	NOAA CPC	From 1995	10 daily	Africa	8 km
`http://earlywarning.usgs.gov/fews/africa/index.php`					
Rainfall estimate TARCAT v2.0 from MSG	READING TAMSAT	From 1983	10 daily	Africa	0.0375°
`http://www.met.reading.ac.uk/~tamsat/about/`					
Rainfall/Temperature from meteo station	JRC MARS	From 1975	daily and 10 daily	Europe and North Africa	25 km
`Site:http://mars.jrc.ec.europa.eu/mars/About-us/AGRI4CAST/Data-distribution/AGRI4CAST-Interpolated-Meteorological-Data`					
Rainfall/Temperature from meteo models	ECMWF JRC	From 1989	10 daily	Global	0.25°
`Site:http://marswiki.jrc.ec.europa.eu/datadownload/index.php`					

Table 1: sources of climatic data

Algeria	Lebanon	Morocco	Tunesia	Turkey
Chrea	Jabal Moussa	Ifrane	Barbara	Ciglikara
Djelfa/Senalba	Kamouaa	Mamora	Siliana	
		Taznakht		

Table 2: JRC / MARS climatic data coverage

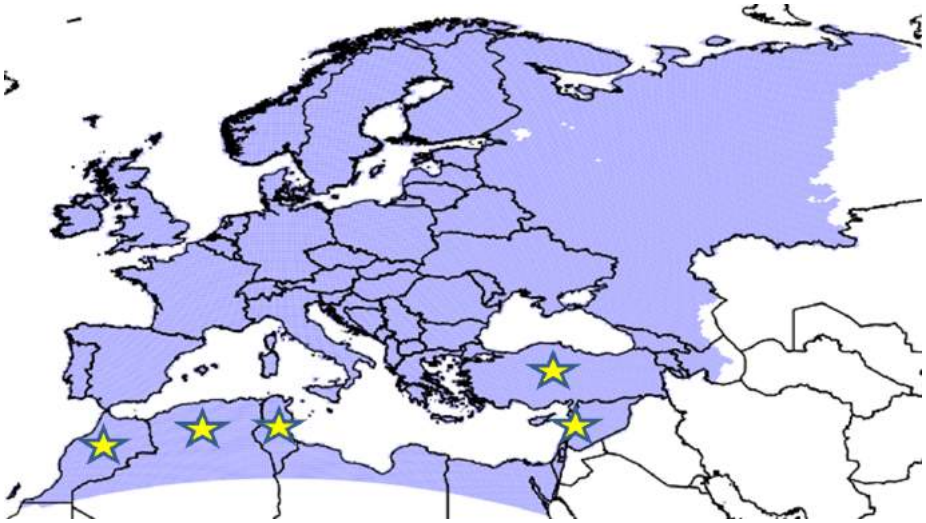


Figure 2: JRC/MARS climatic data coverage (in blue) with the FFEM sites (yellow stars)

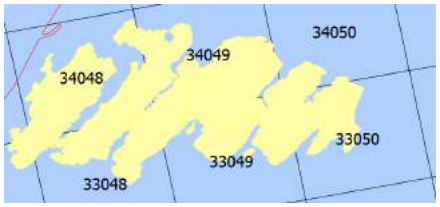


Figure 3: Grid reference of the Mamora forest site (yellow) in Morocco

- Temperatures (min, max, average)
- Number of days above various thresholds(25/30/35/40C) to identify sequences of extreme temperature levels
- Longest heat stress consecutive periods and how many heat stress events of at least 3 consecutive days
- Yearly precipitations and to identify any potential drought spell or extreme precipitation period the number of days for different rain thresholds (> 5 mm and > 30 mm)
- Radiation

The overall idea is to possibly identify changes in the yearly magnitude of the indicators and in theirs frequency/distribution over time.

A database containing the original daily indicators were consolidated in "Access" software. The graphs were mainly generated with the online MARS viewer and can be reproduced for any FFME site.

In Fig. 4-6, the different graphs are shown for one of the six grids covering the Mamora site.

To facilitate the analysis the different indicators were grouped in 3 thematic dashboards :

RAINFALL panel (see Annex A)

- Yearly rainfall
- Standard Precipitation Index
- Number of days > 5 mm
- Number of days > 30 mm

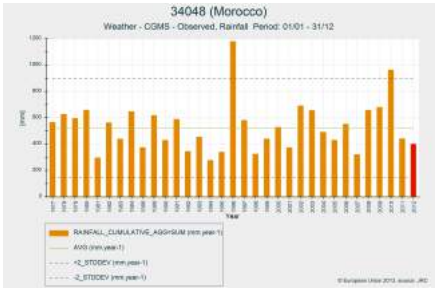
From 1975 only 2006 and 2010 appear as a significant outliers with a standard precipitation index over +0.8 to +1.3. However there is no specific trend for the yearly cumul. In terms of occurence of heavy daily rainfall (> 30 mm) the main significant year is 1996 that recorded more than 12 days of important daily cumul. A part from this specific event no obvious trend can be seen.

HEAT WAVE panel (see Annex B)

- Number of days > 35°
- Number of days > 40°
- Number of heat waves
- Longest heat wave period

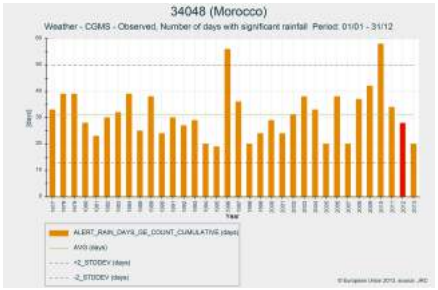
TEMPERATURE panel (see Annex C)

- Maximum temperature (daily average)

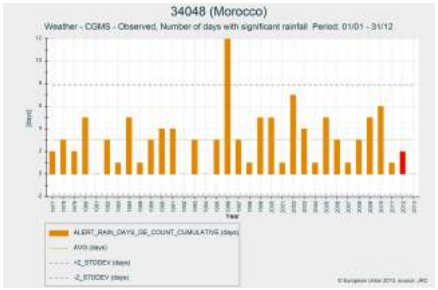


(a) yearly precipitation

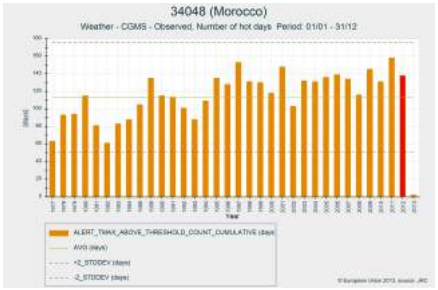
(b) Standard Precipitation Index for the yearly rainfall cumul



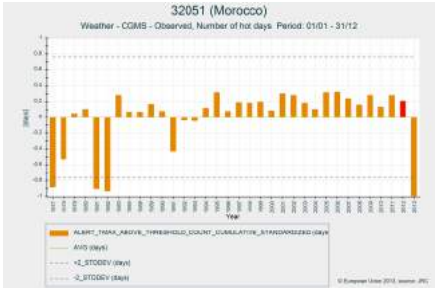
(c) Number of days with significant rainfall > 5 mm



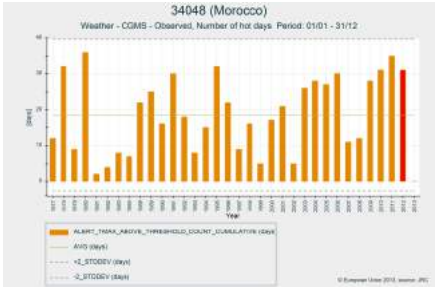
(d) Number of days with significant rainfall > 30 mm



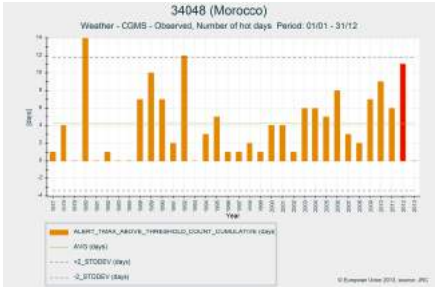
(e) Number of days above 25° C



(f) Standardised number of days above 25° C

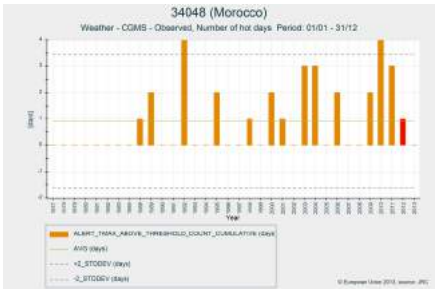


(g) Number of days above 30° C

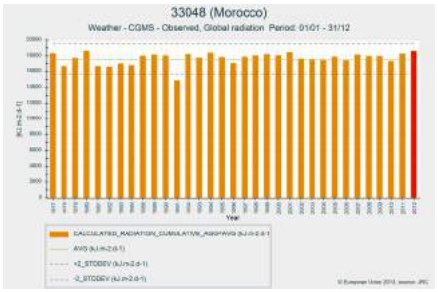


(h) Number of days above 35° C

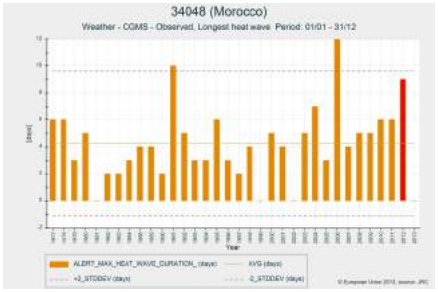
Figure 4: different graphs generated for one of the six grids covering the Mamora site (part 1)



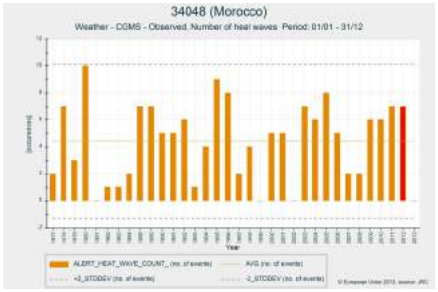
(a) Number of days above 40° C



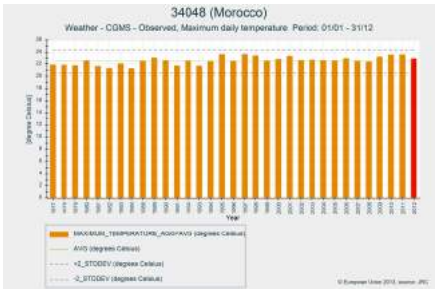
(b) Cumul radiation



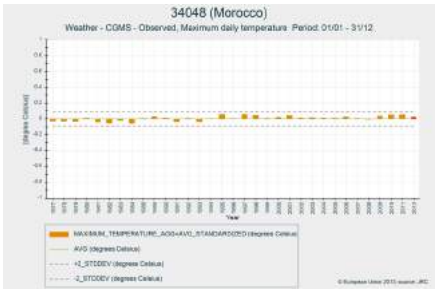
(c) Longest heat wave



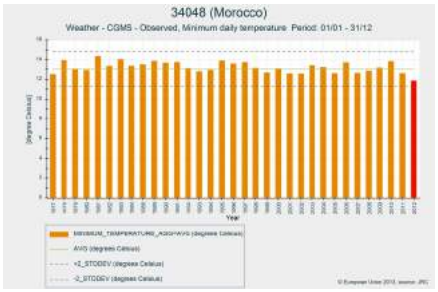
(d) Number of heat wave



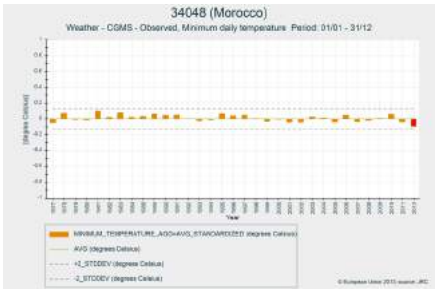
(e) Average maximum temperature



(f) Standard average maximum temperature

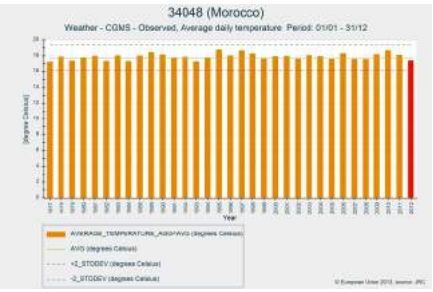


(g) Average minimum temperature

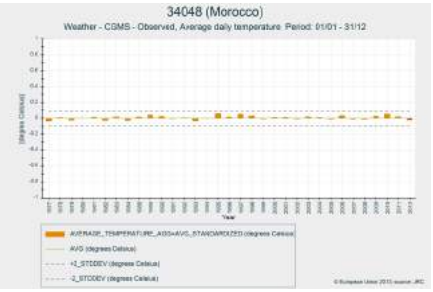


(h) Standard average minimum temperature

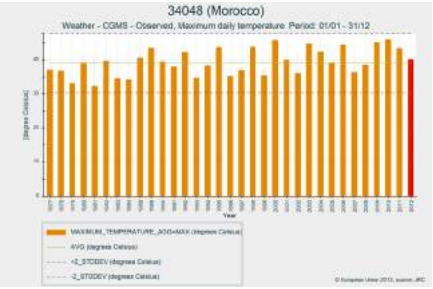
Figure 5: different graphs generated for one of the six grids covering the Mamora site (part 2)



(a) Average mean temperature



(b) Standard average mean temperature



(c) Absolute maximum temperature

(d) Absolute minimum temperature

Figure 6: different graphs generated for one of the six grids covering the Mamora site (part 3)

- Standard maximum temperature
- Minimum temperature (daily average)
- Standard minimum temperature
- Average temperature (daily average)
- Standard average temperature

No significant outlier appears for temperatures and the standard temperature index remains below 0.15 and for 2012 around 0.10. From the overall analysis no significant time series reduction or increase can be seen however a 5-6 years cycles is visible in the standardized temperature graphs.

These different indicators can be considered as a basic data set for preliminary analysis that should be completed by local data and national expert interpretation. The proposed approach can be reproduced for the other Mediterranean sites by the local experts without any specific software a part from Access. Due do the spatial and historical indicators consistency it can also be used for sites comparison.

4 Vegetation cover

In order to look for potential changes in the vegetation cover we used the SPOT VGT satellite that have consistant time series from 1998 and being global can cover the different Mediterranean sites. The well known Normalized Difference Vegetation Index that is linked to aerial phytomass was analysed through different maps and graphs that can be used to detect changes in the land cover. Two types of maps were generated each ten day over Mamora forest:

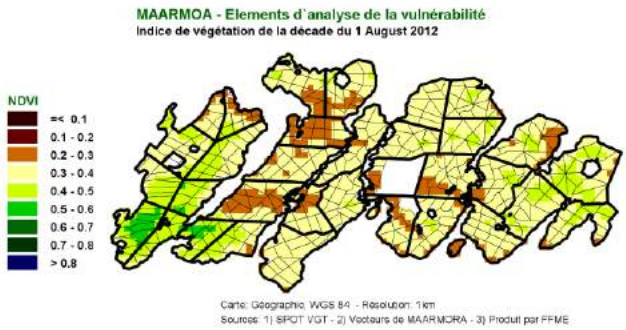
- ten day NDVI
- ten day NDVI anomaly ([2012-Mean]/2012)

For the graphs the NDVI aggregation was made at canton and group levels over different time periods. Three types of profils were produced:

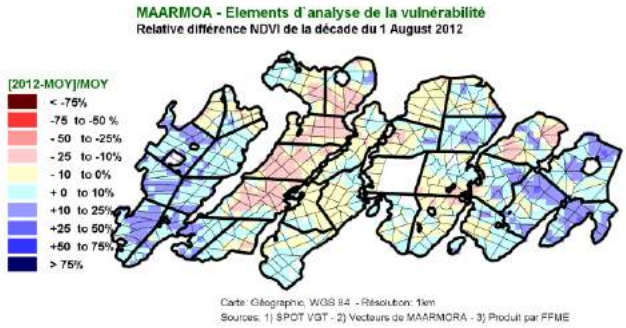
- ten day NDVI from 1998 to 2012 with the long term MIN/MAX/AVERAGE
- yearly NDVI from 1998 to 2012 with the yearly MIN/MAX/AVERAGE

4.1 Ten day NDVI and anomalies maps

The first ten day set can be used to analyse a specific ten day anomaly or sequences of anomalies when the yearly data set can be used for any specific season analysis with comparison with long term average.



(a) Example of ten day NDVI in 2012



(b) Ten day NDVI anomaly(2012-mean)/mean

Figure 7: Ten day NDVI and anomalies maps

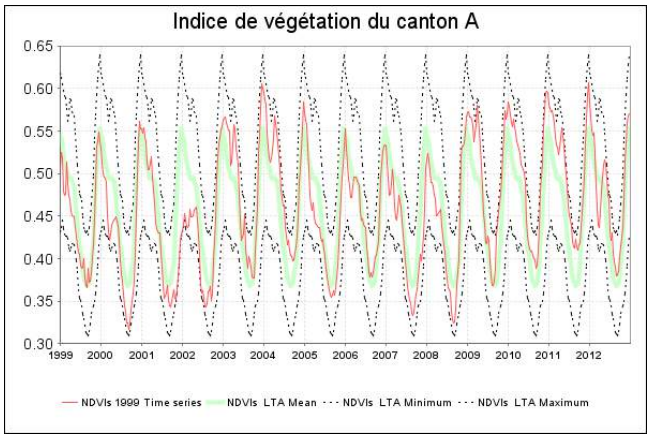


Figure 8: Ten day NDVI evolution from 1999 to 2012 (Profils per canton)

4.2 Ten day NDVI evolution graphs

To get a temporal overview of the NDVI evolution three graphs products were generated at canton or group levels.

4.2.1 Overall view of the time serie profil comparing the value of the year with ten day min/max/average

It gives an overview of the historical time series and allow a quick identification of the extreme years. For example in the Canton A the vegetation index in 2002 is the lowest profil of the time series probably as a results of very dry condition. (see the 2002 portion of the above graph much below than average).

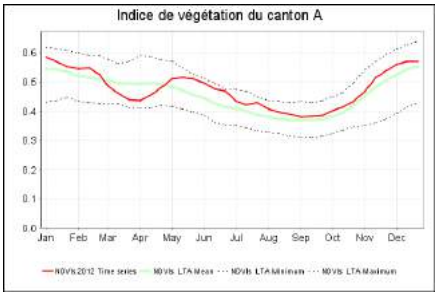
4.2.2 Specific year (here 2012) compared with long term MIN/-MAX/AVERAGE

Such profils are used to focus on a specific season and analyse

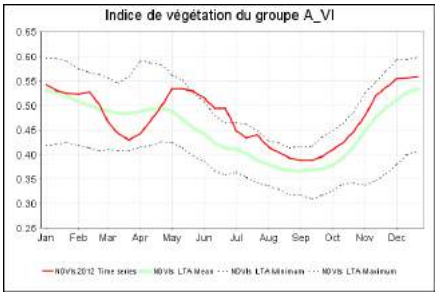
- when the vegetation condition start to be below or above the normal condition
- how important are the anomalies compared with the MIN/MAX/MEAN
- and the duration of such anomalies to analyse the potential impact on the vegetation growth and as an element of the vulnerability analysis.

4.2.3 Yearly NDVI from 1998 to 2012 with the yearly MIN/MAX/AV-ERAGE

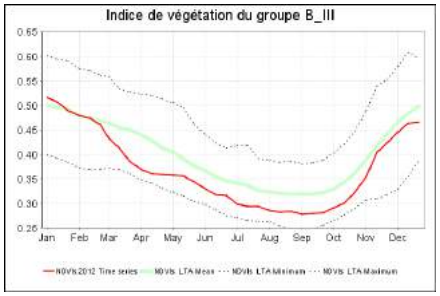
It is based on a synthetic indicator like mean/max/min NDVI which is calculated for each year. Display on a graph at group or canton level it shows the time



(a) 2012 NDVI compared with MIN/-MAX/AVERAGE Profils per canton

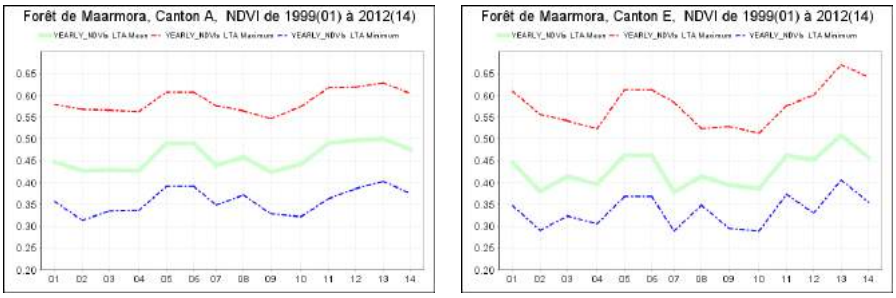


(b) 2012 NDVI compared with MIN/-MAX/AVERAGE. Profils per group (ex. of higher NDVI in 2012)



(c) 2012 NDVI compared with MIN/-MAX/AVERAGE. Profils per group (ex. of lower NDVI in 2012)

Figure 9: Specific year (here 2012) compared with long term MIN/MAX/AVERAGE



(a) canton A (b) canton E

Figure 10: NDVI yearly MAX/MIN/AVERAGE evolution from 1999(01) to 2012(14)

series evolution and give a synthetic view of the overall evolution of vegetation for the last 13 years. For example the graphs of Canton E shows that 2003 and 2004 experienced good vegetation development and after lower indices there is a progressive vegetation increase from 2009.

Once again as for the climatic indicators these SPOT VGT NDVI can be downloaded freely for any Mediterranean site of the FFEM. The different maps and graphs can be produced by the national experts by using the free SPIRITS software. Due to the spatial and historical consistency these indicators can also be used to make inter comparisons of the differents sites. Analysis from the local experts are required to interpret the possible anomalies and changes and link them with the specific local conditions within the frame of the vulnerability study.

5 National forest inventory data base

5.1 Input data

National forest inventory (NFI) data were provided as polygons in vector shape files. Spaces in all filenames were replaced with underscores due to compatibility issues with a Linux file system. The following convention of the file names was followed: `canton<C><G>_strate.shp`, with `<C>` the canton [A-E] and `<G>` the group (I, II, III, ...). Each polygon contained three attributes (fields of type String) from which ESSENCES was the most informative for this study.

The ESSENCES field contained the main vegetation types within the forest stand. In case of a mixture of species, up to four species were listed. In addition to the vegetation type, the ESSENCE string contained: two characters [Rg, Rb] for planning, one integer [1-4] for forest cover, one character [f,t] for trunk type and one character [j,a,v] for the tree age (see Table 4). For example, “Rg Qs3fj Qs2fa “ indicated a mixed forest stand that was regenerated (Rg). The stand contained a mixture of both young (j) and adult (a) *Quercus suber L.* with a

Type	comment
Qs	*Quercus suber L.*
Vs	*vide sylvatique*
Eu.ca	*Eucalyptys Camalendulensis (dominant Eucalyptus species in Mamora)*
ac	*Acacia*
Pcm	*Poirier de la Mamora*
A.mo	*Acacia Mollissima (dominant acacia species)*
A.cy	*Acacia cyanophylla*
C.cu	
Eu.si	*Eucalyptus Sideraux Xylon*
Eu.gr	*Eucalyptus graindis*
Eu.cla	*Eucalyptus Cladocalyx*
Eu.go	*Eucalyptus Gonphocephala*
P.h (Ph)	*Pins Halepinsis*
P.pa	*Pins pinaster*
P.r (=P.rad)	*Pins radiata*

Table 3: Type information in the existing field ESSENCE

respective tree cover of $5-25\%$ (3) and $25-45\%$ (2). All plants were seeded (f).

5.2 Re-processed data

To facilitate queries of specific data within the data base, the information contained in the provided vector shape file was re-organised. In particular the string in the ESSENCE field was parsed and classified into a number of new fields: type, density, plan, trunk and stage. The new structure of the data base is shown in Tables 5-9.

Within the forest, the forest types were reduced to four main types: Acacia, Eucalyptus, Pine and *Quercus suber L.*. Together with a no forest class, the forest type map consisted of five classes (see Table 5).

The processing of the provided vector shape files was hampered by some irregularities in the ESSENCE field. Extra effort was needed in an attempt to regularize them. As an example, there was no consistent use of age values. New mixed values were introduced, e.g., Rg Qs1fj/a. The use of spaces within the string was not used consistently either:

- Both "RgQs2fj" and "Rg Qs2fj" were used
- Both "RgQs3fjQs2fa" and "Rg Qs3fj Qs2fa" were used.

The result was stored in a new vector shape file, `nfi_mamora.shp` containing a triplet for each of the new fields, e.g., type1, type2 and type3. The reason for the triplets is because of the mixed forest stands. Up to three different species were reported for each stand. In case of homogeneous stands, only the first entry was used. In addition, the original ESSENCE field was retained for

value	description
Rg	Forest re-generation (*Quercus suber L.*)
Rb	Re-forestation (*Pins*, *Acacia*, *Eucalyptus*)
1	cover $> 45\%$
2	$25\% <$ cover $< 45\%$
3	$5\% <$ cover $< 25\%$
4	cover $< 5\%$
f	futaie (seed-tree or coppice with standards)
t	taillis (coppice)
j	jeune
a	adulte
v	vieux

Table 4: Additional information in the existing field ESSENCE

String	name of tree species	value in GeoTiff map
a	Acacia	12
eu	Eucalyptus	13
p	Pine	14
qs	*Quercus suber L.*	11
NF	No forest	20

Table 5: new field for type in new structure of data base

value (Integer)	description
1	cover $> 45\%$
2	$25\% <$ cover $< 45\%$
3	$5\% <$ cover $< 25\%$
4	cover $< 5\%$

Table 6: new field for density in new structure of data base

value (Integer)	description
0	0 not specified
1	forest re-generation
2	re-forestation

Table 7: new field for plan in new structure of data base

value (Char)	description
t	tailli (Fr)=coppice (En)
f	futaie (Fr)=seed-tree or coppice with standards (En)

Table 8: new field for trunk in new structure of data base

value (Char)	description
j	jeune (Fr)=young (En)
a	adult (Fr=En)
v	viex (Fr) = old (En)
e	no tree cover due to failed regeneration or reforestation

Table 9: new field for stage in new structure of data base

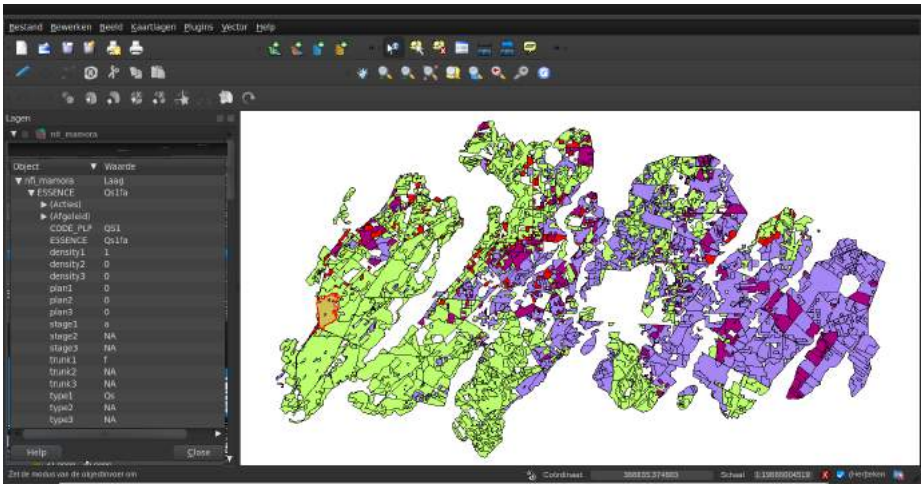


Figure 11: Processed vector shape loaded in a geographic information system

verification purposes. The advantage of the new processed vector shape file is it can easilty be queried for the different new fields. An overview is given in Fig. 11, showing the processed vector shape file loaded in a free and open source geographic information system (QGIS).

6 Remote sensing data

6.1 Image data

Fine spatial resolution remote sensing data (30 m) were downloaded from the USGS Landsat data archive `http://landsat.usgs.gov`. In total, 42 scenes of 170 × 185 km were obtained, covering a period from 1984 until 2010. An overview of available data and acquisition times is given in Table 10. Based on Table 10, the scenes of Table 11 were used. Quicklook images for each of the reference dates (1990, 2000, 2005 and 2010) are shown in Fig. 12.

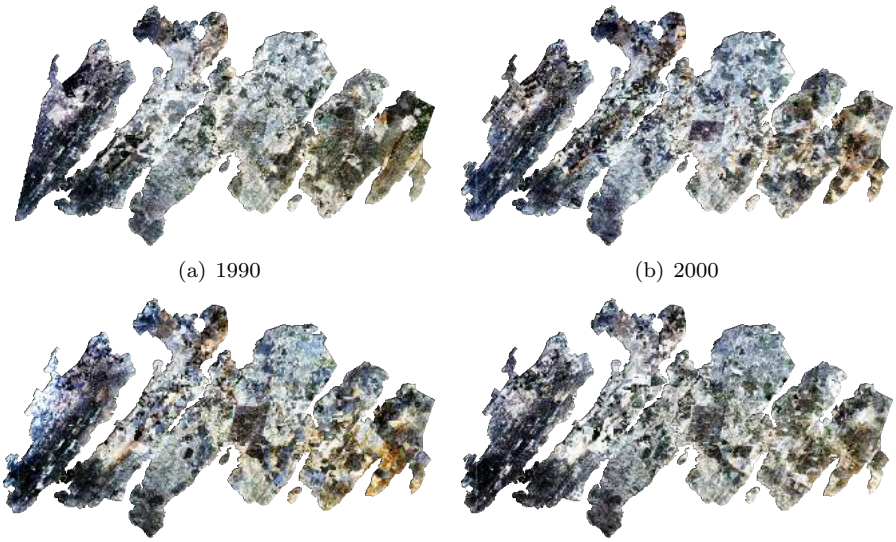
The spectral and spatial characteristics of the Landsat sensor are shown in Table 12. Even though the sensor characteristics of Landsat TM and ETM+ are similar, important differences in the digital numbers acquired by the sensors exist. Data sets from the Landsat 5 TM are not routinely corrected for a num-

year	Jan	Feb	Mar	Apr	May	Jun	**Jul**	**Aug**	**Sep**	Oct	Nov	Dec
1985				15								
1987				5		24	**10**	**27**	**12**			
1988	26											
1999								**20**			8	
2000			15		2		**21**	**22**				12
2001								**9**				
2002		1	21	22	16	17,25			**29**			
2003				25	27		**6**	**7**	**24**			
2007			11		30		**1**					
2009								**23**	**24**	10		
2010				20		7						

Table 10: Acquisition dates for available Landsat data over Mamora site (suitable dates in bold)

reference	year	month	filename
1990	1987	07	LT52010361987191XXX02.tif
	1987	08	LT52010361987239XXX03.tif
	1987	09	LT52010361987255XXX03.tif
2000	1999	08	LE72010361999232EDC00.tif
	2000	07	LE72010362000203EDC00.tif
	2000	08	LE72010362000235EDC00.tif
	2001	08	LE72010362001221EDC00.tif
	2002	09	LE72010362002272EDC00.tif
2005	2003	07	LT52010362003187MTI01.tif
	2003	08	LT52010362003219MTI01.tif
	2003	09	LT52010362003267MTI01.tif
	2007	07	LT52010362007182MPS00.tif
2010	2009	08	LT52010362009235MPS00.tif
	2009	09	LT52010362009267MPS00.tif

Table 11: list of Landsat scenes used for forest mapping



(a) 1990 (b) 2000

(c) 2005 (d) 2010

Figure 12: Quicklooks of Landsat scenes representing the reference years

Sensor code	Landsat number(s)	band	spectral range	spatial resolution
TM	4,5	1	$0.45 - 0.52$ μm	30 m
		2	$0.52 - 0.60$ μm	30 m
		3	$0.63 - 0.69$ μm	30 m
		4	$0.76 - 0.90$ μm	30 m
		5	$1.55 - 1.75$ μm	30 m
		6	$10.40 - 12.50$ μm	60 m
		7	$2.08 - 2.35$ μm	30 m
ETM+	7	1	$0.45 - 0.52$ μm	30 m
		2	$0.52 - 0.60$ μm	30 m
		3	$0.63 - 0.69$ μm	30 m
		4	$0.77 - 0.90$ μm	30 m
		5	$1.55 - 1.75$ μm	30 m
		6a	$10.40 - 12.50$ μm (Low Gain)	60 m
		6b	$10.40 - 12.50$ μm (High Gain)	60 m
		7	$2.08 - 2.35$ μm	30 m
		8	$0.52 - 0.90$ μm	15 m

Table 12: characteristics of Landsat sensors

ber of radiometric and geometric artifacts, including memory effect, gain/bias, and inter-focal plane misalignment. Unfortunately, the images for the four reference years were not acquired with the same sensor, which is problematic for monitoring purposes. Ideally, a radiometric normalization should be performed. However, this was not done for this study.

The available input Landsat scenes for each reference year 1990, 2000, 2005 and 2010 are shown in Table 11. A composit scene was created for each reference year by selecting the median pixel value over all available scenes covered by that reference year.

7 Forest mapping

7.1 Current situation (reference year 2010)

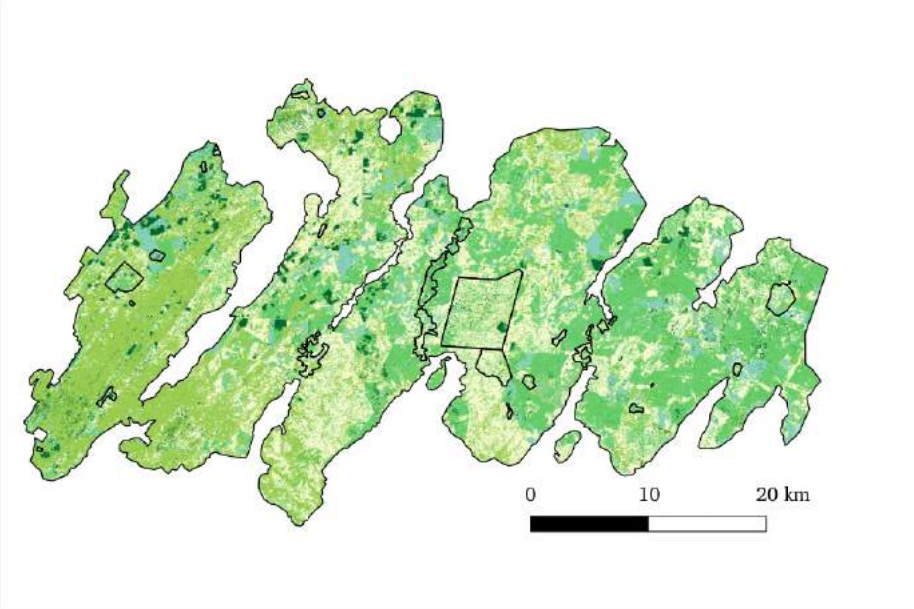
For the classification of forest types, a pixel based support vector machine was selected, which is a state of the art supervised classifier. The training data were derived from the processed NFI reference data. The implementation was based on pktools, an open source software toolbox for remote sensing image processing in a Linux environment[1]. The classification process was automated using a script written in bash (see appendix D.2). Five classes were defined: non-forest, *Quercus suber L*, Acacia, Eucalyptus and Pine.

A training set was selected from the available reference data by extracting the centroid pixel in each forest stand. There was no reference data available for non-forest areas. To train the classifier for non-forest, forest stands with a density less than 25% were selected. For the forest classes the following rules for the forest stands applied:

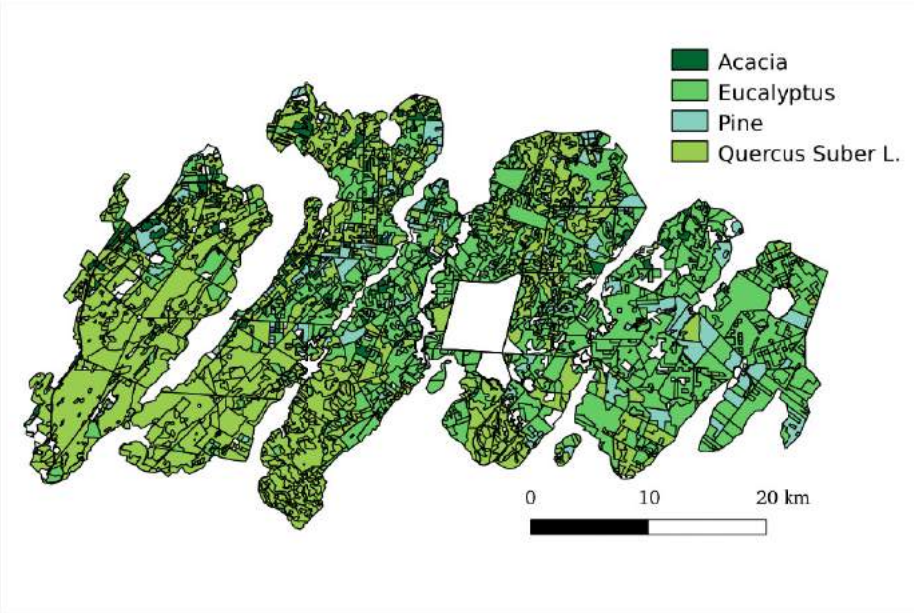
1. density should be at least 45% (attribute *density1* is 1)
2. no mixtures (both attributes *density2* and *density3* are 0)
3. regeneration or reforestation should be successful (attribute *stage* not equal to 'e')

The result of the pixel based classification is shown in Fig. 13(a). Reference data was available only at forest stand level. Therefore, the classification map could only be assessed on a forest stand level. Moreover, no information was available for the non-forest class. Though a real validation was beyond the scope of this study, a preliminary accuracy assessment was conducted using the forest stands in the processed vector shape file. The forest stands were overlayed on the classified forest type map. Then, the majority forest class for the pixels within each forest stand was extracted. Non-forest pixels were disregarded as no reference data was available for this class. The majority class was then assigned to the forest stand as the classified forest type. Doing this for each forest stand in the Mamora forest, a classified forest type vector map was obtained (shown

[1] `http://pktools.nongnu.org`



(a) pixel based forest type map



(b) vector based forest type map based on forest stands

Figure 13: forest type map based on NFI training for reference year 2010

0	*Quercus suber L.*	Acacia	Eucalyptus	Pine
Quercus suber L.	612170	6850	136019	15766
Acacia	3367	25822	2545	180
Eucalyptus	20590	3259	441092	12577
Pine	6810	680	24382	80511
User Acc. (%)	95	71	73	74
Prod. Acc. (%)	79	81	92	72

Table 13: Accuracy assessment for Mamora forest type map 2010: overall accuracy=83% (Kappa=0.72)

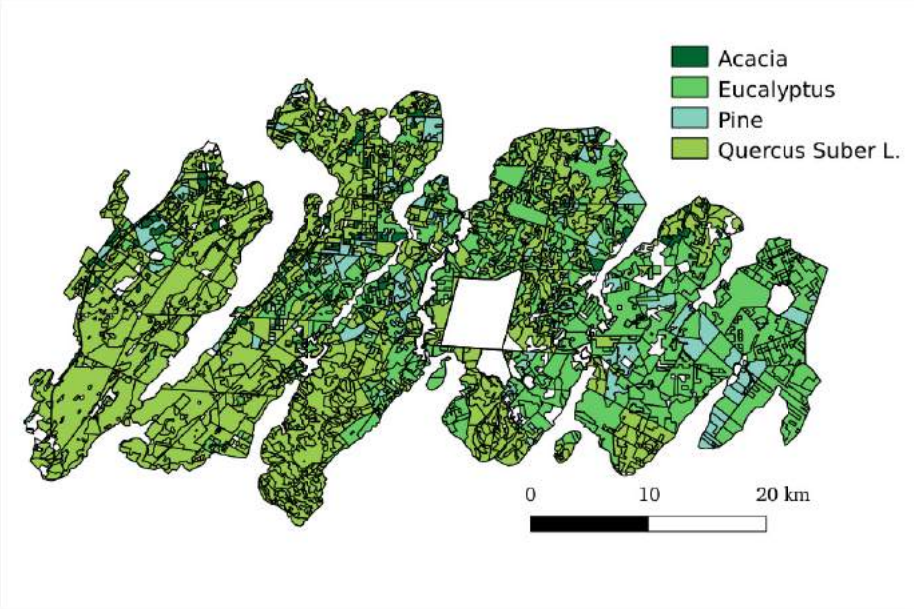
in Fig. 13(b). This vector map was then compared to the labeled vector file from the Moroccan NFI authorities (the processed vector shape file). The areas of the forest stands were taken into account when calculating the classification accuracies.

The accuracy assessment was performed using the 2403 forest stands available in the processed vector shape file. A confusion matrix showing the accuracies is shown in Table 13. To obtain the confusion matrix in table 13, the forest stands were weighted by their area (approximated by the number of pixels covered).

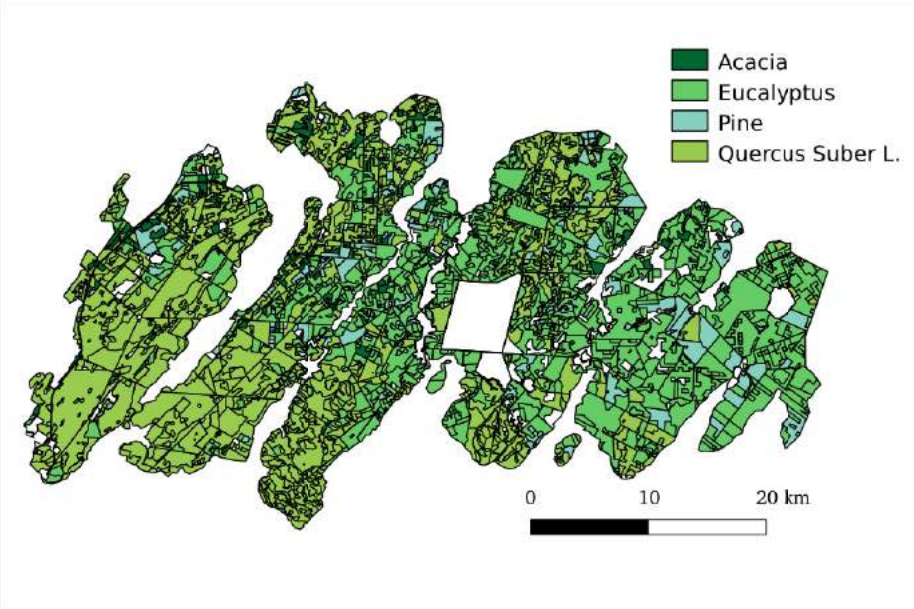
Because the classification concentrated on forest types only, it can be expected that the classification accuracy decreases with forest density level. This was confirmed by conducting the accuracy assessment at different forest density levels. An overall accuracy of 90% was obtained for level 1 (forest density above 45%). At level 2 and 3 (forest densities of $[25-45\%]$ and $[5-25\%]$), the overall accuracy decreased to 72% and 70% respectively. At the lowest level, the test sample was not sufficiently large and not all classes were represented.

The results of the accuracy assessment confirmed the similarity of the classified map to the original map as provided by the Moroccan NFI authorities, as shown in Fig. 14.

The availability of reliable ground reference data is a well known problem. The Mamora forest has been well studied over the last decade. It can be foreseen that for other regions in the Mediterranean area reference data are even more sparse. Though no ideal solution exists, there are a number of approaches that can be followed in that case. Image data of superior quality can be visually interpreted. Typically this is done by local experts on image data covering a selected area at a fine spatial resolution (typically 1-5 m). The idea of superior quality is that the reference product, the result of the visual interpretation, is more accurate than the product to be validated. Image data from Google-earth® is one source of information that is free, provided fine spatial resolution data from the year of interest is available. This approach can be used for training and validation purposes. As an illustration a forest/non-forest map of Mamora has been generated for the reference year 2010, using a small training sample based on Google-earth® imagery (Fig. 15). Another approach is to use unsupervised classification. This method clusters the image data in a number of



(a) NFI reference data



(b) classified forest type map vectorized to forest stands

Figure 14: NFI reference data and classified result for reference year 2010

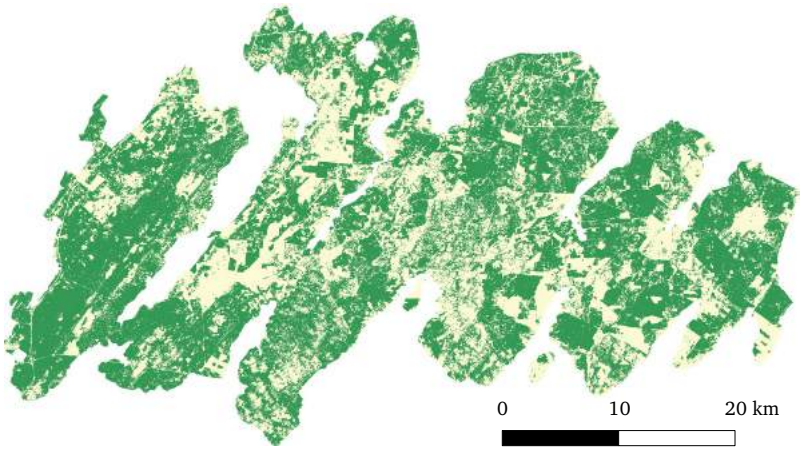


Figure 15: forest cover map based on a small training set based on Google-earth® imagery (reference year 2010)

unlabeled classes. Nevertheless, at some stage, these clusters need to be labeled by, e.g., local experts.

7.2 Mapping forest dynamics through historical data

Unfortunately, no reference data were available in digital format for the reference years previous to 2010. As a result, we were not able to calibrate or validate forest dynamics in a quantitative way. Nevertheless, as illustration, some approaches to derive forest change maps are presented here.

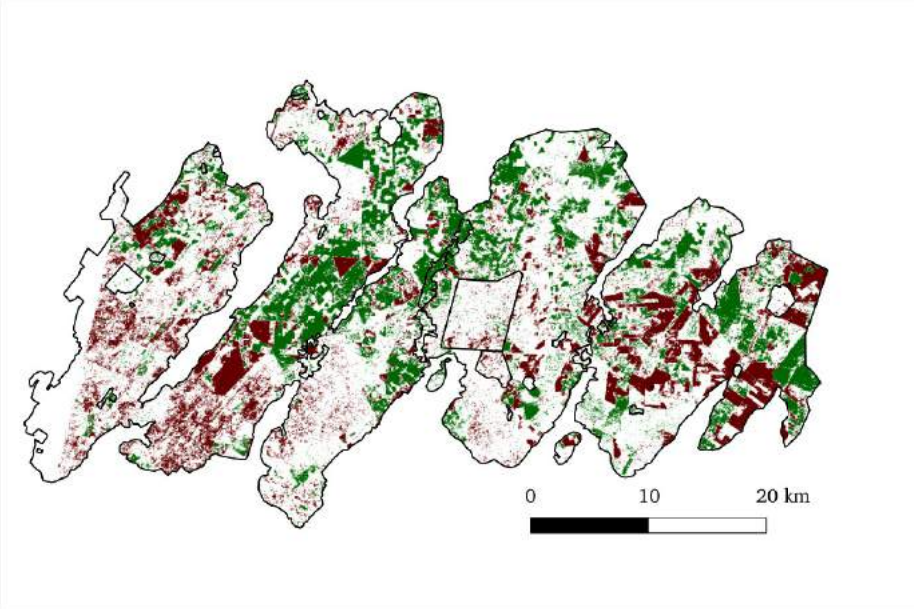
One approach is to conduct a relative analysis using a time series of some index. The normalized difference vegetation index (NDVI) is an example of a quantitative measure of vegetation greenness. For images acquired from July to September, the dry season in the Mamora forest area, the calculated NDVI can be linked to forest cover. For an absolute measure, reference data is still needed for calibration. However, monitoring the NDVI over time provides insight in the forest cover dynamics. The potential forest loss and gain for the periods 1990-2005 and 2005-2010 are shown in Fig. 16. Expert knowledge from the field is needed to interpret these results. Forest change detection is challenging and prone to errors. False detections (errors of commission) are particularly problematic due to changing conditions in acquiring the images (sun and view angles, atmosphere, ...). The advantage of a normalized index such as the NDVI is that some of these factors can dependencies can be reduced. Nevertheless, it

remains difficult to compare indices derived from different sensors and proper calibration is needed. As a result, due to the different sensor characteristics of TM and ETM+ (explained in section 6.1), the dynamics for the year 2000 were excluded.

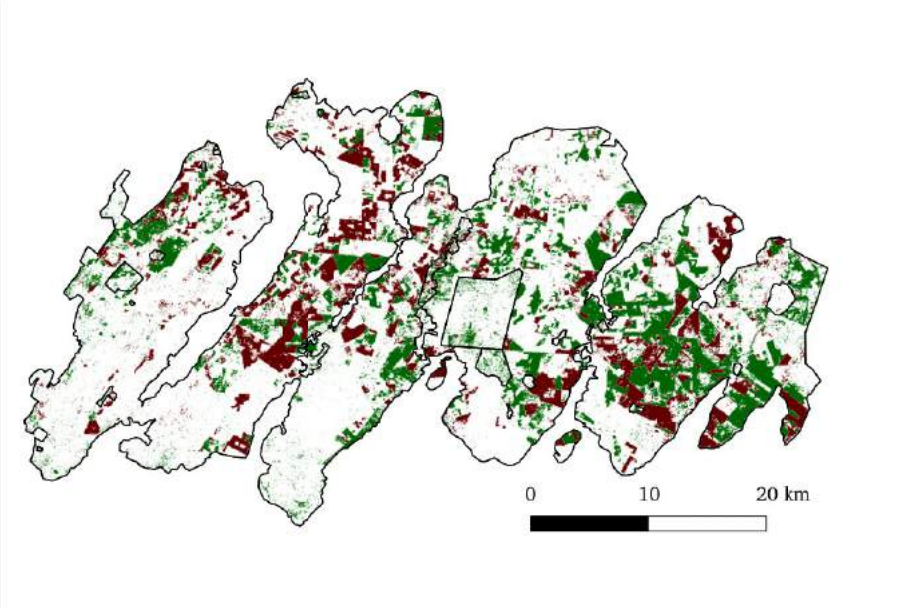
Another approach is to use 2010 reference data as training to classify image data from previous years. It can be expected that parts of the forest area have been changed. As a result, using the 2010 reference data as training for previous years will introduce some errors. Assuming that an important part of the Mamora forest has remained unchanged and the selected classifier was robust to some errors in the training data, this approach was tested for the reference years prior to 2010. The results after aggregation to forest stand level are shown in Fig. 17.

It can be expected that the majority forest type per forest stand remains relatively stable. This is confirmed by the results in Fig. 17. From the pixel based forest type maps, the proportion of an individual forest type can also be calculated for each forest stand. This was done for *Quercus suber L.*, which is an important forest type for the Mamora forest due to its high economic value. There has been an effort over the last decades to increase the proportion of *Quercus suber L.*. This was confirmed by the calculated forest proportion maps for *Quercus suber L.*, as illustrated for canton A in Fig. 18.

Forest type monitoring based on remote sensing methods is not limited to administrative regions that have been defined, sometimes due to historical reasons. It can be interesting to monitor the forest dynamics in new areas that have not been covered by traditional forest surveys. Some examples are green zones (parks) in urban areas and military domains. In the Mamora forest the local population has settled in a limited number of areas. The forest types for the largest of these settlements are shown in Fig. 19. Whereas in the managed forest areas the forest density seems to have increased over the last decades, this is not the case for the settlement. Although no data were available to verify this, it is an interesting result and could be subject of further study in collaboration with the Moroccan NFI authorities.

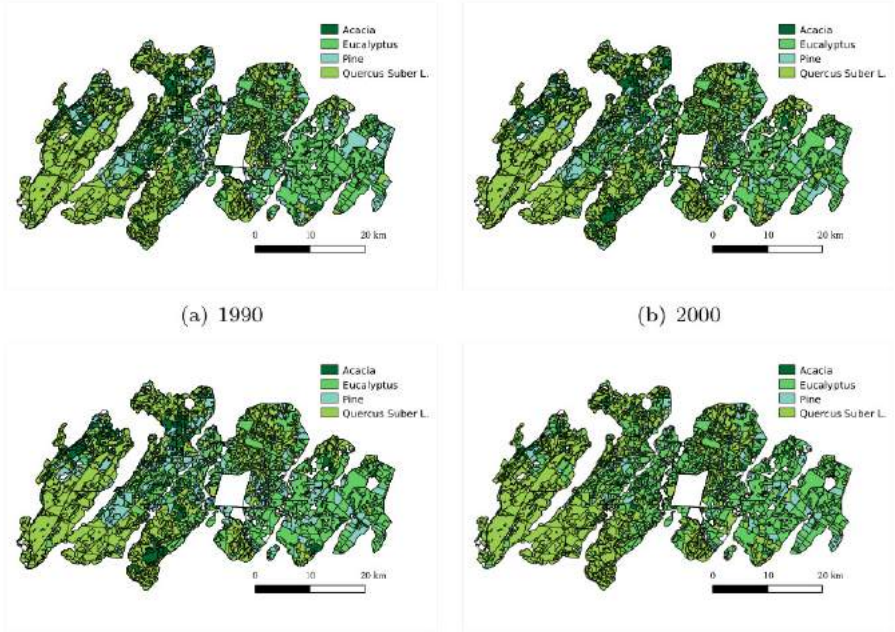


(a) 1990-2005



(b) 2005-2010

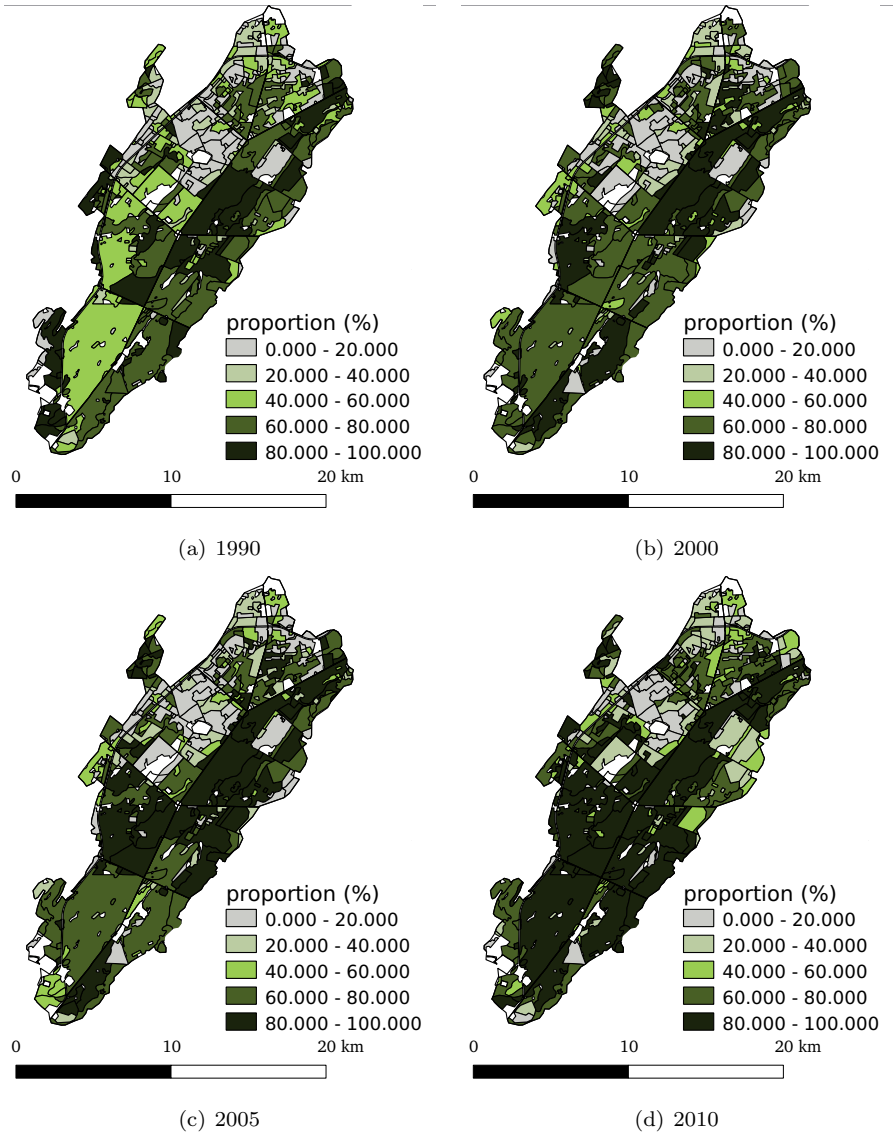
Figure 16: forest cover dynamics based on NDVI (red: potential forest loss, green: potential forest gain)



(a) 1990 (b) 2000

(c) 2005 (d) 2010

Figure 17: vector based forest type maps on forest stand level for different reference years



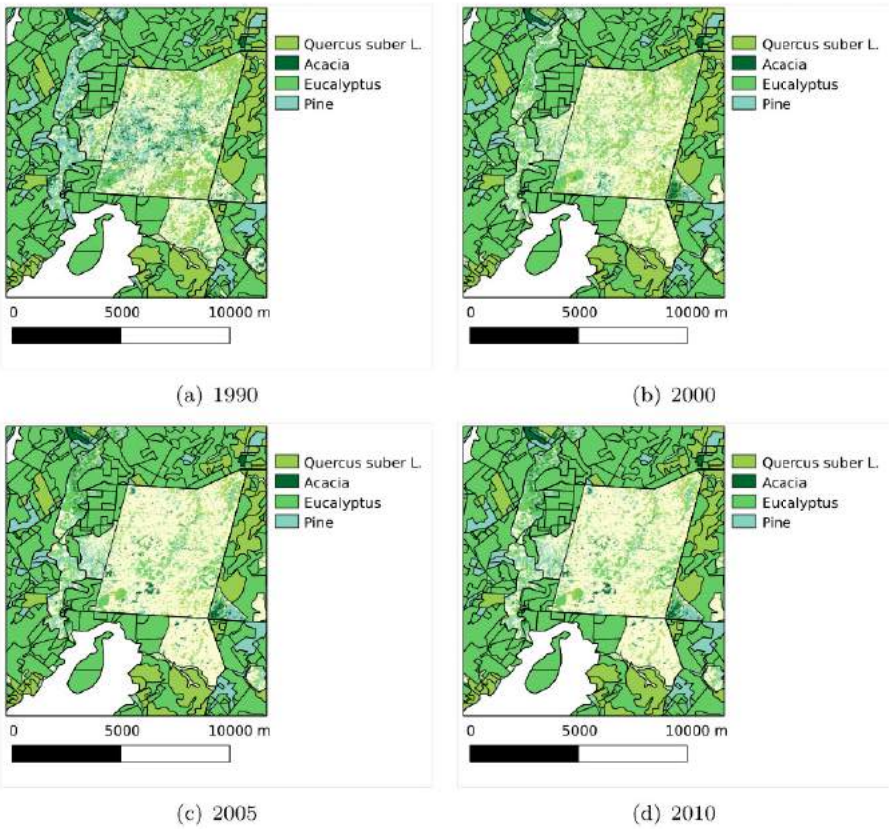
(a) 1990

(b) 2000

(c) 2005

(d) 2010

Figure 18: Proportion of *Quercus suber L.* in forest stands



(a) 1990 (b) 2000

(c) 2005 (d) 2010

Figure 19: Proportion of *Quercus suber L.* in forest stands

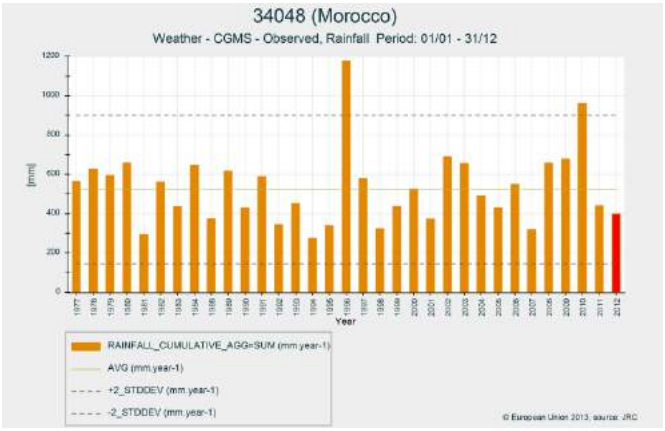
A Dashboard rainfall

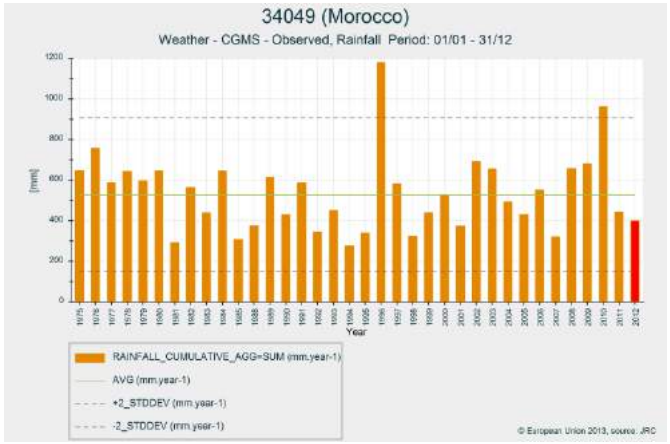
Rainfall panel:

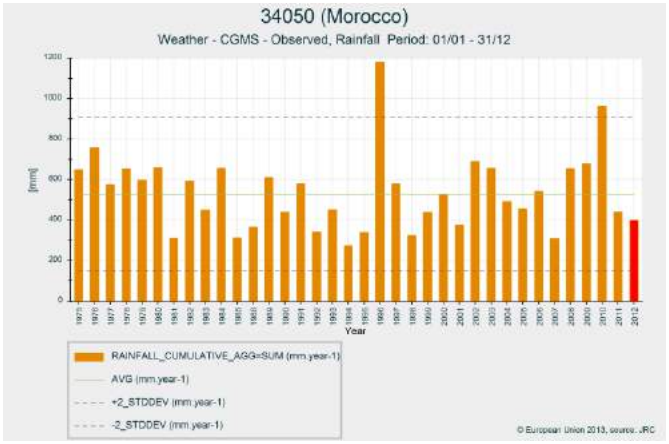
- Yearly rainfall
- Standard Precipitation Index
- Number of days > 5mm
- Number of days > 30mm

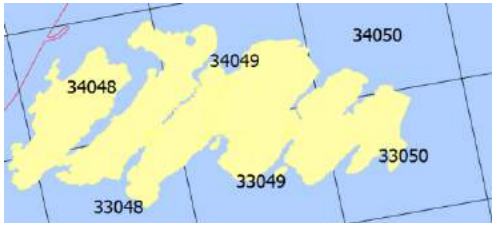
Maamora forest domain

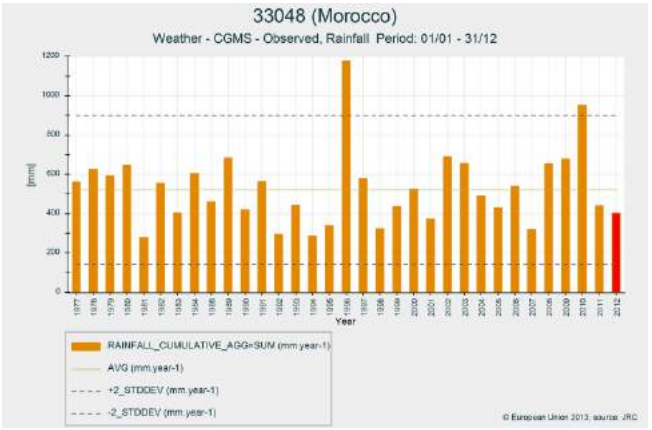
Yearly rainfall

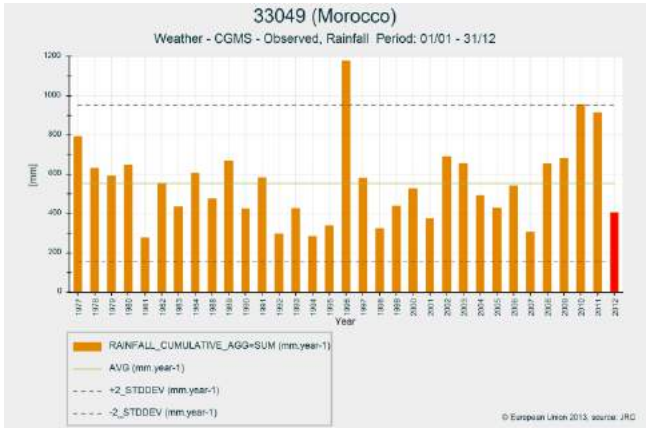


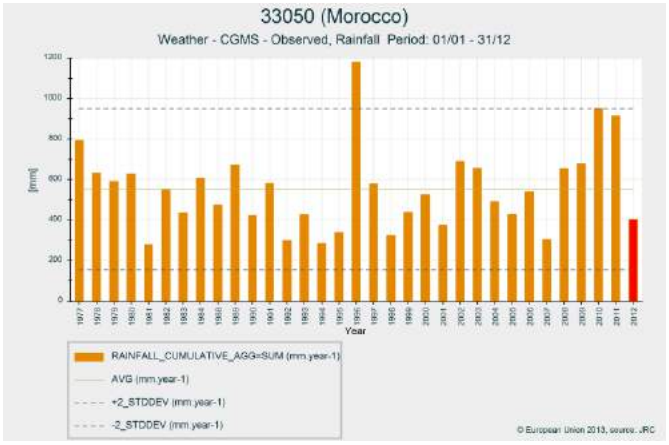






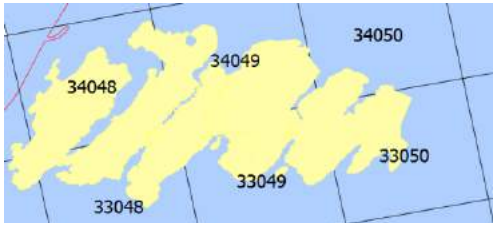


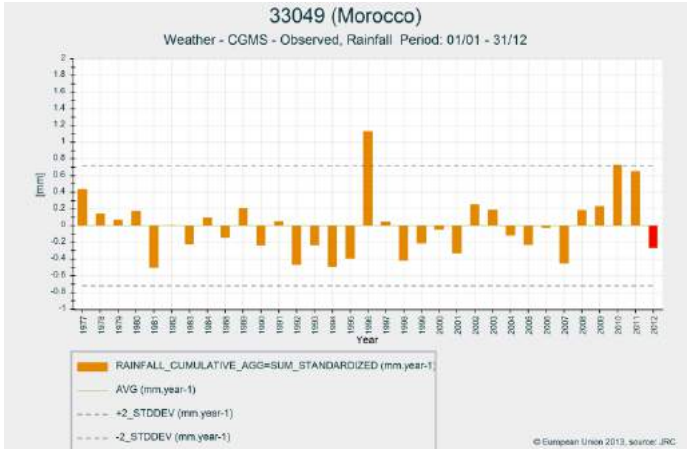




Maamora forest domain

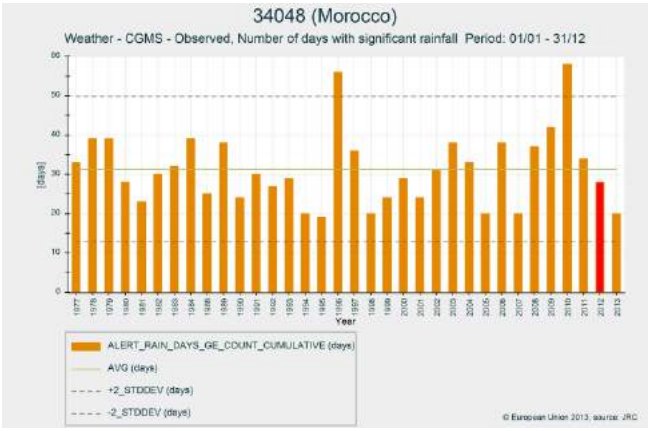
Standard precipitation index

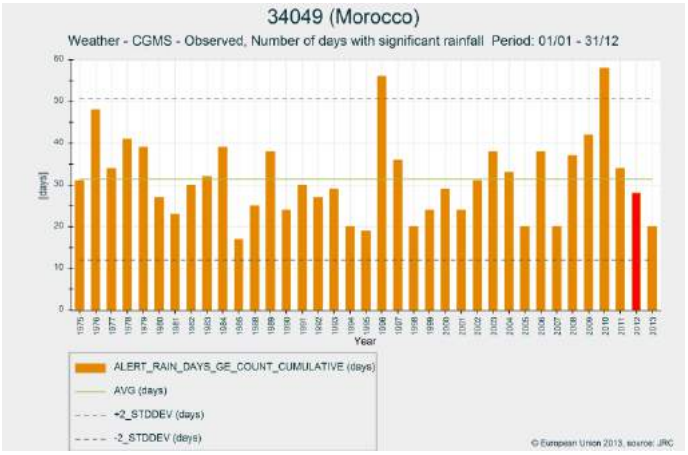


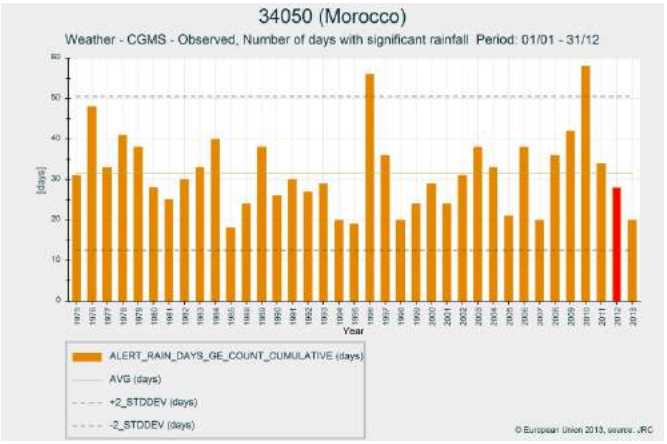


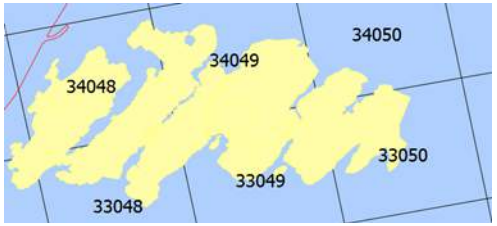
Maamora forest domain

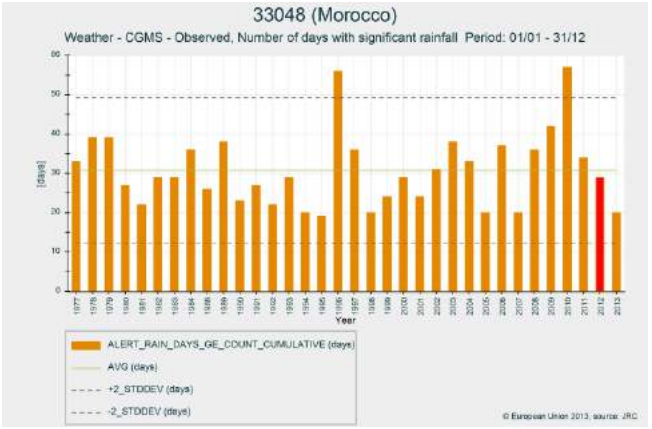
Number of daily rainfall >5mm

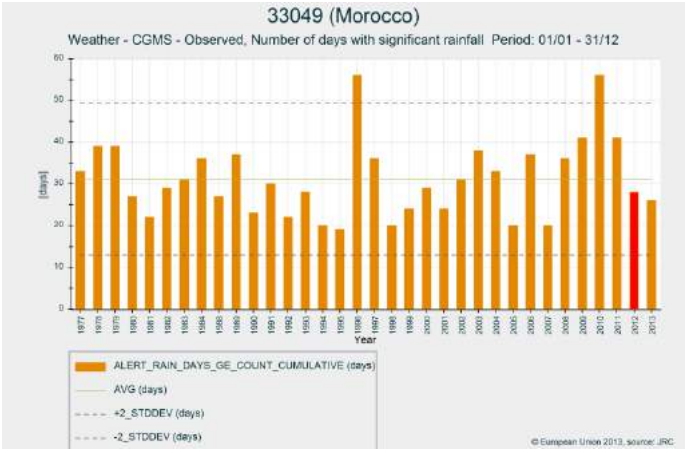


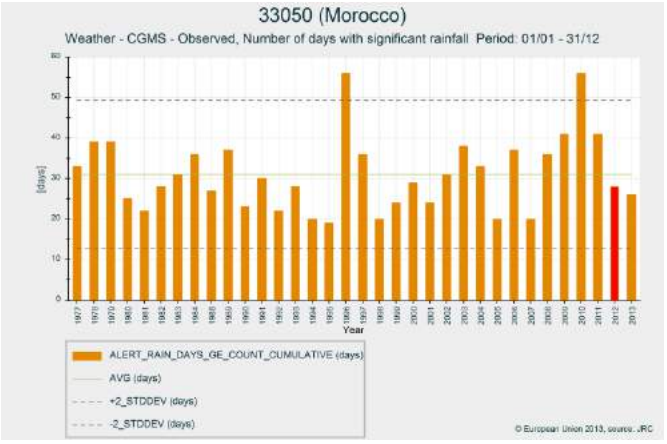






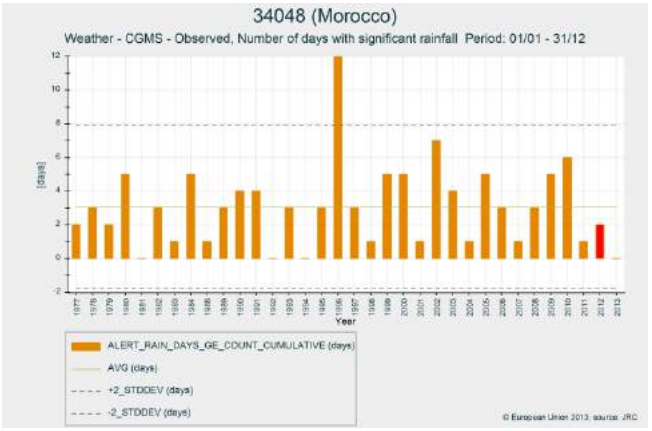


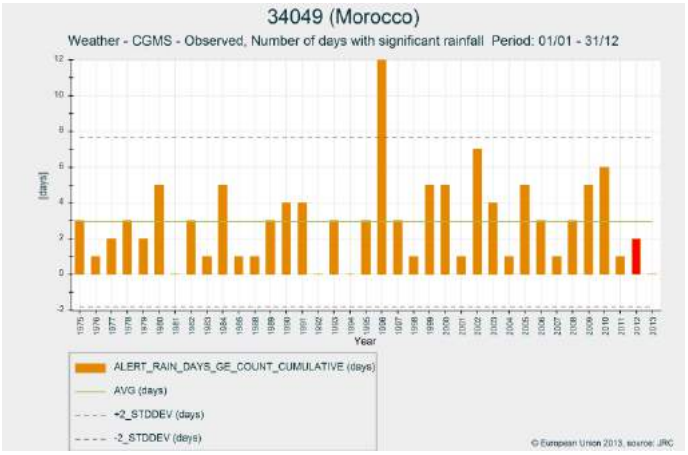


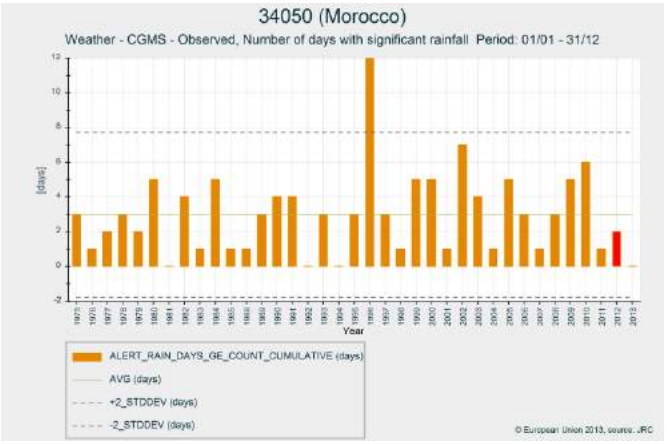


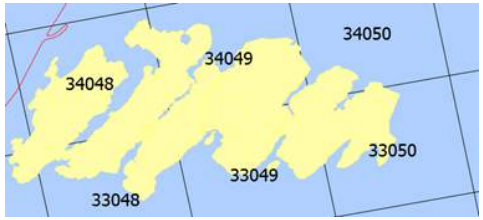
Maamora forest domain

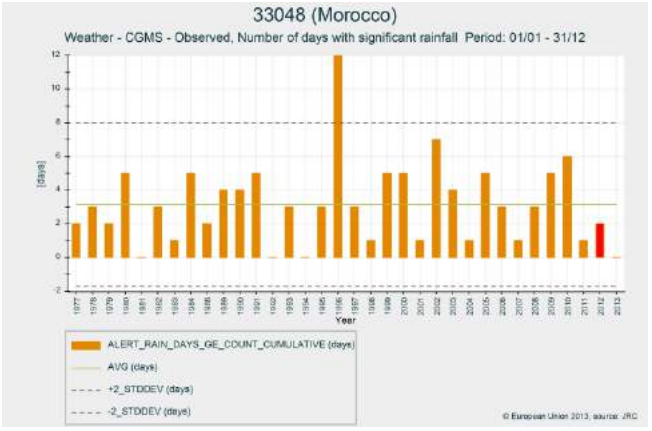
Number of daily rainfall >30mm

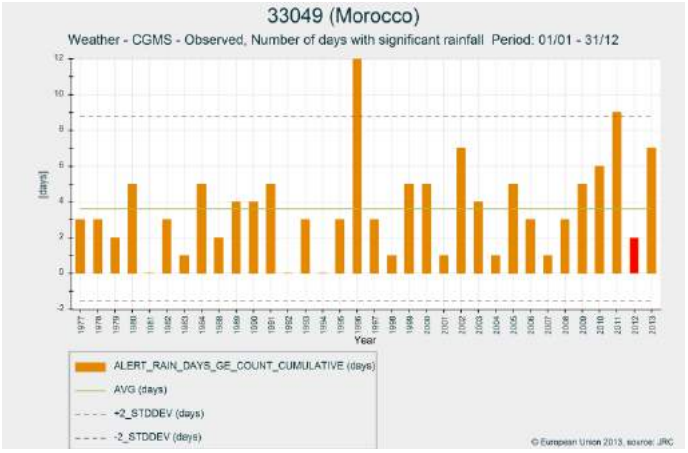


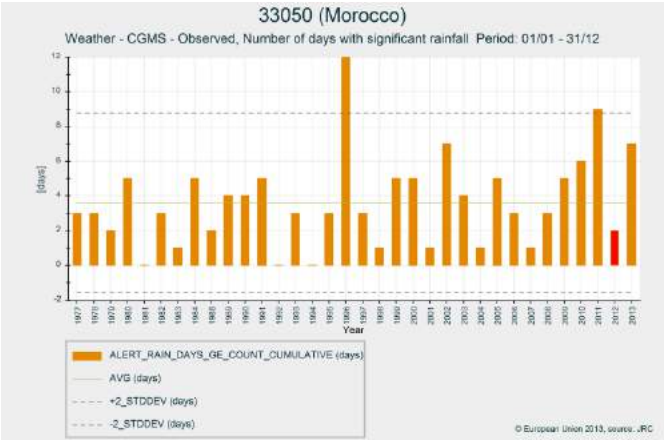












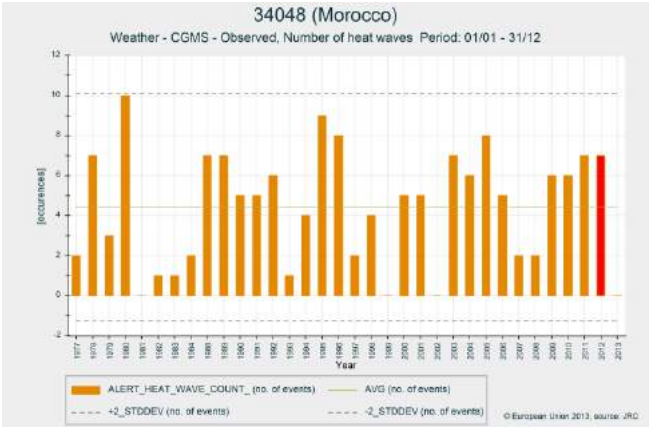
B Dashboard Heat Wave

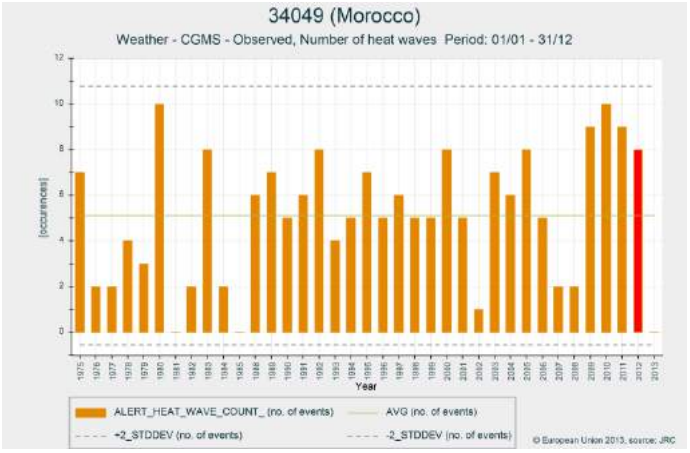
Dash board on the heat wave indicators

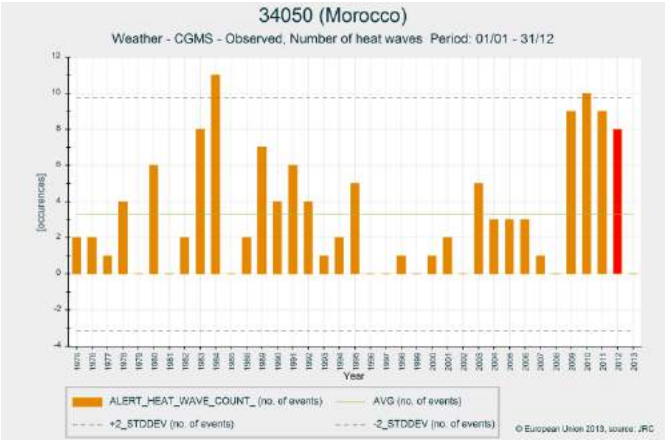
- Number of days >35°C
- Number of days >40°C
- Number of heat waves
- Longest heat wave period

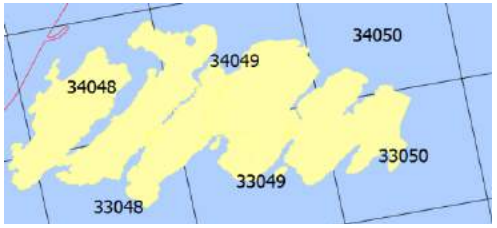
Maamora forest domain

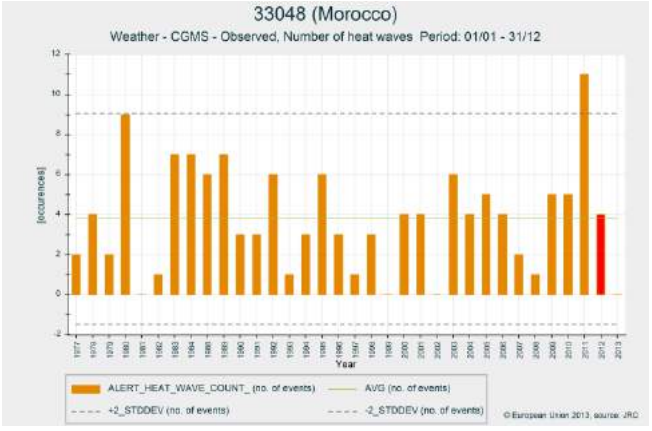
Nb of heat waves (>= 3 consecutive days)

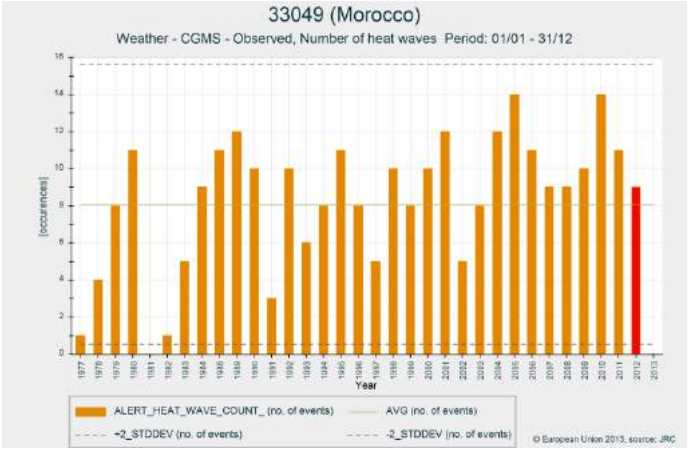


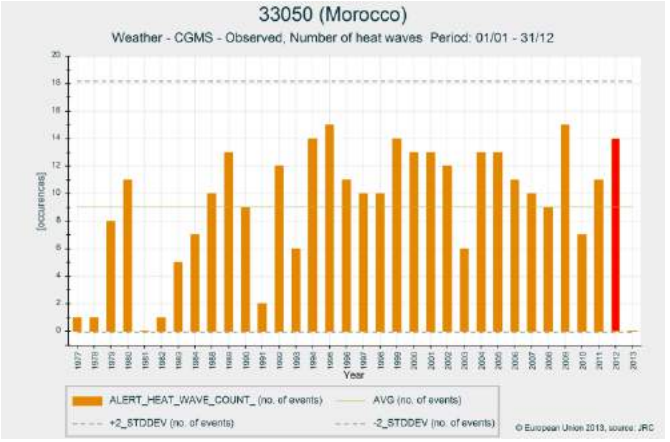






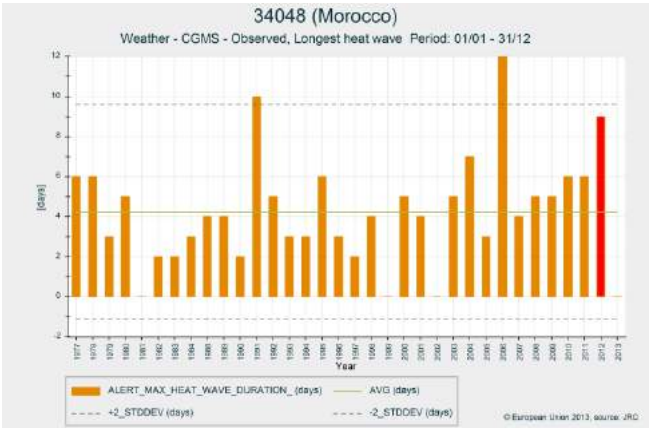


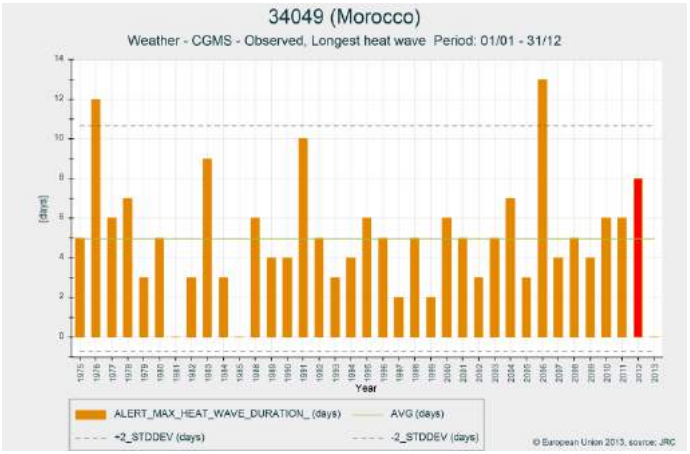


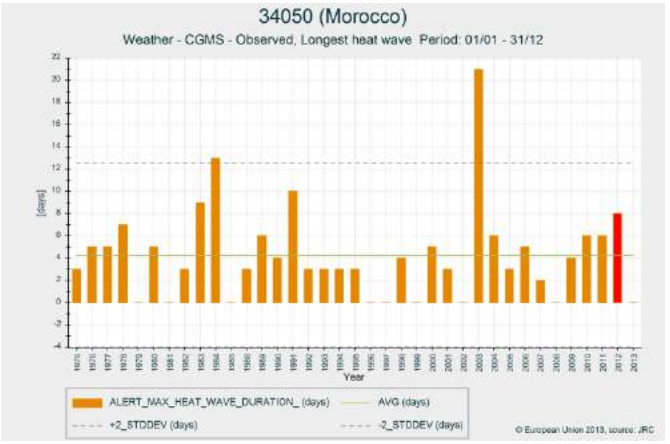


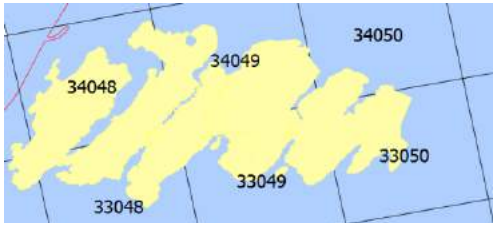
Maamora forest domain

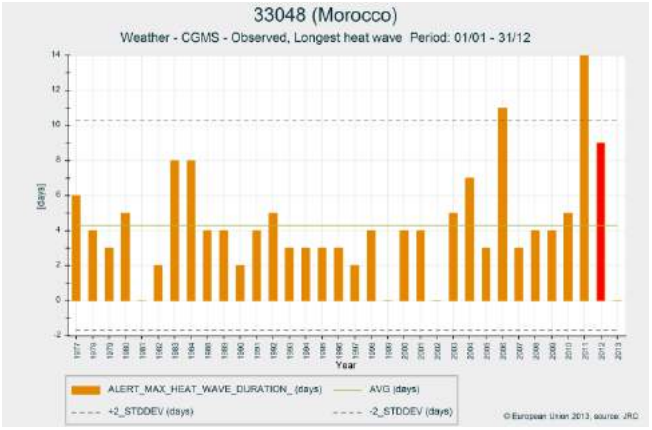
Longest heat wave (consecutive number of heat days)

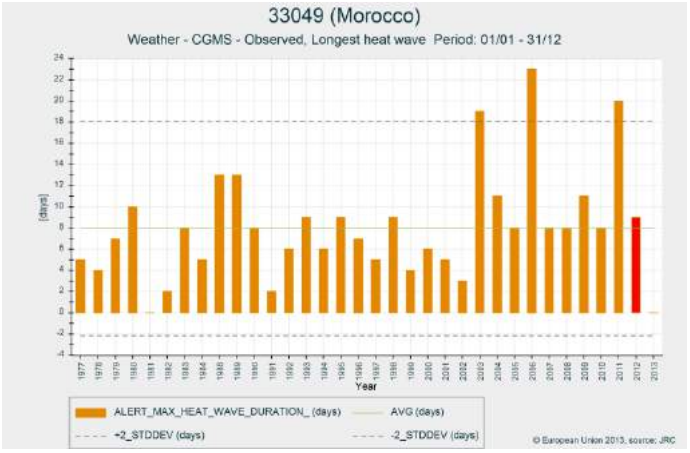


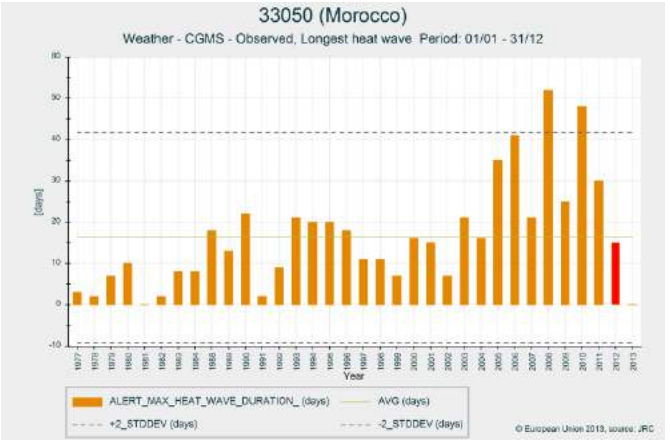






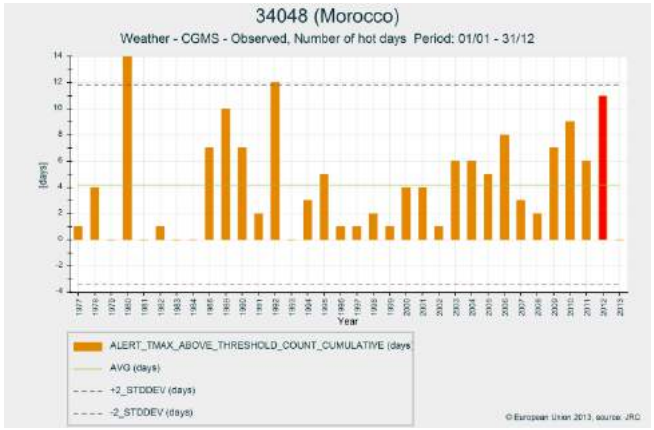


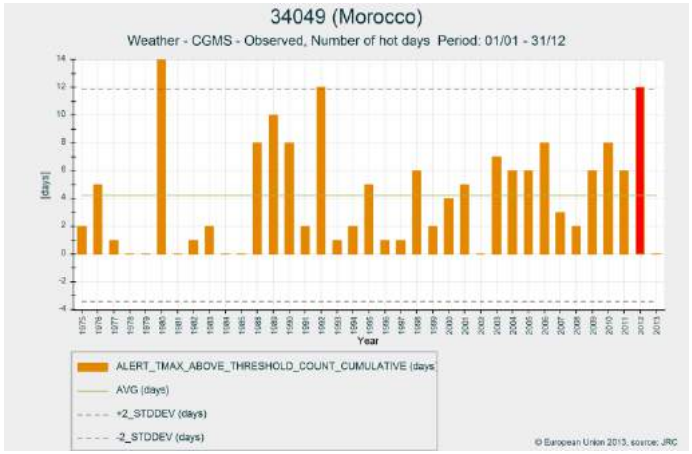


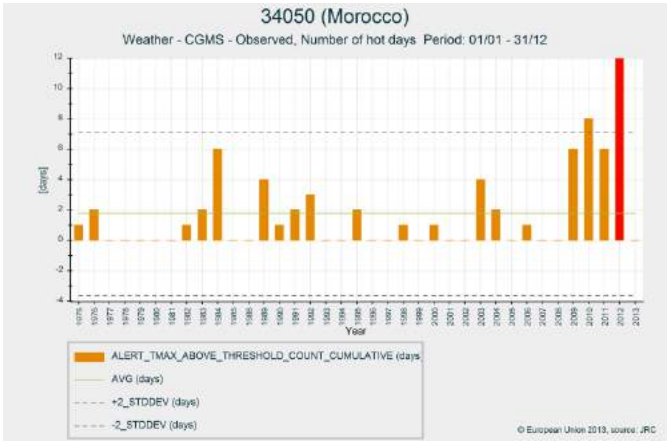


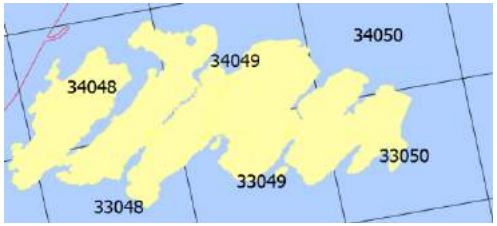
Maamora forest domain

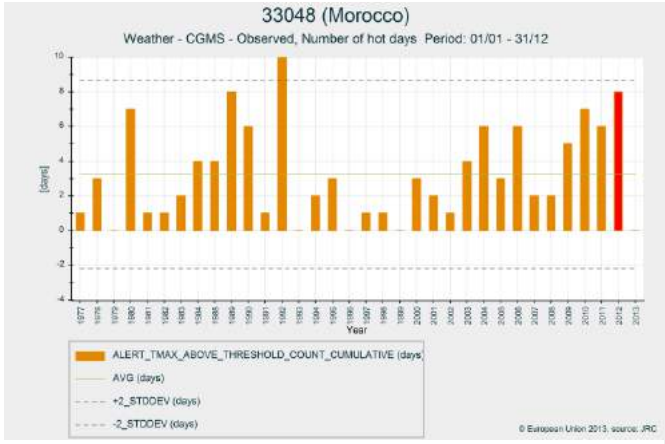
Number of days > 35°C

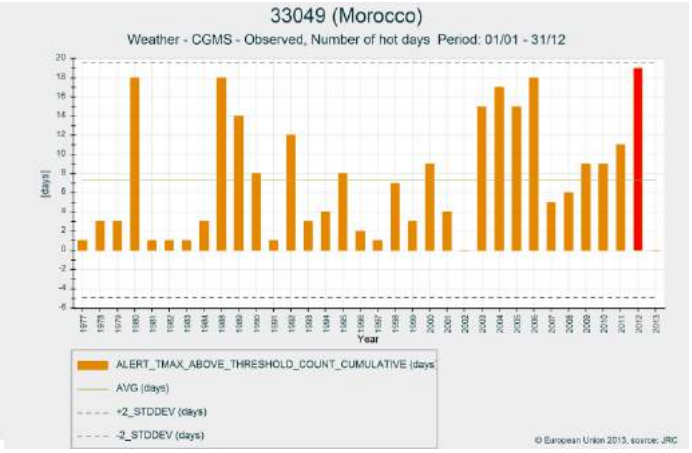


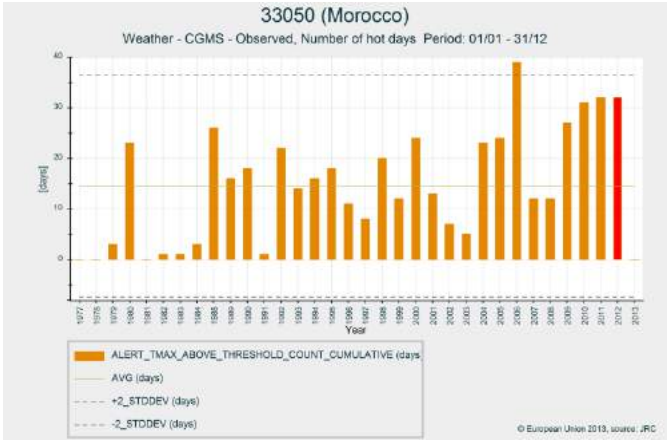






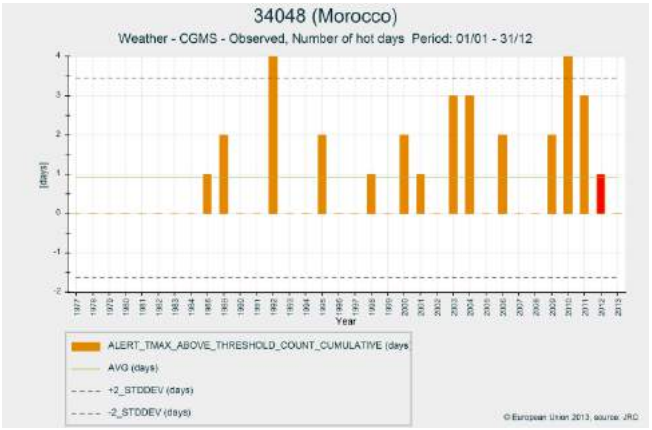


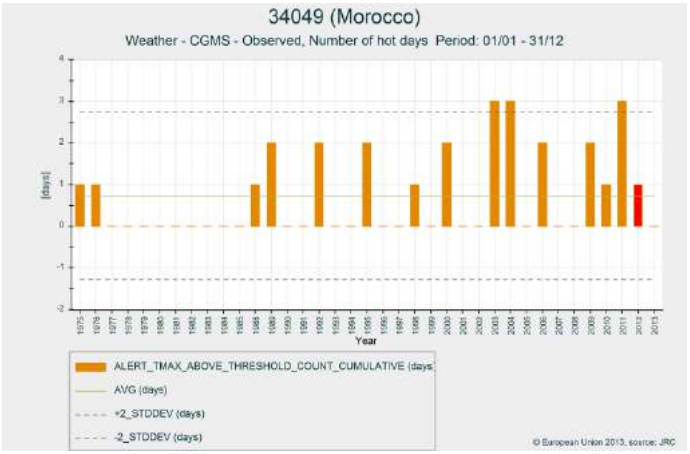


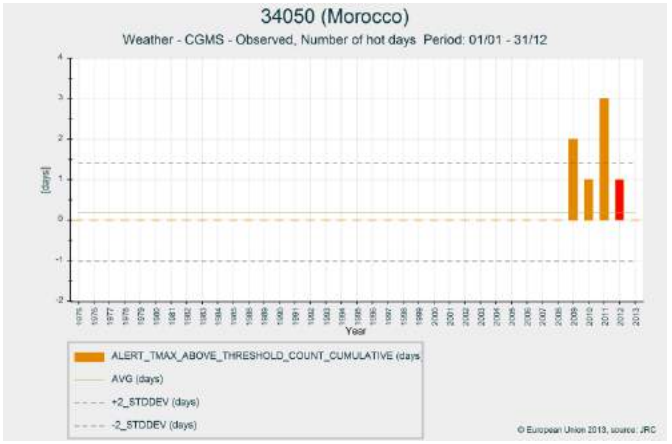


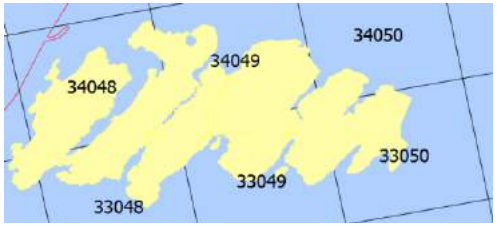
Maamora forest domain

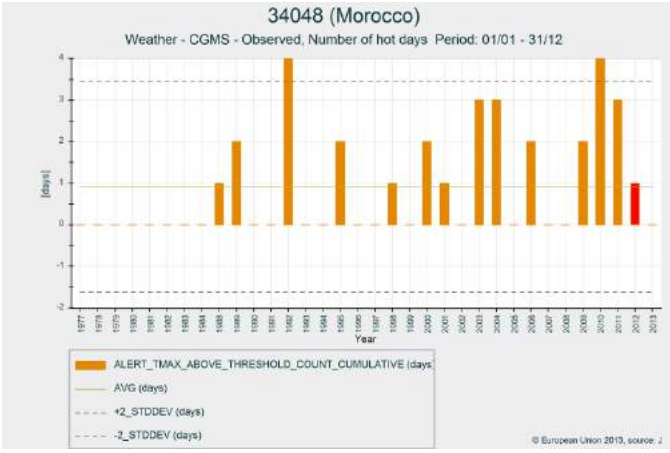
Number of days > 40°C

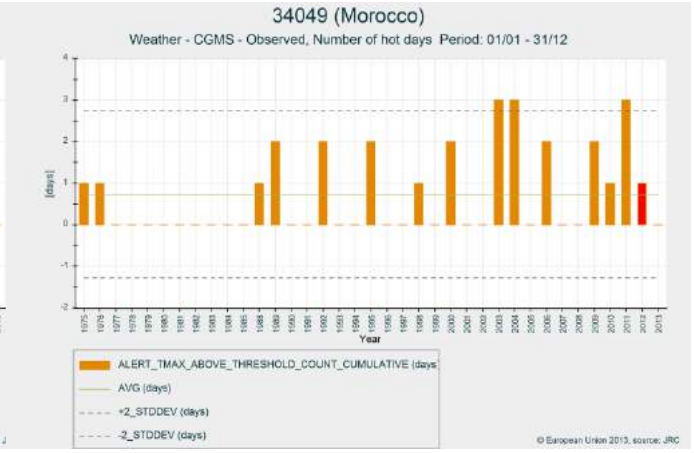


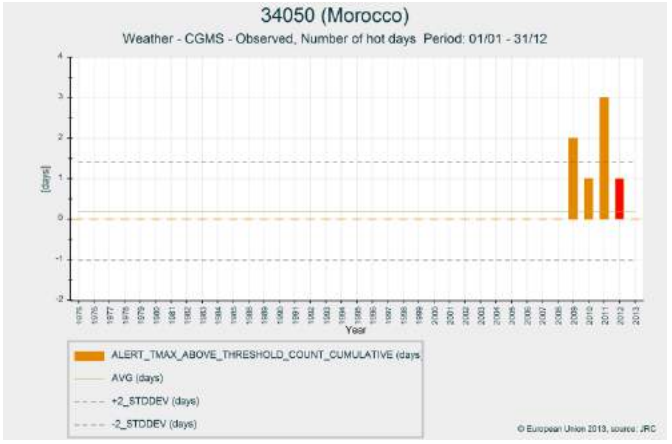












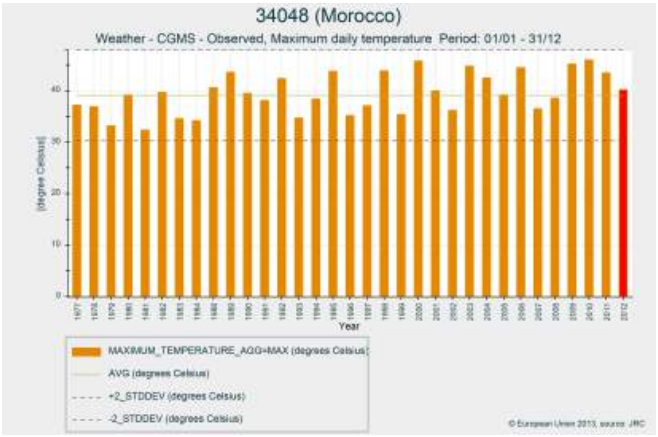
C Dashboard Temperature

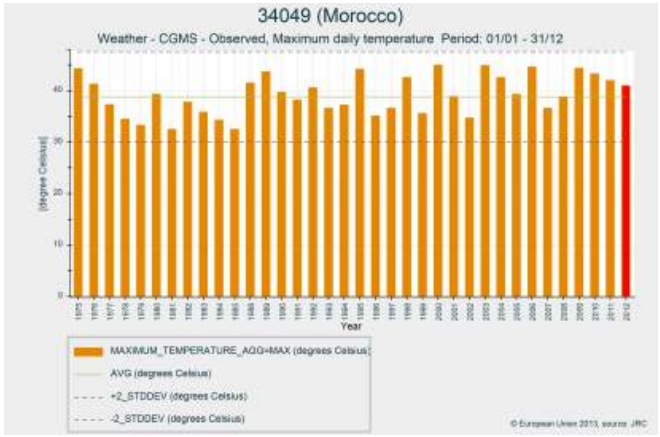
TEMPERATURE panel

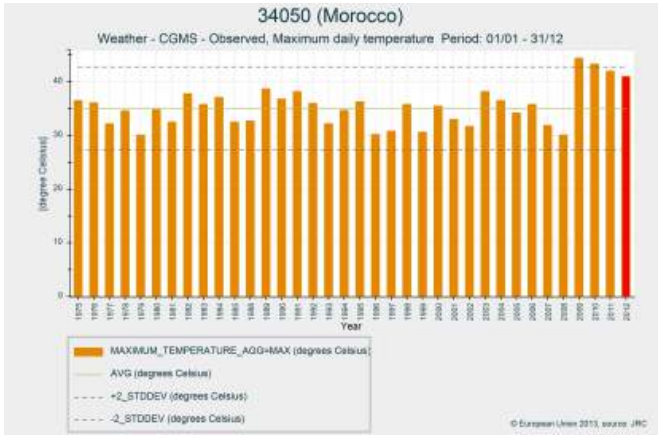
- Maximum temperature (daily average)
- Standard maximum temperature
- Minimum temperature (daily average)
- Standard minimum temperature
- Average temperature (daily average)
- Standard average temperature

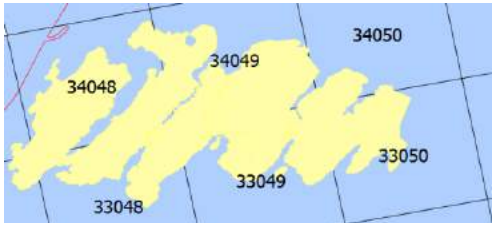
Maamora forest domain

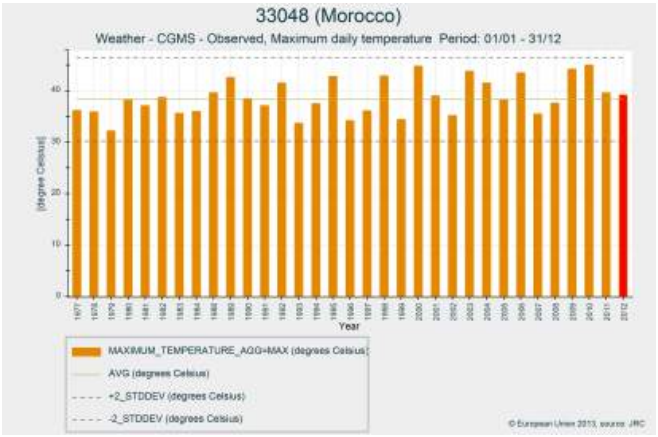
MAXIMUM temperature (daily average)

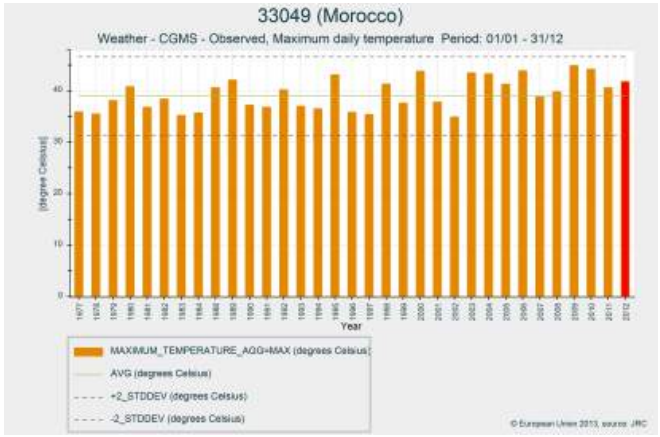


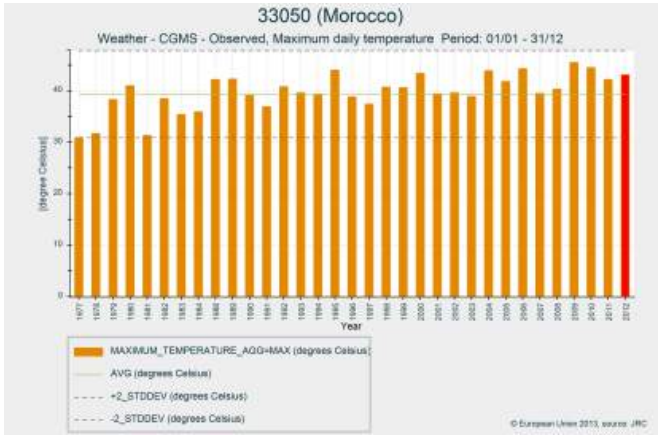






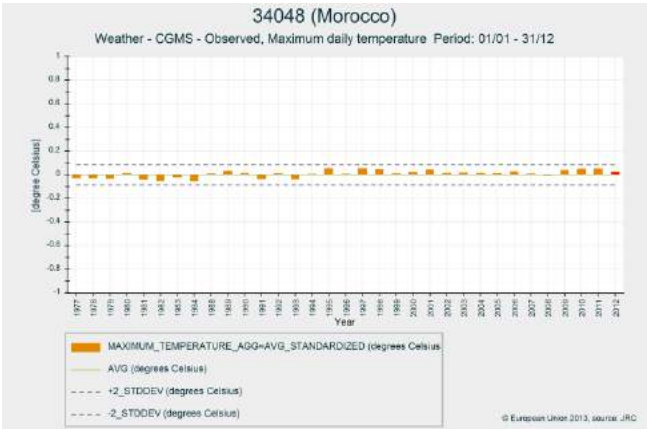


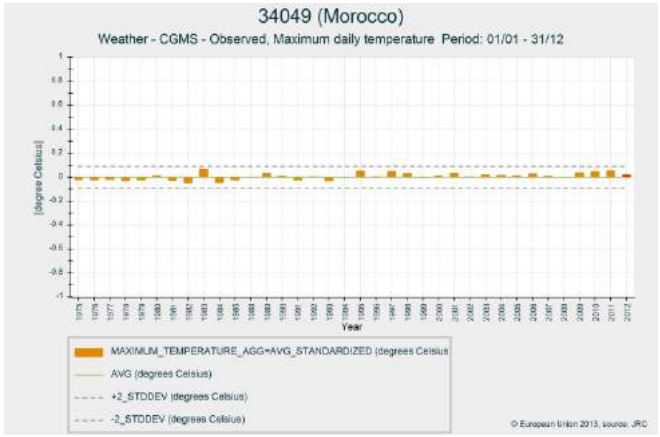


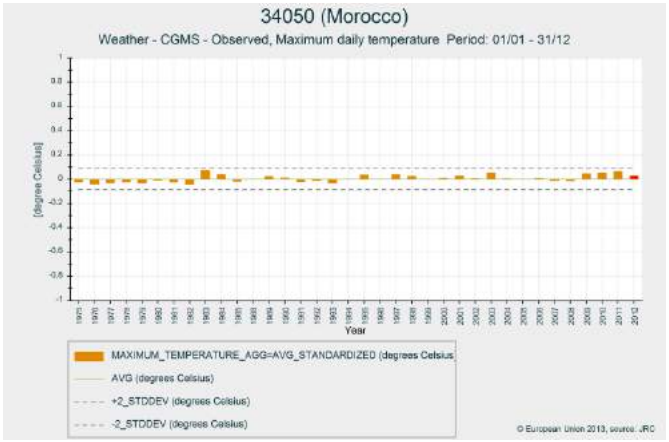


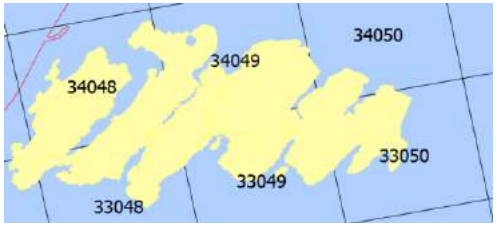
Maamora forest domain

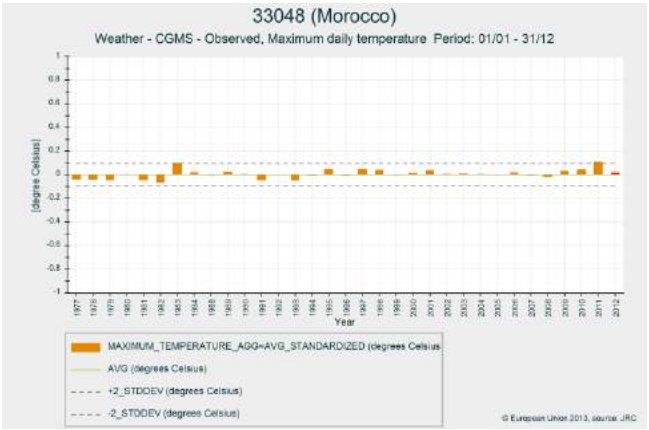
Standard T. MAX index

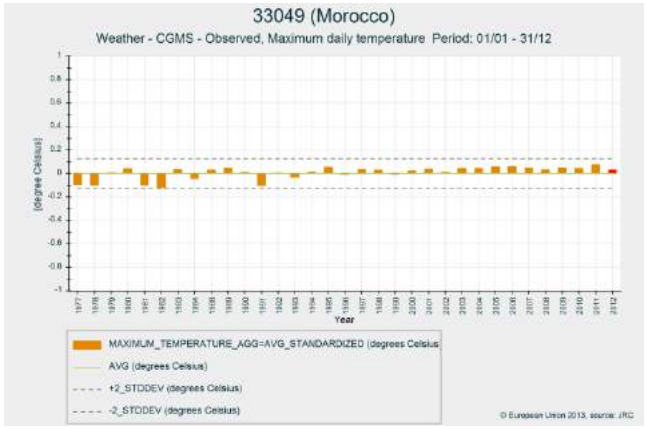


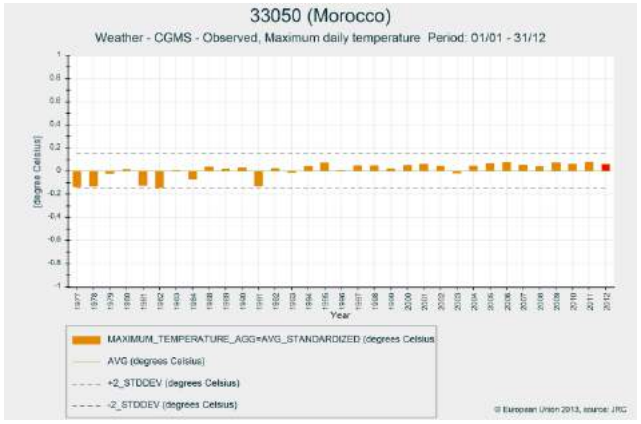






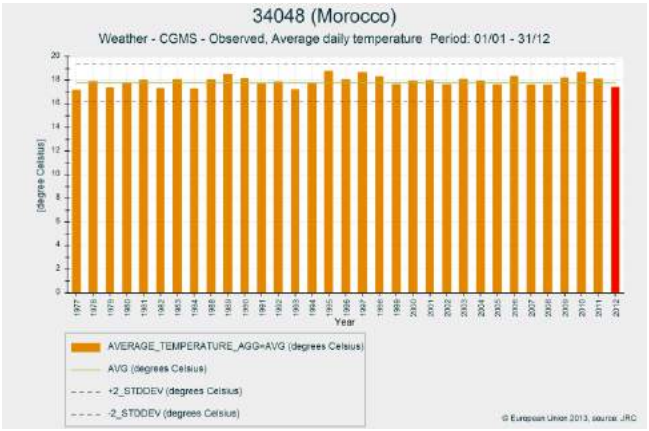


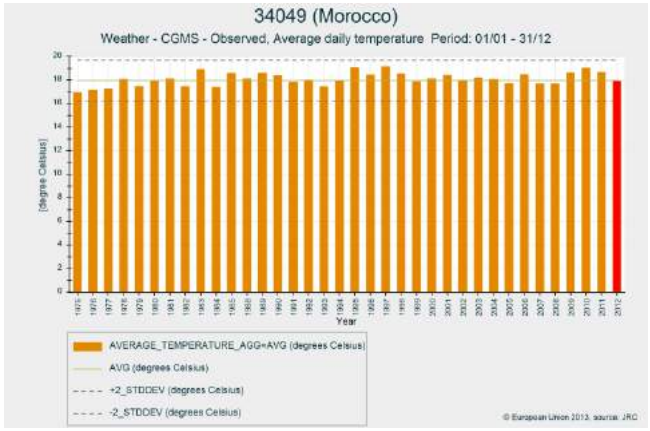


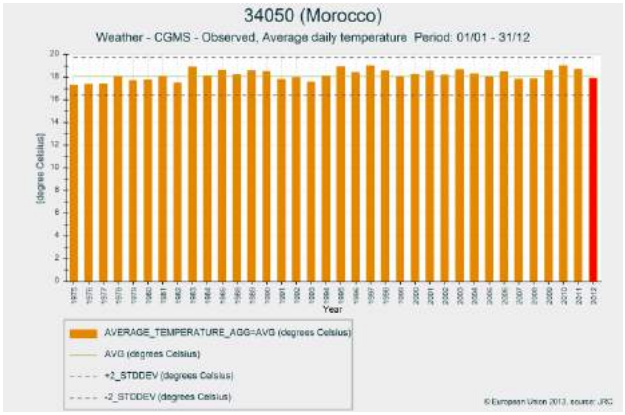


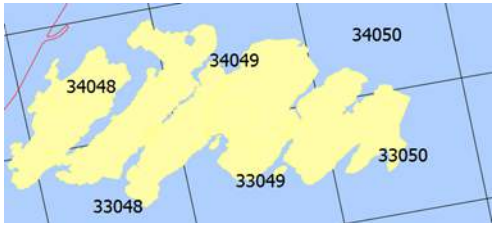
Maamora forest domain

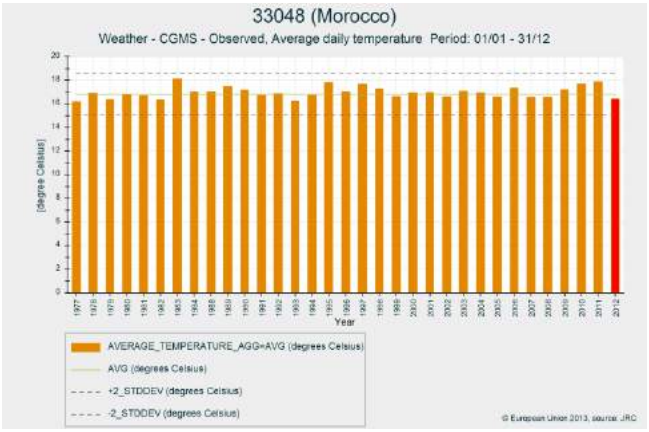
MINIMUM temperature (daily average)

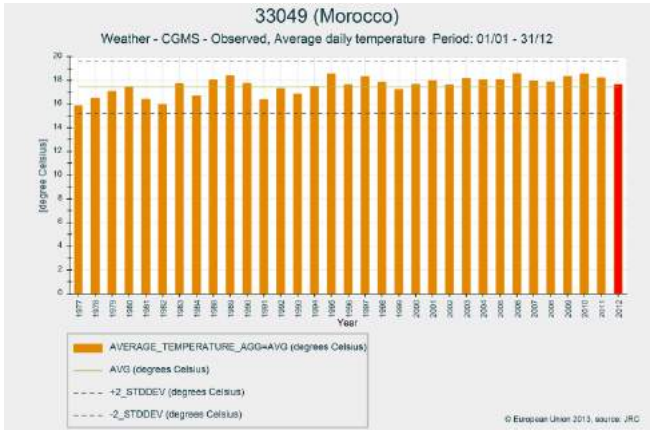


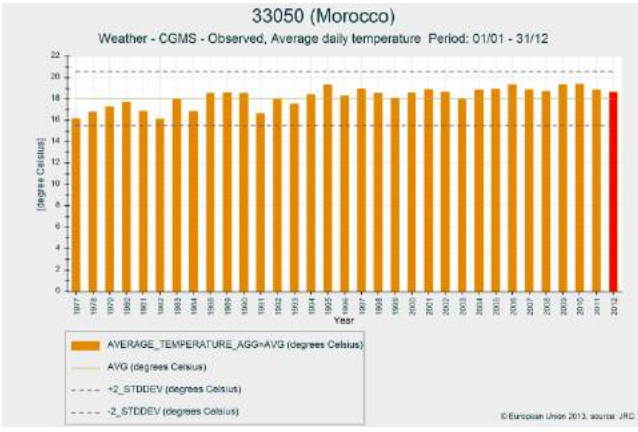






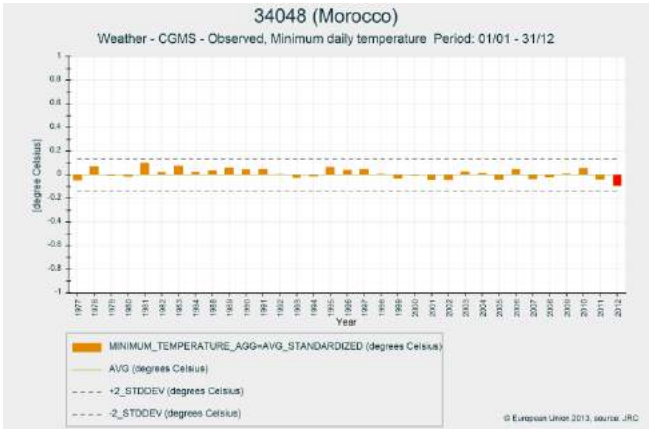


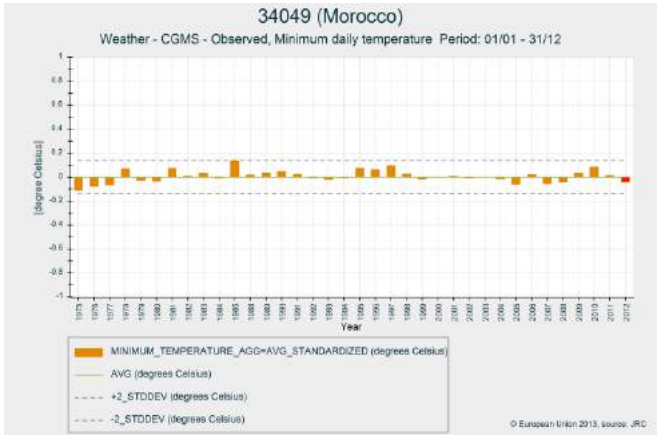


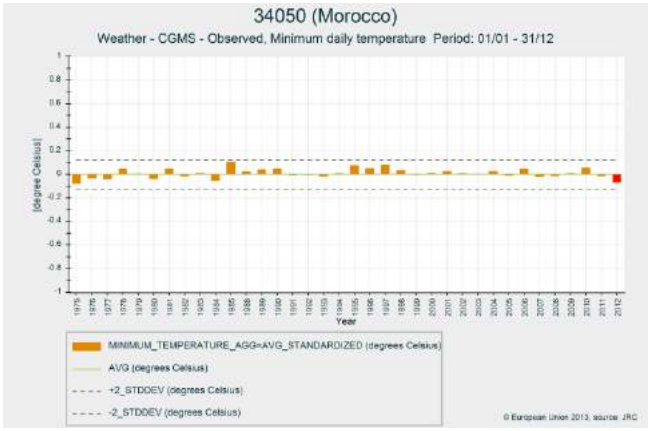


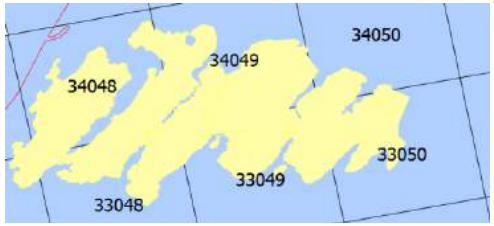
Maamora forest domain

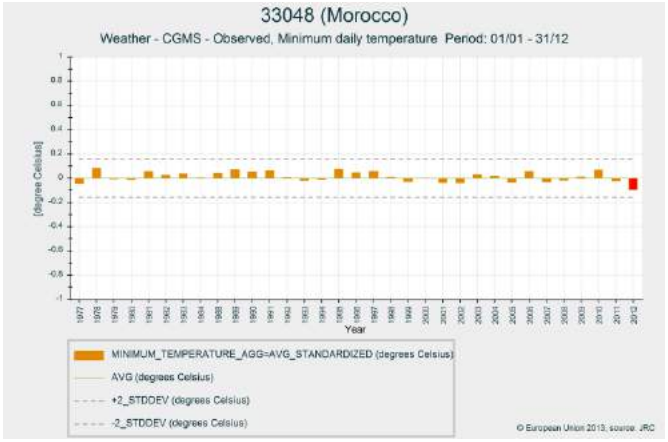
Standard T. MIN index

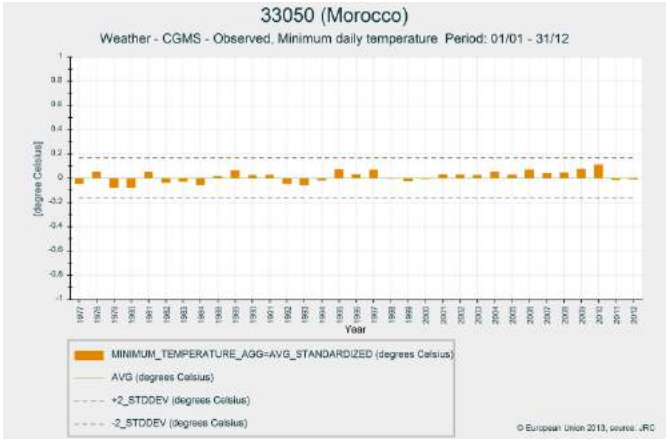






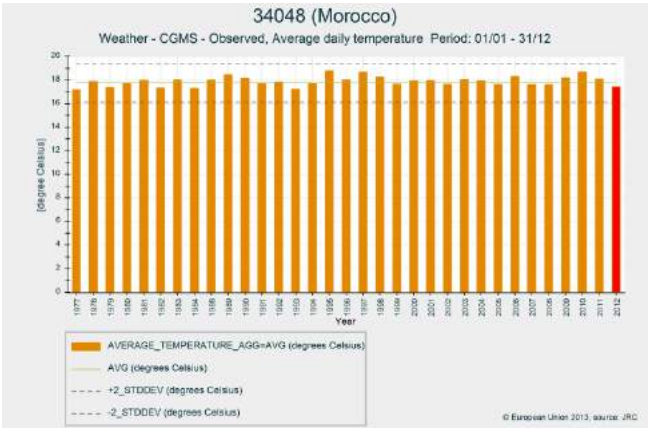


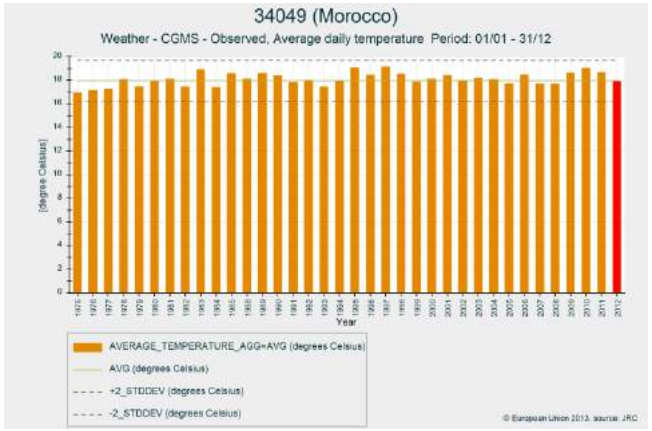


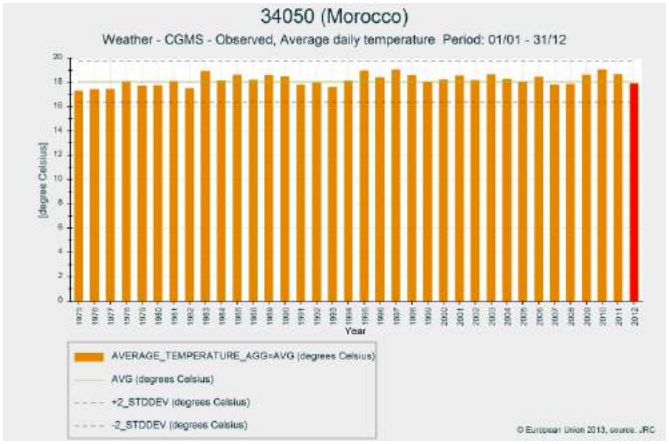


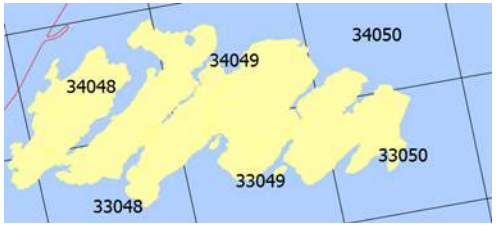
Maamora forest domain

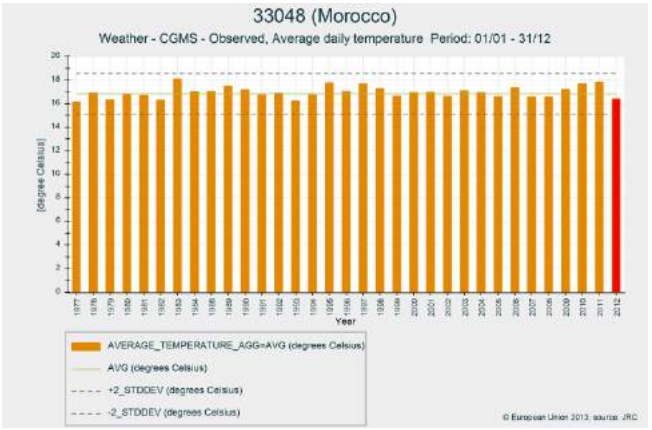
MEAN temperature (daily average)

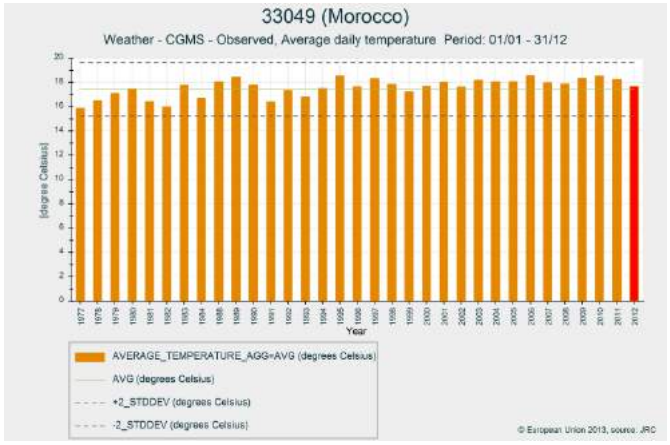


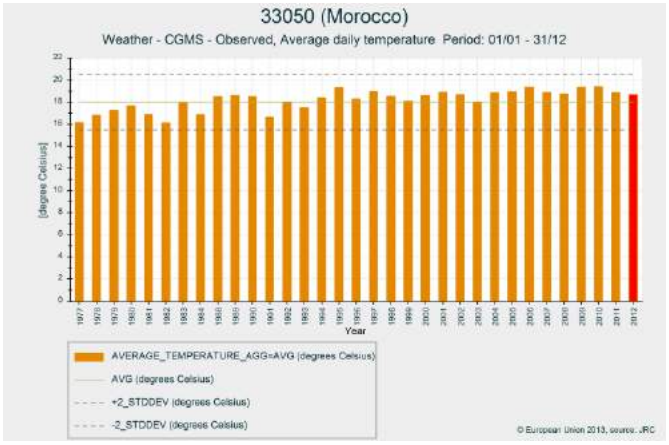




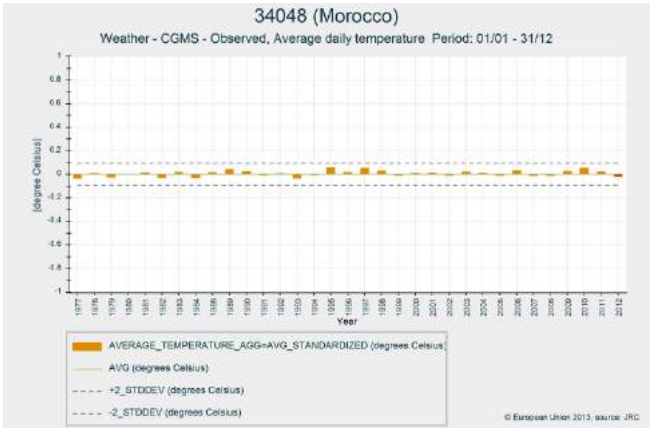


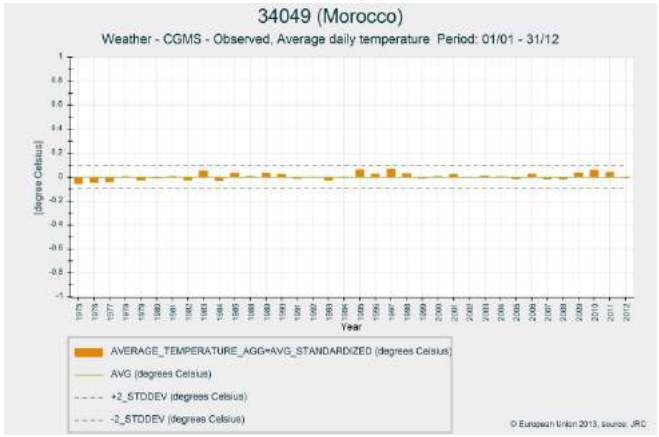


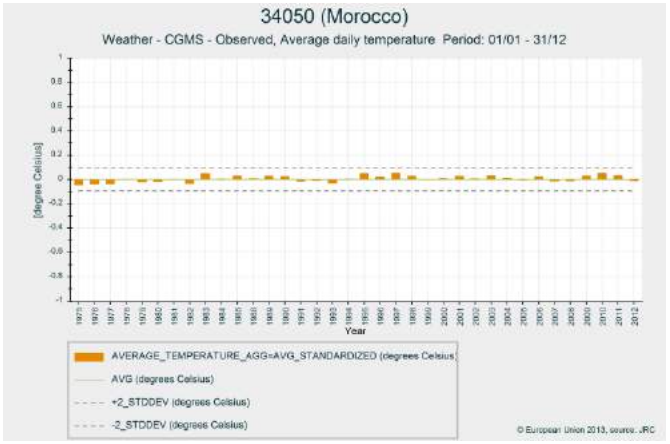


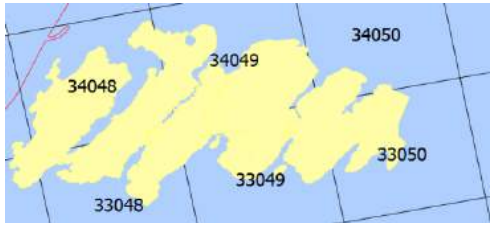


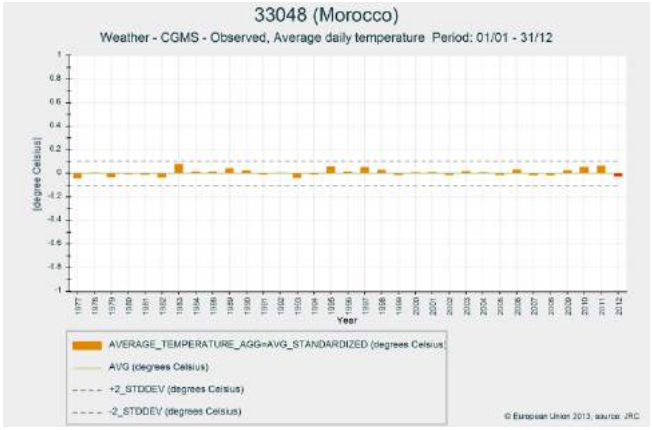
Maamora forest domain
Standard T. MEAN index

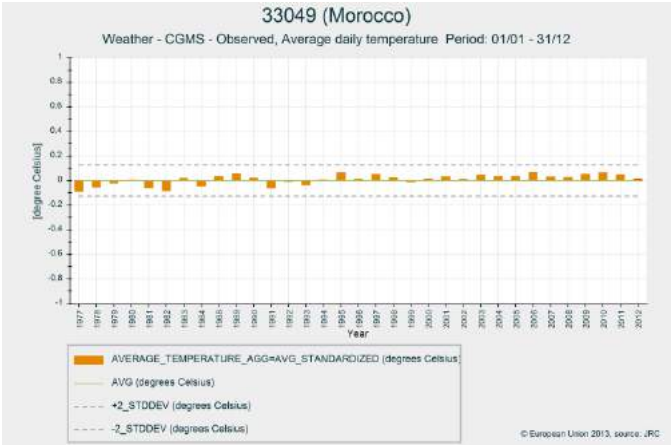


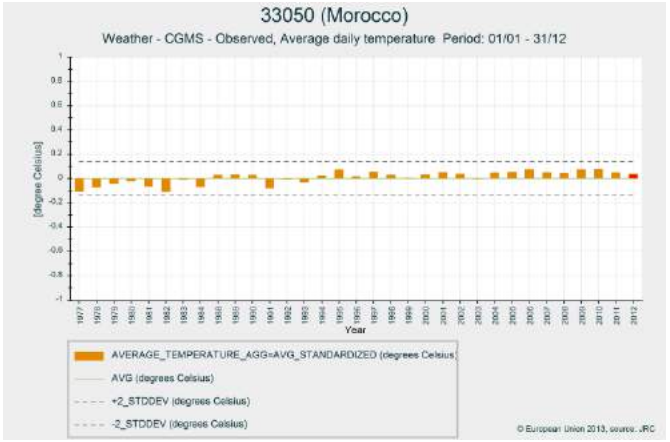












D Listings

D.1 Processing the NFI data base (bash script)

```
#!/bin/bash
ROOTDIR=/storage/hyper2/_Projects/NI079_FFEM/nfi/STRATES

for file in $(find . -name 'canton*.shp');do ogrinfo
    → $file $(basename $file .shp)|grep ESSENCE|awk -v
    → FS='=' '{print $2}';done|sort -u >
    → ${ROOTDIR}/essence.txt

echo "Found some inconsistencies in ESSENCE fields:
    → manually corrected using qgis: edit attribute
    → table "
echo "Todo: species mixtures should be resolved
    → (Eu.gr/ca1tj, Eu.gr/ca1tj/a, Eu.gr/ca2ta, ..."

echo "RgQs2fj should be Rg Qs2fj (missing space)"
echo "in file":
for file in $(find . -name 'canton*.shp');do
    → COUNT=$(ogrinfo $file $(basename $file .shp)
    → -where ESSENCE='RgQs2fj'|grep Count|awk -v
    → FS=':' '{print $2}'); if [ "${COUNT}" -gt 0
    → ];then echo "$file $COUNT";fi;done
#echo ${ROOTDIR}/cantonBII_strate.shp

echo "Qs 2fv should be Qs2fv (space should be
    → deleted)"
echo "in file":
for file in $(find . -name 'canton*.shp');do
    → FOUND=$(ogrinfo $file $(basename $file
    → .shp)|grep ESSENCE|grep "Qs 2fv");if [ -n
    → "${FOUND}" ];then echo $file;fi;done
#echo ${ROOTDIR}/cantonBIV_strate.shp

echo "P.p should be P.pa"
echo "in file":
for file in $(find . -name 'canton*.shp');do
    → FOUND=$(ogrinfo $file $(basename $file
    → .shp)|grep ESSENCE|grep "P.p[0-9]");if [ -n
    → "${FOUND}" ];then echo $file;fi;done
#echo ${ROOTDIR}/cantonAIV_strate.shp
#echo ${ROOTDIR}/cantonAII_strate.shp
#echo ${ROOTDIR}/cantonBV_strate.shp
#echo ${ROOTDIR}/cantonAVI_strate.shp
```

```
echo "Pc should be P.c"
echo "in file":
for file in $(find . -name 'canton*.shp');do
    → FOUND=$(ogrinfo $file $(basename $file
    → .shp)|grep ESSENCE|grep "Pc[0-9]");if [ -n
    → "${FOUND}" ];then echo $file;fi;done
#echo ${ROOTDIR}/cantonBV_strate.shp

echo "Eu.cla should be Eu.cl"
echo "in file":
for file in $(find . -name 'canton*.shp');do
    → FOUND=$(ogrinfo $file $(basename $file
    → .shp)|grep ESSENCE|grep "Eu.cla");if [ -n
    → "${FOUND}" ];then echo $file;fi;done
#echo ${ROOTDIR}/cantonDIV_strate.shp

echo "Rb Eu.cl/si1tj should be Rb Eu.cl1tj Eu.si1tj"
echo "in file":
for file in $(find . -name 'canton*_strate.shp');do
    → FOUND=$(ogrinfo $file $(basename $file
    → .shp)|grep ESSENCE|grep "Eu.cl/si1tj");if [ -n
    → "${FOUND}" ];then echo $file;fi;done
#echo ${ROOTDIR}/cantonEV_strate.shp

for file in $(find . -name 'canton*_strate.shp');do
    → FOUND=$(ogrinfo $file $(basename $file
    → .shp)|grep ESSENCE|grep "Rb Eu.go/ca1ta");if [
    → -n "${FOUND}" ];then echo $file;fi;done
#echo ${ROOTDIR}/cantonCII_strate.shp

echo "Echec should be echec"
echo "in file":
for file in $(find . -name 'canton*.shp');do
    → FOUND=$(ogrinfo $file $(basename $file
    → .shp)|grep ESSENCE|grep "Echec");if [ -n
    → "${FOUND}" ];then echo $file;fi;done
#echo ${ROOTDIR}/cantonAIV_strate.shp

echo "P.rad should be P.r"
echo "in file":
for file in $(find . -name 'canton*.shp');do
    → FOUND=$(ogrinfo $file $(basename $file
    → .shp)|grep ESSENCE|grep "P.rad");if [ -n
    → "${FOUND}" ];then echo $file;fi;done
#echo ${ROOTDIR}/cantonEI_strate.shp
```

```
55 echo "Qs3j/a should be Qs3fj/a ?"
   echo "in file":
57 for file in $(find . -name 'canton*.shp');do
      → FOUND=$(ogrinfo $file $(basename $file
      → .shp)|grep ESSENCE|grep "Qs3j");if [ -n
      → "${FOUND}" ];then echo $file;fi;done
   #echo ${ROOTDIR}/
59
   echo "Now corrected in new version of
      → /storage/hyper2/_Projects/NI079_FFEM/nfi/STRATES"
61
   for file in $(find . -name 'canton*.shp');do ogrinfo
      → $file $(basename $file .shp)|grep ESSENCE|awk -v
      → FS='=' '{print $2}';done|sort -u >
      → ${ROOTDIR}/essence_corrected.txt
63
   #different cases:
65 echo "process one essence"
   awk '{ if (NF==1) print }'
      → ${ROOTDIR}/essence_corrected.txt
67 #examples:
   #Qs1fa
69 #Qs1fa/v
   TMPDIR=${ROOTDIR}/tmp
71 mkdir -p ${TMPDIR}
   OUTPUTDIR=${ROOTDIR}/vito
73 mkdir -p ${OUTPUTDIR}
   awk '{ if (NF==1) print $1 }'
      → ${ROOTDIR}/essence_corrected.txt|while read
      → ORIGSTRING1;do
75     STRING1=$(echo $ORIGSTRING1|sed -e
      → 's/[jav]\/./m/g' -e 's/\/.//')
       echo "${STRING1}"|sed
      → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3
      → \4/'|while read TYPE1 DENSITY1 TRUNK1 STAGE1;do
77   if [ -n "${TYPE1}" -a -n "${DENSITY1}" -a -n
      → "${TRUNK1}" -a -n "${STAGE1}" ];then
         for file in $(find ${ROOTDIR} -name
      → 'canton[ABCDE]*_strate.shp');do
79     output=${TMPDIR}/$(basename $file
      → .shp)_${STRING1}.shp
       pkeditogr -i $file -o $output -an type1 -at
      → String -av ${TYPE1} -an type2 -at String -av NA
      → -an type3 -at String -av NA -an density1 -at
      → Integer -av ${DENSITY1} -an density2 -at Integer
```

```
    → -av 0 -an density3 -at Integer -av 0 -an plan1
    → -at Integer -av 0 -an plan2 -at Integer -av 0
    → -an plan3 -at Integer -av 0 -an trunk1 -at
    → String -av ${TRUNK1} -an trunk2 -at String -av
    → NA -an trunk3 -at String -av NA -an stage1 -at
    → String -av ${STAGE1} -an stage2 -at String -av
    → NA -an stage3 -at String -av NA -select ESSENCE
    → -like "$ORIGSTRING1" -st
81   COUNT=$(ogrinfo -so $output $(basename $output
    → .shp) |grep Count|awk -v FS=':' '{print $2}')
  if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
83     ogr2ogr -update -append
    → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
    → nfi_mamora
  else
85     ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
  fi
87 rm ${TMPDIR}/$(basename $output .shp).*
       done
89     echo "processed $ORIGSTRING1"
  else
91     echo "Warning: $ORIGSTRING1 not processed due
    → to missing information"
  fi
93   done
done
95
echo "process one essence in Regeneration (plan1=1)"
97 awk '{ if (NF==2&&$1=="Rg") print }'
    → ${ROOTDIR}/essence_corrected.txt
#examples:
99 #Rg A.mo1fj
TMPDIR=${ROOTDIR}/tmp
101 mkdir -p ${TMPDIR}
OUTPUTDIR=${ROOTDIR}/vito
103 mkdir -p ${OUTPUTDIR}
ORIGSTRING=
105 awk '{ if (NF==2&&$1=="Rg") print $2 }'
    → ${ROOTDIR}/essence_corrected.txt|while read
    → ORIGSTRING1;do
     STRING1=$(echo $ORIGSTRING1|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')
107
     echo "${STRING1}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3
    → \4/'|while read TYPE1 DENSITY1 TRUNK1 STAGE1;do
```

```
  if [ -n "${TYPE1}" -a -n "${DENSITY1}" -a -n
   → "${TRUNK1}" -a -n "${STAGE1}" ];then
     for file in $(find ${ROOTDIR} -name
   → 'canton[ABCDE]*_strate.shp');do
    output=${TMPDIR}/$(basename $file
   → .shp)_${STRING1}.shp
    pkeditogr -i $file -o $output -an type1 -at
   → String -av ${TYPE1} -an type2 -at String -av NA
   → -an type3 -at String -av NA -an density1 -at
   → Integer -av ${DENSITY1} -an density2 -at Integer
   → -av 0 -an density3 -at Integer -av 0 -an plan1
   → -at Integer -av 1 -an plan2 -at Integer -av 0
   → -an plan3 -at Integer -av 0 -an trunk1 -at
   → String -av ${TRUNK1} -an trunk2 -at String -av
   → NA -an trunk3 -at String -av NA -an stage1 -at
   → String -av ${STAGE1} -an stage2 -at String -av
   → NA -an stage3 -at String -av NA -select ESSENCE
   → -like "Rg ${ORIGSTRING1}" -st
    COUNT=$(ogrinfo -so $output $(basename $output
   → .shp) |grep Count|awk -v FS=':' '{print $2}')
    if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
        ogr2ogr -update -append
   → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
   → nfi_mamora
    else
        ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
    fi
    rm ${TMPDIR}/$(basename $output .shp).*
      done
      echo "processed $ORIGSTRING1"
  else
      echo "Warning: $ORIGSTRING1 not processed due
   → to missing information"
  fi
    done
done

echo "process one essence in Regeneration echec
   → (plan=1, stage=e)"
awk '{ if (NF==3&&$1=="Rg"&&$3=="echec") print }'
   → ${ROOTDIR}/essence_corrected.txt
#examples:
#Rg Qs1fj echec
#Rg Qs echec
TMPDIR=${ROOTDIR}/tmp
mkdir -p ${TMPDIR}
```

```
OUTPUTDIR=${ROOTDIR}/vito
mkdir -p ${OUTPUTDIR}
awk '{ if (NF==3&&$1=="Rg"&&$3=="echec") print $2 }'
    → ${ROOTDIR}/essence_corrected.txt|while read
    → ORIGSTRING1;do
     STRING1=$(echo $ORIGSTRING1|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')
     echo "${STRING1}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3
    → \4/'|while read TYPE1 DENSITY1 TRUNK1 STAGE1;do
   if [ -n "${TYPE1}" ];then
       for file in $(find ${ROOTDIR} -name
    → 'canton[ABCDE]*_strate.shp');do
     output=${TMPDIR}/$(basename $file
    → .shp)_${STRING1}.shp
     pkeditogr -i $file -o $output -an type1 -at
    → String -av ${TYPE1} -an type2 -at String -av NA
    → -an type3 -at String -av NA -an density1 -at
    → Integer -av 0 -an density2 -at Integer -av 0 -an
    → density3 -at Integer -av 0 -an plan1 -at Integer
    → -av 1 -an plan2 -at Integer -av 0 -an plan3 -at
    → Integer -av 0 -an trunk1 -at String -av NA -an
    → trunk2 -at String -av NA -an trunk3 -at String
    → -av NA -an stage1 -at String -av e -an stage2
    → -at String -av NA -an stage3 -at String -av NA
    → -select ESSENCE -like "Rg ${ORIGSTRING1} echec"
    → -st
     COUNT=$(ogrinfo -so $output $(basename $output
    → .shp) |grep Count|awk -v FS=':' '{print $2}')
     if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
         ogr2ogr -update -append
    → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
    → nfi_mamora
     else
         ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
     fi
     rm ${TMPDIR}/$(basename $output .shp).*
       done
       echo "processed $ORIGSTRING1"
   else
       echo "Warning: $ORIGSTRING1 not processed due
    → to missing information"
   fi
     done
done
```

```
159 echo "process one essence in Reboisement (plan1=2)"
    awk '{ if (NF==2&&$1=="Rb") print }'
       → ${ROOTDIR}/essence_corrected.txt
161 #examples:
    #Rb Eu.go3tj
163 #Rb Eu.go/ca2ta
    #Rb Eu.gr/ca1tj/a
165 #Rb Eu.gr/ca2ta
    TMPDIR=${ROOTDIR}/tmp
167 mkdir -p ${TMPDIR}
    OUTPUTDIR=${ROOTDIR}/vito
169 mkdir -p ${OUTPUTDIR}
    awk '{ if (NF==2&&$1=="Rb") print $2 }'
       → ${ROOTDIR}/essence_corrected.txt|while read
       → ORIGSTRING1;do
171     STRING1=$(echo $ORIGSTRING1|sed -e
       → 's/[jav]\/./m/g' -e 's/\/.//')

173     echo "${STRING1}"|sed
       → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3
       → \4/'|while read TYPE1 DENSITY1 TRUNK1 STAGE1;do
      if [ -n "${TYPE1}" -a -n "${DENSITY1}" -a -n
       → "${TRUNK1}" -a -n "${STAGE1}" ];then
175       for file in $(find ${ROOTDIR} -name
       → 'canton[ABCDE]*_strate.shp');do
       output=${TMPDIR}/$(basename $file
       → .shp)_${STRING1}.shp
177     pkeditogr -i $file -o $output -an type1 -at
       → String -av ${TYPE1} -an type2 -at String -av NA
       → -an type3 -at String -av NA -an density1 -at
       → Integer -av ${DENSITY1} -an density2 -at Integer
       → -av 0 -an density3 -at Integer -av 0 -an plan1
       → -at Integer -av 2 -an plan2 -at Integer -av 0
       → -an plan3 -at Integer -av 0 -an trunk1 -at
       → String -av ${TRUNK1} -an trunk2 -at String -av
       → NA -an trunk3 -at String -av NA -an stage1 -at
       → String -av ${STAGE1} -an stage2 -at String -av
       → NA -an stage3 -at String -av NA -select ESSENCE
       → -like "Rb ${ORIGSTRING1}" -st
       COUNT=$(ogrinfo -so $output $(basename $output
       → .shp) |grep Count|awk -v FS=':' '{print $2}')
179   if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
        ogr2ogr -update -append
       → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
       → nfi_mamora
181   else
```

```
        ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
183   fi
    rm ${TMPDIR}/$(basename $output .shp).*
185       done
          echo "processed $ORIGSTRING1"
187   else
          echo "Warning: $ORIGSTRING1 not processed due
     → to missing information"
189   fi
        done
191 done

193 echo "process one essence in Reboisement echec
     → (plan1=2,stage=e)"
    awk '{ if (NF==3&&$1=="Rb"&&$3=="echec") print }'
     → ${ROOTDIR}/essence_corrected.txt
195 #examples:
    #Rb P.pa echec
197 TMPDIR=${ROOTDIR}/tmp
    mkdir -p ${TMPDIR}
199 OUTPUTDIR=${ROOTDIR}/vito
    mkdir -p ${OUTPUTDIR}
201 awk '{ if (NF==3&&$1=="Rb"&&$3=="echec") print $2 }'
     → ${ROOTDIR}/essence_corrected.txt|while read
     → ORIGSTRING1;do
      STRING1=$(echo $ORIGSTRING1|sed -e
     → 's/[jav]\/./m/g' -e 's/\/.//')
203
      echo "${STRING1}"|sed
     → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3
     → \4/'|while read TYPE1 DENSITY1 TRUNK1 STAGE1;do
205   if [ -n "${TYPE1}" ];then
          for file in $(find ${ROOTDIR} -name
     → 'canton[ABCDE]*_strate.shp');do
207     output=${TMPDIR}/$(basename $file
     → .shp)_${STRING1}.shp
        pkeditogr -i $file -o $output -an type1 -at
     → String -av ${TYPE1} -an type2 -at String -av NA
     → -an type3 -at String -av NA -an density1 -at
     → Integer -av 0 -an density2 -at Integer -av 0 -an
     → density3 -at Integer -av 0 -an plan1 -at Integer
     → -av 2 -an plan2 -at Integer -av 0 -an plan3 -at
     → Integer -av 0 -an trunk1 -at String -av NA -an
     → trunk2 -at String -av NA -an trunk3 -at String
     → -av NA -an stage1 -at String -av e -an stage2
     → -at String -av NA -an stage3 -at String -av NA
```

```
    → -select ESSENCE -like "Rb ${ORIGSTRING1} echec"
    → -st
209     COUNT=$(ogrinfo -so $output $(basename $output
    → .shp) |grep Count|awk -v FS=':' '{print $2}')
    if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
211         ogr2ogr -update -append
    → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
    → nfi_mamora
    else
213         ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
    fi
215 rm ${TMPDIR}/$(basename $output .shp).*
        done
217     echo "processed $ORIGSTRING1"
    else
219         echo "Warning: $ORIGSTRING1 not processed due
    → to missing information"
    fi
221   done
done
223
echo "process two essences"
225 awk '{ if (NF==2&&$1!="Rg"&&$1!="Rb") print }'
    → ${ROOTDIR}/essence_corrected.txt
#examples:
227 #Qs1fa/v Pcm4fa
#Qs2fa Qs3fv
229 TMPDIR=${ROOTDIR}/tmp
mkdir -p ${TMPDIR}
231 OUTPUTDIR=${ROOTDIR}/vito
mkdir -p ${OUTPUTDIR}
233 awk '{ if (NF==2&&$1!="Rg"&&$1!="Rb") print $1,$2}'
    → ${ROOTDIR}/essence_corrected.txt|while read
    → ORIGSTRING1 ORIGSTRING2;do
    STRING1=$(echo $ORIGSTRING1|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')
235 STRING2=$(echo $ORIGSTRING2|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')

237   echo "${STRING1} ${STRING2}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)
    → \(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3 \4
    → \5 \6 \7 \8/'|while read TYPE1 DENSITY1 TRUNK1
    → STAGE1 TYPE2 DENSITY2 TRUNK2 STAGE2;do
    if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n
    → "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}"
```

```
    → -a -n "${DENSITY2}"  -a -n "${TRUNK2}" -a -n
    → "${STAGE2}" ];then
239     for file in $(find ${ROOTDIR} -name
    → 'canton[ABCDE]*_strate.shp');do
     output=${TMPDIR}/$(basename $file
    → .shp)_${STRING1}_${STRING2}.shp
241     pkeditogr -i $file -o $output -an type1 -at
    → String -av ${TYPE1} -an type2 -at String -av
    → ${TYPE2} -an type3 -at String -av NA -an
    → density1 -at Integer -av ${DENSITY1} -an
    → density2 -at Integer -av ${DENSITY2} -an
    → density3 -at Integer -av 0 -an plan1 -at Integer
    → -av 0 -an plan2 -at Integer -av 0 -an plan3 -at
    → Integer -av 0 -an trunk1 -at String -av
    → ${TRUNK1} -an trunk2 -at String -av ${TRUNK2}
    → -an trunk3 -at String -av NA -an stage1 -at
    → String -av ${STAGE1} -an stage2 -at String -av
    → ${STAGE2} -an stage3 -at String -av NA -select
    → ESSENCE -like "${ORIGSTRING1} ${ORIGSTRING2}" -st
     COUNT=$(ogrinfo -so $output $(basename $output
    → .shp) |grep Count|awk -v FS=':' '{print $2}')
243   if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
        ogr2ogr -update -append
     → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
     → nfi_mamora
245   else
        ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
247   fi
      rm ${TMPDIR}/$(basename $output .shp).*
249       done
          echo "processed $ORIGSTRING1 $ORIGSTRING2"
251   else
          echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
      → processed due to missing information"
253   fi
        done
255 done

257 echo "process two essence in Regeneration (plan1=1)"
    awk '{ if (NF==3&&$1=="Rg"&&$3!="echec") print }'
       → ${ROOTDIR}/essence_corrected.txt
259 #examples:
    #Rg Qs1fj/a Qs3fv
261 #Rg Qs1fj Qs3fv/a
    TMPDIR=${ROOTDIR}/tmp
263 mkdir -p ${TMPDIR}
```

```
OUTPUTDIR=${ROOTDIR}/vito
mkdir -p ${OUTPUTDIR}
awk '{ if (NF==3&&$1=="Rg"&&$3!="echec") print $2,$3 }' ${ROOTDIR}/essence_corrected.txt|while read ORIGSTRING1 ORIGSTRING2;do
 STRING1=$(echo $ORIGSTRING1|sed -e 's/[jav]\/./m/g' -e 's/\/.//')
 STRING2=$(echo $ORIGSTRING2|sed -e 's/[jav]\/./m/g' -e 's/\/.//')

  echo "${STRING1} ${STRING2}"|sed 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3 \4 \5 \6 \7 \8/'|while read TYPE1 DENSITY1 TRUNK1 STAGE1 TYPE2 DENSITY2 TRUNK2 STAGE2;do
 if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}" -a -n "${DENSITY2}"  -a -n "${TRUNK2}" -a -n "${STAGE2}" ];then
    for file in $(find ${ROOTDIR} -name 'canton[ABCDE]*_strate.shp');do
  output=${TMPDIR}/$(basename $file .shp)_${STRING1}_${STRING2}.shp
  pkeditogr -i $file -o $output -an type1 -at String -av ${TYPE1} -an type2 -at String -av ${TYPE2} -an type3 -at String -av NA -an density1 -at Integer -av ${DENSITY1} -an density2 -at Integer -av ${DENSITY2} -an density3 -at Integer -av 0 -an plan1 -at Integer -av 1 -an plan2 -at Integer -av 0 -an plan3 -at Integer -av 0 -an trunk1 -at String -av ${TRUNK1} -an trunk2 -at String -av ${TRUNK2} -an trunk3 -at String -av NA -an stage1 -at String -av ${STAGE1} -an stage2 -at String -av ${STAGE2} -an stage3 -at String -av NA -select ESSENCE -like "Rg ${ORIGSTRING1} ${ORIGSTRING2}" -st
  COUNT=$(ogrinfo -so $output $(basename $output .shp) |grep Count|awk -v FS=':' '{print $2}')
 if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
     ogr2ogr -update -append ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln nfi_mamora
 else
     ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
 fi
```

```
281   rm ${TMPDIR}/$(basename $output .shp).*
          done
283       echo "processed $ORIGSTRING1 $ORIGSTRING2"
      else
285       echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
       → processed due to missing information"
      fi
287     done
  done
289
  echo "process two essence in Regeneration (plan2=1)"
291 awk '{ if (NF==3&&$2=="Rg") print }'
      → ${ROOTDIR}/essence_corrected.txt
  #examples:
293 #Qs3fv Rg Qs1fj
  TMPDIR=${ROOTDIR}/tmp
295 mkdir -p ${TMPDIR}
  OUTPUTDIR=${ROOTDIR}/vito
297 mkdir -p ${OUTPUTDIR}
  awk '{ if (NF==3&&$2=="Rg") print $1,$3 }'
      → ${ROOTDIR}/essence_corrected.txt|while read
      → ORIGSTRING1 ORIGSTRING2;do
299   STRING1=$(echo $ORIGSTRING1|sed -e
      → 's/[jav]\/./m/g' -e 's/\/.//')
      STRING2=$(echo $ORIGSTRING2|sed -e
      → 's/[jav]\/./m/g' -e 's/\/.//')
301
       echo "${STRING1} ${STRING2}"|sed
      → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)
      → \(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3 \4
      → \5 \6 \7 \8/'|while read TYPE1 DENSITY1 TRUNK1
      → STAGE1 TYPE2 DENSITY2 TRUNK2 STAGE2;do
303   if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n
       → "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}"
       → -a -n "${DENSITY2}"  -a -n "${TRUNK2}" -a -n
       → "${STAGE2}" ];then
          for file in $(find ${ROOTDIR} -name
       → 'canton[ABCDE]*_strate.shp');do
305    output=${TMPDIR}/$(basename $file
       → .shp)_${STRING1}_${STRING2}.shp
       pkeditogr -i $file -o $output -an type1 -at
       → String -av ${TYPE1} -an type2 -at String -av
       → ${TYPE2} -an type3 -at String -av NA -an
       → density1 -at Integer -av ${DENSITY1} -an
       → density2 -at Integer -av ${DENSITY2} -an
       → density3 -at Integer -av 0 -an plan1 -at Integer
```

```
    → -av 0 -an plan2 -at Integer -av 1 -an plan3 -at
    → Integer -av 0 -an trunk1 -at String -av
    → ${TRUNK1} -an trunk2 -at String -av ${TRUNK2}
    → -an trunk3 -at String -av NA -an stage1 -at
    → String -av ${STAGE1} -an stage2 -at String -av
    → ${STAGE2} -an stage3 -at String -av NA -select
    → ESSENCE -like "${ORIGSTRING1} Rg ${ORIGSTRING2}"
    → -st
307  COUNT=$(ogrinfo -so $output $(basename $output
    → .shp) |grep Count|awk -v FS=':' '{print $2}')
   if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
309      ogr2ogr -update -append
    → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
    → nfi_mamora
   else
311      ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
   fi
313  rm ${TMPDIR}/$(basename $output .shp).*
      done
315      echo "processed $ORIGSTRING1 $ORIGSTRING2"
   else
317      echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
    → processed due to missing information"
   fi
319    done
done
321
echo "process two essence in Reboisement (plan1=2)"
323 awk '{ if (NF==3&&$1=="Rb"&&$3!="echec") print }'
    → ${ROOTDIR}/essence_corrected.txt
#examples:
325 #Rb Eu.ca1fa Eu.si1fa
#Rb P.pa1fj/a Qs3fv
327 #Rb P.pa3fj Es
TMPDIR=${ROOTDIR}/tmp
329 mkdir -p ${TMPDIR}
OUTPUTDIR=${ROOTDIR}/vito
331 mkdir -p ${OUTPUTDIR}
awk '{ if (NF==3&&$1=="Rb"&&$3!="echec") print $2,$3
    → }' ${ROOTDIR}/essence_corrected.txt|while read
    → ORIGSTRING1 ORIGSTRING2;do
333  STRING1=$(echo $ORIGSTRING1|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')
    STRING2=$(echo $ORIGSTRING2|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')
335
```

```
      echo "${STRING1} ${STRING2}"|sed
     → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)
     → \(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3 \4
     → \5 \6 \7 \8/'|while read TYPE1 DENSITY1 TRUNK1
     → STAGE1 TYPE2 DENSITY2 TRUNK2 STAGE2;do
337   if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n
     → "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}"
     → -a -n "${DENSITY2}"  -a -n "${TRUNK2}" -a -n
     → "${STAGE2}" ];then
        for file in $(find ${ROOTDIR} -name
     → 'canton[ABCDE]*_strate.shp');do
339     output=${TMPDIR}/$(basename $file
     → .shp)_${STRING1}_${STRING2}.shp
        pkeditogr -i $file -o $output -an type1 -at
     → String -av ${TYPE1} -an type2 -at String -av
     → ${TYPE2} -an type3 -at String -av NA -an
     → density1 -at Integer -av ${DENSITY1} -an
     → density2 -at Integer -av ${DENSITY2} -an
     → density3 -at Integer -av 0 -an plan1 -at Integer
     → -av 2 -an plan2 -at Integer -av 0 -an plan3 -at
     → Integer -av 0 -an trunk1 -at String -av
     → ${TRUNK1} -an trunk2 -at String -av ${TRUNK2}
     → -an trunk3 -at String -av NA -an stage1 -at
     → String -av ${STAGE1} -an stage2 -at String -av
     → ${STAGE2} -an stage3 -at String -av NA -select
     → ESSENCE -like "Rb ${ORIGSTRING1} ${ORIGSTRING2}"
     → -st
341     COUNT=$(ogrinfo -so $output $(basename $output
     → .shp) |grep Count|awk -v FS=':' '{print $2}')
      if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
343       ogr2ogr -update -append
     → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
     → nfi_mamora
      else
345       ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
      fi
347   rm ${TMPDIR}/$(basename $output .shp).*
        done
349     echo "processed $ORIGSTRING1 $ORIGSTRING2"
      else
351       echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
     → processed due to missing information"
      fi
353     done
  done
355
```

```
   echo "process two essence in Reboisement (plan2=2)"
357 awk '{ if (NF==3&&$2=="Rb") print }'
      → ${ROOTDIR}/essence_corrected.txt
   #examples:
359 #Qs2fv Rb A.mo2fj
   TMPDIR=${ROOTDIR}/tmp
361 mkdir -p ${TMPDIR}
   OUTPUTDIR=${ROOTDIR}/vito
363 mkdir -p ${OUTPUTDIR}
   awk '{ if (NF==3&&$2=="Rb") print $1,$3 }'
      → ${ROOTDIR}/essence_corrected.txt|while read
      → ORIGSTRING1 ORIGSTRING2;do
365    STRING1=$(echo $ORIGSTRING1|sed -e
      → 's/[jav]\/./m/g' -e 's/\/.//')
      STRING2=$(echo $ORIGSTRING2|sed -e
      → 's/[jav]\/./m/g' -e 's/\/.//')
367
       echo "${STRING1} ${STRING2}"|sed
      → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)
      → \(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3 \4
      → \5 \6 \7 \8/'|while read TYPE1 DENSITY1 TRUNK1
      → STAGE1 TYPE2 DENSITY2 TRUNK2 STAGE2;do
369   if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n
      → "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}"
      → -a -n "${DENSITY2}"  -a -n "${TRUNK2}" -a -n
      → "${STAGE2}" ];then
         for file in $(find ${ROOTDIR} -name
      → 'canton[ABCDE]*_strate.shp');do
371     output=${TMPDIR}/$(basename $file
      → .shp)_${STRING1}_${STRING2}.shp
       pkeditogr -i $file -o $output -an type1 -at
      → String -av ${TYPE1} -an type2 -at String -av
      → ${TYPE2} -an type3 -at String -av NA -an
      → density1 -at Integer -av ${DENSITY1} -an
      → density2 -at Integer -av ${DENSITY2} -an
      → density3 -at Integer -av 0 -an plan1 -at Integer
      → -av 0 -an plan2 -at Integer -av 2 -an plan3 -at
      → Integer -av 0 -an trunk1 -at String -av
      → ${TRUNK1} -an trunk2 -at String -av ${TRUNK2}
      → -an trunk3 -at String -av NA -an stage1 -at
      → String -av ${STAGE1} -an stage2 -at String -av
      → ${STAGE2} -an stage3 -at String -av NA -select
      → ESSENCE -like "${ORIGSTRING1} Rb ${ORIGSTRING2}"
      → -st
373     COUNT=$(ogrinfo -so $output $(basename $output
      → .shp) |grep Count|awk -v FS=':' '{print $2}')
```

```
    if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
375     ogr2ogr -update -append
     → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
     → nfi_mamora
    else
377     ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
    fi
379 rm ${TMPDIR}/$(basename $output .shp).*
        done
381     echo "processed $ORIGSTRING1 $ORIGSTRING2"
    else
383     echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
     → processed due to missing information"
    fi
385   done
done
387
echo "process two essence in Regeneration"
389 awk '{ if (NF==4&&$1=="Rg"&&$3=="Rg") print }'
    → ${ROOTDIR}/essence_corrected.txt
#examples:
391 #(empty)

393 echo "process two essence in Reboisement
    → (plan1=2,plan2=2)"
awk '{ if (NF==4&&$1=="Rb"&&$3=="Rb") print }'
    → ${ROOTDIR}/essence_corrected.txt
395 OUTPUTDIR=${ROOTDIR}/vito
mkdir -p ${OUTPUTDIR}
397 TMPDIR=${ROOTDIR}/tmp
mkdir -p ${TMPDIR}
399 #examples:
#Rb A.cy2fa Rb Eu.ca3fa
401 awk '{ if (NF==4&&$1=="Rb"&&$3=="Rb") print $2,$4 }'
    → ${ROOTDIR}/essence_corrected.txt|while read
    → ORIGSTRING1 ORIGSTRING2;do
    STRING1=$(echo $ORIGSTRING1|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')
403 STRING2=$(echo $ORIGSTRING2|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')

405  echo "${STRING1} ${STRING2}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)
    → \(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3 \4
    → \5 \6 \7 \8/'|while read TYPE1 DENSITY1 TRUNK1
    → STAGE1 TYPE2 DENSITY2 TRUNK2 STAGE2;do
```

```
    if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n
     → "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}"
     → -a -n "${DENSITY2}"  -a -n "${TRUNK2}" -a -n
     → "${STAGE2}" ];then
407     for file in $(find ${ROOTDIR} -name
     → 'canton[ABCDE]*_strate.shp');do
      output=${TMPDIR}/$(basename $file
     → .shp)_${STRING1}_${STRING2}.shp
409    pkeditogr -i $file -o $output -an type1 -at
     → String -av ${TYPE1} -an type2 -at String -av
     → ${TYPE2} -an type3 -at String -av NA -an
     → density1 -at Integer -av ${DENSITY1} -an
     → density2 -at Integer -av ${DENSITY2} -an
     → density3 -at Integer -av 0 -an plan1 -at Integer
     → -av 2 -an plan2 -at Integer -av 2 -an plan3 -at
     → Integer -av 0 -an trunk1 -at String -av
     → ${TRUNK1} -an trunk2 -at String -av ${TRUNK2}
     → -an trunk3 -at String -av NA -an stage1 -at
     → String -av ${STAGE1} -an stage2 -at String -av
     → ${STAGE2} -an stage3 -at String -av NA -select
     → ESSENCE -like "Rb ${ORIGSTRING1} Rb
     → ${ORIGSTRING2}" -st
      COUNT=$(ogrinfo -so $output $(basename $output
     → .shp) |grep Count|awk -v FS=':' '{print $2}')
411   if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
        ogr2ogr -update -append
     → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
     → nfi_mamora
413   else
        ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
415   fi
    rm ${TMPDIR}/$(basename $output .shp).*
417     done
        echo "processed $ORIGSTRING1 $ORIGSTRING2"
419   else
        echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
     → processed due to missing information"
421   fi
      done
423 done

425 echo "process two essences in Regeneration and
     → Reboisement (plan1=1,plan2=2)"
   awk '{ if (NF==4&&$1=="Rg"&&$3=="Rb") print }'
     → ${ROOTDIR}/essence_corrected.txt
427 TMPDIR=${ROOTDIR}/tmp
```

```
mkdir -p ${TMPDIR}
OUTPUTDIR=${ROOTDIR}/vito
mkdir -p ${OUTPUTDIR}
echo "process two essences in Regeneration and Reboisement (plan1=1,plan2=2)"
awk '{ if (NF==4&&$1=="Rg"&&$3=="Rb") print }' ${ROOTDIR}/essence_corrected.txt
#examples:
#Rg Qs2fj Rb P.pa3fj
awk '{ if (NF==4&&$1=="Rg"&&$3=="Rb") print $2,$4 }' ${ROOTDIR}/essence_corrected.txt|while read ORIGSTRING1 ORIGSTRING2;do
STRING1=$(echo $ORIGSTRING1|sed -e 's/[jav]\/./m/g' -e 's/\/.//')
STRING2=$(echo $ORIGSTRING2|sed -e 's/[jav]\/./m/g' -e 's/\/.//')

 echo "${STRING1} ${STRING2}"|sed 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1 \2 \3 \4 \5 \6 \7 \8/'|while read TYPE1 DENSITY1 TRUNK1 STAGE1 TYPE2 DENSITY2 TRUNK2 STAGE2;do
if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}" -a -n "${DENSITY2}"  -a -n "${TRUNK2}" -a -n "${STAGE2}" ];then
   for file in $(find ${ROOTDIR} -name 'canton[ABCDE]*_strate.shp');do
 output=${TMPDIR}/$(basename $file .shp)_${STRING1}_${STRING2}.shp
 pkeditogr -i $file -o $output -an type1 -at String -av ${TYPE1} -an type2 -at String -av ${TYPE2} -an type3 -at String -av NA -an density1 -at Integer -av ${DENSITY1} -an density2 -at Integer -av ${DENSITY2} -an density3 -at Integer -av 0 -an plan1 -at Integer -av 1 -an plan2 -at Integer -av 2 -an plan3 -at Integer -av 0 -an trunk1 -at String -av ${TRUNK1} -an trunk2 -at String -av ${TRUNK2} -an trunk3 -at String -av NA -an stage1 -at String -av ${STAGE1} -an stage2 -at String -av ${STAGE2} -an stage3 -at String -av NA -select ESSENCE -like "Rg ${ORIGSTRING1} Rb ${ORIGSTRING2}" -st
 COUNT=$(ogrinfo -so $output $(basename $output .shp) |grep Count|awk -v FS=':' '{print $2}')
```

```
445   if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
          ogr2ogr -update -append
      → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
      → nfi_mamora
447   else
          ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
449   fi
      rm ${TMPDIR}/$(basename $output .shp).*
451       done
          echo "processed $ORIGSTRING1 $ORIGSTRING2"
453   else
          echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
      → processed due to missing information"
455   fi
        done
457 done

459 echo "process two essence in Regeneration echec
      → (plan2=1, stage2=e)"
    awk '{ if (NF==4&&$2=="Rg"&&$4=="echec") print }'
      → ${ROOTDIR}/essence_corrected.txt
461 #examples:
    #Qs3fa/v Rg Qs echec
463 TMPDIR=${ROOTDIR}/tmp
    mkdir -p ${TMPDIR}
465 OUTPUTDIR=${ROOTDIR}/vito
    mkdir -p ${OUTPUTDIR}
467 awk '{ if (NF==4&&$2=="Rg"&&$4=="echec") print $1,$3
      → }' ${ROOTDIR}/essence_corrected.txt|while read
      → ORIGSTRING1 ORIGSTRING2;do
      STRING1=$(echo $ORIGSTRING1|sed -e
      → 's/[jav]\/./m/g' -e 's/\/.//')
469   STRING2=$(echo $ORIGSTRING2|sed -e
      → 's/[jav]\/./m/g' -e 's/\/.//')

471    echo "${STRING1} ${STRING2}"|sed
      → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\) \(.*\)/\1
      → \2 \3 \4 \5/'|while read TYPE1 DENSITY1 TRUNK1
      → STAGE1 TYPE2;do
     if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n
      → "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}"
      → ];then
473      for file in $(find ${ROOTDIR} -name
      → 'canton[ABCDE]*_strate.shp');do
       output=${TMPDIR}/$(basename $file
      → .shp)_${STRING1}_${STRING2}.shp
```

```
475     pkeditogr -i $file -o $output -an type1 -at
      → String -av ${TYPE1} -an type2 -at String -av
      → ${TYPE2} -an type3 -at String -av NA -an
      → density1 -at Integer -av ${DENSITY1} -an
      → density2 -at Integer -av 0 -an density3 -at
      → Integer -av 0 -an plan1 -at Integer -av 0 -an
      → plan2 -at Integer -av 1 -an plan3 -at Integer
      → -av 0 -an trunk1 -at String -av ${TRUNK1} -an
      → trunk2 -at String -av NA -an trunk3 -at String
      → -av NA -an stage1 -at String -av ${STAGE1} -an
      → stage2 -at String -av e -an stage3 -at String
      → -av NA -select ESSENCE -like "${ORIGSTRING1} Rg
      → $ORIGSTRING2 echec" -st
       COUNT=$(ogrinfo -so $output $(basename $output
      → .shp) |grep Count|awk -v FS=':' '{print $2}')
477   if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
          ogr2ogr -update -append
      → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
      → nfi_mamora
479   else
          ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
481   fi
      rm ${TMPDIR}/$(basename $output .shp).*
483       done
          echo "processed $ORIGSTRING1 $ORIGSTRING2"
485   else
          echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
      → processed due to missing information"
487       echo "Type1: ${TYPE1}, Density1: ${DENSITY1},
      → Trunk1: ${TRUNK1}, Stage1: ${STAGE1}, Type2:
      → ${TYPE2}" ];
      fi
489     done
    done
491
    echo "process two essence in Reboisement echec
       → (plan2=2, stage2=e)"
493 awk '{ if (NF==4&&$2=="Rb"&&$4=="echec") print }'
       → ${ROOTDIR}/essence_corrected.txt
    #examples:
495 #Qs2fv Rb A.mo echec
    TMPDIR=${ROOTDIR}/tmp
497 mkdir -p ${TMPDIR}
    OUTPUTDIR=${ROOTDIR}/vito
499 mkdir -p ${OUTPUTDIR}
    awk '{ if (NF==4&&$2=="Rb"&&$4=="echec") print $1,$3
```

```
 → }' ${ROOTDIR}/essence_corrected.txt|while read
 → ORIGSTRING1 ORIGSTRING2;do
501  STRING1=$(echo $ORIGSTRING1|sed -e
 → 's/[jav]\/./m/g' -e 's/\/.//')
 STRING2=$(echo $ORIGSTRING2|sed -e
 → 's/[jav]\/./m/g' -e 's/\/.//')
503
  echo "${STRING1} ${STRING2}"|sed
 → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\) \(.*\)/\1
 → \2 \3 \4 \5/'|while read TYPE1 DENSITY1 TRUNK1
 → STAGE1 TYPE2;do
505 if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n
 → "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}"
 → ];then
    for file in $(find ${ROOTDIR} -name
 → 'canton[ABCDE]*_strate.shp');do
507  output=${TMPDIR}/$(basename $file
 → .shp)_${STRING1}_${STRING2}.shp
  pkeditogr -i $file -o $output -an type1 -at
 → String -av ${TYPE1} -an type2 -at String -av
 → ${TYPE2} -an type3 -at String -av NA -an
 → density1 -at Integer -av ${DENSITY1} -an
 → density2 -at Integer -av 0 -an density3 -at
 → Integer -av 0 -an plan1 -at Integer -av 0 -an
 → plan2 -at Integer -av 2 -an plan3 -at Integer
 → -av 0 -an trunk1 -at String -av ${TRUNK1} -an
 → trunk2 -at String -av NA -an trunk3 -at String
 → -av NA -an stage1 -at String -av ${STAGE1} -an
 → stage2 -at String -av e -an stage3 -at String
 → -av NA -select ESSENCE -like "${ORIGSTRING1} Rb
 → ${ORIGSTRING2} echec" -st
509  COUNT=$(ogrinfo -so $output $(basename $output
 → .shp) |grep Count|awk -v FS=':' '{print $2}')
 if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
511     ogr2ogr -update -append
 → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
 → nfi_mamora
 else
513     ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
 fi
515 rm ${TMPDIR}/$(basename $output .shp).*
    done
517     echo "processed $ORIGSTRING1 $ORIGSTRING2"
 else
519     echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
 → processed due to missing information"
```

```
        echo "Type1: ${TYPE1}, Density1: ${DENSITY1},
    → Trunk1: ${TRUNK1}, Stage1: ${STAGE1}, Type2:
    → ${TYPE2}" ];
  fi
    done
done

echo "process three essences (always with
    → Regeneration or Reboisement)"

echo "process three essences in Regeneration"
awk '{ if (NF==4&&$3=="Rg") print }'
    → ${ROOTDIR}/essence_corrected.txt
#examples:
#Qs2fv Pcm3fv Rg Qs4fj
TMPDIR=${ROOTDIR}/tmp
mkdir -p ${TMPDIR}
OUTPUTDIR=${ROOTDIR}/vito
mkdir -p ${OUTPUTDIR}
awk '{ if (NF==4&&$3=="Rg") print $1,$2,$4 }'
    → ${ROOTDIR}/essence_corrected.txt|while read
    → ORIGSTRING1 ORIGSTRING2 ORIGSTRING3;do
    STRING1=$(echo $ORIGSTRING1|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')
    STRING2=$(echo $ORIGSTRING2|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')
    STRING3=$(echo $ORIGSTRING3|sed -e
    → 's/[jav]\/./m/g' -e 's/\/.//')

     TYPE1=$(echo "${STRING1}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1/')
     DENSITY1=$(echo "${STRING1}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\2/')
     TRUNK1=$(echo "${STRING1}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\3/')
     STAGE1=$(echo "${STRING1}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\4/')
     TYPE2=$(echo "${STRING2}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1/')
     DENSITY2=$(echo "${STRING2}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\2/')
     TRUNK2=$(echo "${STRING2}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\3/')
     STAGE2=$(echo "${STRING2}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\4/')
     TYPE3=$(echo "${STRING3}"|sed
```

```
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\1/')
549  DENSITY3=$(echo "${STRING3}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\2/')
     TRUNK3=$(echo "${STRING3}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\3/')
551  STAGE3=$(echo "${STRING3}"|sed
    → 's/\(.*\)\([1234]\)\([ft]\)\([javm]\)/\4/')
    if [ -n "${TYPE1}" -a -n "${DENSITY1}"  -a -n
    → "${TRUNK1}" -a -n "${STAGE1}" -a -n "${TYPE2}"
    → -a -n "${DENSITY2}" -a -n "${TRUNK2}" -a -n
    → "${STAGE2}"  -a -n "${TYPE3}" -a -n
    → "${DENSITY3}"  -a -n "${TRUNK3}" -a -n
    → "${STAGE3}"  ];then
553     echo "Type1: ${TYPE1}, Density1: ${DENSITY1},
    → Trunk1: ${TRUNK1}, Stage1: ${STAGE1} Type2:
    → ${TYPE2}, Density2: ${DENSITY2}, Trunk2:
    → ${TRUNK2}, Stage2: ${STAGE2} Type3: ${TYPE3},
    → Density3: ${DENSITY3}, Trunk3: ${TRUNK3},
    → Stage3: ${STAGE3}"
        for file in $(find ${ROOTDIR} -name
    → 'canton[ABCDE]*_strate.shp');do
555  output=${TMPDIR}/$(basename $file
    → .shp)_${STRING1}_${STRING2}_${STRING3}.shp
     pkeditogr -i $file -o $output -an type1 -at
    → String -av ${TYPE1} -an type2 -at String -av
    → ${TYPE2} -an type3 -at String -av ${TYPE3} -an
    → density1 -at Integer -av ${DENSITY1} -an
    → density2 -at Integer -av ${DENSITY2} -an
    → density3 -at Integer -av ${DENSITY3} -an plan1
    → -at Integer -av 0 -an plan2 -at Integer -av 0
    → -an plan3 -at Integer -av 1 -an trunk1 -at
    → String -av ${TRUNK1} -an trunk2 -at String -av
    → ${TRUNK2} -an trunk3 -at String -av ${TRUNK3}
    → -an stage1 -at String -av ${STAGE1} -an stage2
    → -at String -av ${STAGE2} -an stage3 -at String
    → -av ${STAGE3} -select ESSENCE -like
    → "${ORIGSTRING1} ${ORIGSTRING2} Rg
    → ${ORIGSTRING3}" -st
557  COUNT=$(ogrinfo -so $output $(basename $output
    → .shp) |grep Count|awk -v FS=':' '{print $2}')
    if [ -f ${OUTPUTDIR}/nfi_mamora.shp ];then
559     ogr2ogr -update -append
    → ${OUTPUTDIR}/nfi_mamora.shp ${output} -nln
    → nfi_mamora
    else
561     ogr2ogr ${OUTPUTDIR}/nfi_mamora.shp ${output}
```

```
    fi
563 rm ${TMPDIR}/$(basename $output .shp).*
        done
565     echo "processed $ORIGSTRING1 $ORIGSTRING2
     → $ORIGSTRING3"
    else
567     echo "Warning: $ORIGSTRING1 $ORIGSTRING2 not
     → processed due to missing information"
        echo "Type1: ${TYPE1}, Density1: ${DENSITY1},
     → Trunk1: ${TRUNK1}, Stage1: ${STAGE1} Type2:
     → ${TYPE2}, Density2: ${DENSITY2}, Trunk2:
     → ${TRUNK2}, Stage2: ${STAGE2} Type3: ${TYPE3},
     → Density3: ${DENSITY3}, Trunk3: ${TRUNK3},
     → Stage3: ${STAGE3}" ];
569 fi
done
571
echo "process three essences in Reboisement"
573 awk '{ if (NF==4&&$2=="Rb"&&$4!="echec") print }'
    → ${ROOTDIR}/essence_corrected.txt
#examples:
575 #(empty)

577 rmdir ${TMPDIR}

579 for DENSITY in 1 2 3 4;do
     ogr2ogr -where "(type1="Qs" AND type2="NA" and
    → type3="NA" and density1=$DENSITY and density2=0
    → and density3=0)" ${OUTPUTDIR}/qs_${DENSITY}.shp
    → ${OUTPUTDIR}/nfi_mamora.shp
581  ogr2ogr -where "((type1="Eu.ca" or type1="Eu.cl"
    → or type1="Eu.go" or type1="Eu.gr") AND
    → type2="NA" and type3="NA" and density1=$DENSITY
    → and density2=0 and density3=0)"
    → eu_${DENSITY}.shp nfi_mamora.shp
     ogr2ogr -where "((type1="A.cy" or type1="A.mo")
    → AND type2="NA" and type3="NA" and
    → density1=$DENSITY and density2=0 and
    → density3=0)" a_${DENSITY}.shp nfi_mamora.shp
583  ogr2ogr -where "((type1="Ph" or type1="P.pa" or
    → type1="P.r") AND type2="NA" and type3="NA" and
    → density1=$DENSITY and density2=0 and
    → density3=0)" p_${DENSITY}.shp nfi_mamora.shp
done
```

/storage/hyper2/_Projects/NI079_FFEM/nfi/process.sh

D.2 Automated classification process (bash script)

```
#!/bin/bash

MINCOVER=25
BALANCE=200
#KMLDIR=/media/1MAR/0LANDSAT/REF/TRAINING/KML
REFDIR=/storage/hyper2/_Projects/NI079_FFEM/ref
KMLDIR=/storage/hyper2/_Projects/NI079_FFEM/training/kml
#SHPDIR=/media/1MAR/0LANDSAT/REF/TRAINING/SHP
SHPDIR=/storage/hyper2/_Projects/NI079_FFEM/training/shp
NFIDIR=/storage/hyper2/_Projects/NI079_FFEM/nfi/STRATES/vito
#INPUTDIR=/media/1MAR/0LANDSAT/TM
ORIGINPUTDIR=/storage/hyper2/_Projects/NI079_FFEM/landsat/orig
INPUTDIR=/storage/hyper2/_Projects/NI079_FFEM/landsat/input
#MAPDIR=/media/1MAR/0LANDSAT/MAP
MAPDIR=/storage/hyper2/_Projects/NI079_FFEM/landsat/output
FRACTIONDIR=/storage/hyper2/_Projects/NI079_FFEM/landsat/output

CREATE_INPUT=0
INIT_TRAINING=0
CREATE_TRAINING=0
KML=0
NFI=1
SDVI=0
CLASSIFY=0
MASKMAP=1
CLASSIFIER=SVM

function usage
{
    echo "usage: $0 --stack --init --mincover
    → --training --classify --classifier [svm|ann]
    → --input <input> --output <output> [-h]"
    echo
    echo "--init     : create training sample from
    → KML"
    echo "--stack    : create input images (stack
    → bands)"
    echo "--mincover : minimum forest cover in
    → percentage, use one of (5 10 15 25 30 35 40 45
    → 50 55 60 65 70 75 80 85 90 95)"
    echo "--training : create training sample for
    → selected input image"
    echo "--classify : classfiy selected input image"
    echo "--input    : input image (e.g., landsat
```

```
    → image) with all bands stacked"
38     echo "--output   : output map"
}
40
while [ "$1" != "" ]; do
42     case $1 in
          -s | --stack )    CREATE_INPUT=1
44       echo "create input images (stack bands)"
                                ;;
46         -init | --init )    INIT_TRAINING=1
         echo "create training sample"
48                              ;;
          -kml | --kml )    KML=1
50       echo "use KML vectors"
                                ;;
52         -nfi | --nfi )    NFI=1
         echo "use NFI vectors"
54                              ;;
          -sdvi | --sdvi )    SDVI=1
56       echo "calculate SDVI as a forest cover proxy"
                                ;;
58         -min | --mincover )          shift
             MINCOVER=$1
60       echo "minimum forest cover in percentage:
    → ${MINCOVER}%"
             ;;
62         -i | --input )          shift
         INPUT=$1
64       echo "input is $INPUT"
                                ;;
66         -o | --output )          shift
         OUTPUT=$1
68       echo "output is $OUTPUT"
                                ;;
70         -t | --training )    CREATE_TRAINING=1
         echo "create training for input image"
72                              ;;
          -c | --classify )    CLASSIFY=1
74       echo "classify image"
                                ;;
76         -ct | --classifier ) shift
         CLASSIFIER=$1
78       echo "classifier=$CLASSIFIER"
             ;;
80         -h | --help )           usage
                                exit
```

```
82                                      ;;
        * )                             usage
84                                      exit 1
    esac
86  shift
done
88
if [ "${CREATE_INPUT}" -eq 1 ];then
90  echo "stack bands 1 2 3 4 for LM data"
    for file in ${ORIGINPUTDIR}/LM*_B1.TIF;do
92  pkcrop $(for B in 1 2 3 4;do echo -i
    → ${ORIGINPUTDIR}/$(basename $file .TIF|sed
    → "s/B1/B$B/").TIF;done) -e
    → ${REFDIR}/Maamora_limite_32630.shp -o
    → ${ORIGINPUTDIR}/$(basename $file .TIF|sed
    → "s/_B1//").tif -co COMPRESS=LZW -co
    → INTERLEAVE=BAND -co TILED=YES
    gdalwarp -t_srs epsg:26191
    → ${ORIGINPUTDIR}/$(basename $file .TIF|sed
    → "s/_B1//").tif ${INPUTDIR}/$(basename $file
    → .TIF|sed "s/_B1//").tif
94  done
    echo "stack bands 1 2 3 4 5 7 for LT and LE data"
96  for file in ${ORIGINPUTDIR}/L[ET]*_B1.TIF;do
    pkcrop $(for B in 1 2 3 4 5 7;do echo -i
    → ${ORIGINPUTDIR}/$(basename $file .TIF|sed
    → "s/B1/B$B/").TIF;done) -e
    → ${REFDIR}/Maamora_limite_32630.shp -o
    → ${ORIGINPUTDIR}/$(basename $file .TIF|sed
    → "s/_B1//").tif -co COMPRESS=LZW -co
    → INTERLEAVE=BAND -co TILED=YES
98  gdalwarp -t_srs epsg:26191
    → ${ORIGINPUTDIR}/$(basename $file .TIF|sed
    → "s/_B1//").tif ${INPUTDIR}/$(basename $file
    → .TIF|sed "s/_B1//").tif
    done
100 fi

102 if [ "${INIT_TRAINING}" -eq 1 ];then
    if [ "${KML}" -eq 1 ];then
104 rm -f ${SHPDIR}/*_kml.shp;
    echo "convert KML to ESRI Shapefile and reproject
    → to UTM (epsg:26191)"
106 for file in ${KMLDIR}/*.kml; do
        rm -f ${SHPDIR}/$(basename $file .kml).shp
108     ogr2ogr -t_srs epsg:26191 -f "ESRI Shapefile"
```

```
  → ${SHPDIR}/$(basename $file .kml).shp $file
done
echo "create non forest training set"
for file in ${SHPDIR}/noforest_*[0-9].shp;do
    rm -f ${SHPDIR}/$(basename $file .shp)_kml.shp
    pkeditogr -i $file -o ${SHPDIR}/$(basename
  → $file .shp)_kml.shp -an label -at Integer -av 2
done
echo "create forest training set"
for ((T=$MINCOVER;T<=100;T+=5));do
    for file in ${SHPDIR}/forest_${T}_*[0-9].shp;do
  if [ -f $file ];then
      rm -f ${SHPDIR}/$(basename $file .shp)_kml.shp
      pkeditogr -i $file -o ${SHPDIR}/$(basename
  → $file .shp)_kml.shp -an label -at Integer -av 1
  fi
    done
done
echo "merge all shape files to single training set"
rm -f ${SHPDIR}/training_kml.shp
FILENR=1
for file in ${SHPDIR}/*_kml.shp;do
    if [ "${FILENR}" -eq 1 ];then
  ogr2ogr ${SHPDIR}/training_kml.shp $file
    else
  ogr2ogr -append -update
  → ${SHPDIR}/training_kml.shp $file -nln
  → training_kml
    fi
    FILENR=$(($FILENR+1))
done
  fi
  if [ "${NFI}" -eq 1 ];then
rm -f ${SHPDIR}/*_nfi.shp;
TMPDIRSHP=${SHPDIR}/tmp$$
mkdir $TMPDIRSHP;
echo "init training for Pine forest"
ogr2ogr -where "type1 like 'P%' AND density1=1 AND
  → density2=0 AND density3=0" $TMPDIRSHP/p.shp
  → $NFIDIR/nfi_mamora.shp
pkeditogr -i $TMPDIRSHP/p.shp -o
  → ${TMPDIRSHP}/p_label.shp -an label -at String
  → -av p
echo "init training for Acacia forest"
ogr2ogr -where "type1 like 'A%' AND density1=1 AND
  → density2=0 AND density3=0" $TMPDIRSHP/a.shp
```

```
 → $NFIDIR/nfi_mamora.shp
pkeditogr -i $TMPDIRSHP/a.shp -o
 → ${TMPDIRSHP}/a_label.shp -an label -at String
 → -av a
echo "init training for Eucalyptus forest"
ogr2ogr -where "type1 like 'Eu%' AND density1=1 AND
 → density2=0 AND density3=0" $TMPDIRSHP/eu.shp
 → $NFIDIR/nfi_mamora.shp
pkeditogr -i $TMPDIRSHP/eu.shp -o
 → ${TMPDIRSHP}/eu_label.shp -an label -at String
 → -av eu
echo "init training for Quercus forest"
ogr2ogr -where "type1='qs' AND stage1!='e' AND
 → stage2!='e' and stage3!='3' AND density1=1 AND
 → density2=0 AND density3=0" $TMPDIRSHP/qs.shp
 → $NFIDIR/nfi_mamora.shp
pkeditogr -i $TMPDIRSHP/qs.shp -o
 → ${TMPDIRSHP}/qs_label.shp -an label -at String
 → -av qs
echo "init training for Non forest"
ogr2ogr -where "density1=3 OR density1=4"
 → $TMPDIRSHP/nf.shp $NFIDIR/nfi_mamora.shp
pkeditogr -i $TMPDIRSHP/nf.shp -o
 → ${TMPDIRSHP}/nf_label.shp -an label -at String
 → -av NF
echo "merge training"
rm -f ${SHPDIR}/training_nfi.shp
FILENR=1
for file in ${TMPDIRSHP}/*_label.shp;do
    if [ "${FILENR}" -eq 1 ];then
  ogr2ogr ${SHPDIR}/training_nfi.shp $file
    else
  ogr2ogr -append -update
 → ${SHPDIR}/training_nfi.shp $file -nln
 → training_nfi
    fi
    FILENR=$(($FILENR+1))
done
rm $TMPDIRSHP/*.dbf $TMPDIRSHP/*.prj
 → $TMPDIRSHP/*.shp $TMPDIRSHP/*.shx
rmdir $TMPDIRSHP
  fi
  #   for TYPE in a eu p qs;do
  #       echo "init training for forest based on
 → $TYPE"
  #       for DENSITY in 1;do
```

```
#     file=${NFIDIR}/${TYPE}_${DENSITY}.shp;
#     if [ -f $file ];then
#         rm -f ${SHPDIR}/$(basename $file .shp)_nfi.shp
#         pkeditogr -i $file -o ${SHPDIR}/$(basename $file .shp)_nfi.shp -an label -at String -av $TYPE
#     fi
#       done
#       echo "init training for non forest based on $TYPE"
#       for DENSITY in 3 4;do
#     file=${NFIDIR}/${TYPE}_${DENSITY}.shp;
#     if [ -f $file ];then
#         rm -f ${SHPDIR}/$(basename $file .shp)_nfi.shp
#         pkeditogr -i $file -o ${SHPDIR}/$(basename $file .shp)_nfi.shp -an label -at String -av NF
#     fi
#       done
#   done
#   echo "merge training"
#   rm -f ${SHPDIR}/training_nfi.shp
#   FILENR=1
#   for file in ${SHPDIR}/*_nfi.shp;do
#       if [ "${FILENR}" -eq 1 ];then
#     ogr2ogr ${SHPDIR}/training_nfi.shp $file
#       else
#     ogr2ogr -append -update ${SHPDIR}/training_nfi.shp $file -nln training_nfi
#       fi
#       FILENR=$(($FILENR+1))
#   done
# fi
fi

if [ "${SDVI}" -eq 1 ];then
    for TYPE in a eu p qs;do
  for DENSITY in 1 2 3 4;do
      file=${NFIDIR}/${TYPE}_${DENSITY}.shp;
      if [ -f $file ];then
    echo "extract reflectance information"
    COUNT=$(ogrinfo $file $(basename $file .shp) |grep Count|awk -v FS=':' '{print $2}')
```

```
208     if [ "${COUNT}" -gt 0 ];then
            rm -f ${FRACTIONDIR}/$(basename ${INPUT}
      → .tif)_${TYPE}_${DENSITY}.*
210         pkextract -i ${INPUT} -s ${file} -o
      → ${FRACTIONDIR}/$(basename ${INPUT}
      → .tif)_${TYPE}_${DENSITY}.shp -l -r point
            pkdumpogr -i ${FRACTIONDIR}/$(basename
      → ${INPUT} .tif)_${TYPE}_${DENSITY}.shp -n B1 -n
      → B2 -n B3 -n B4 -n B5 -n B6 -o
      → ${FRACTIONDIR}/$(basename ${INPUT}
      → .tif)_${TYPE}_${DENSITY}.txt
212     fi
        fi
214   done
      echo "calculate red and nir for soil and dense
       → vegetation"
216   echo "calculate SDVI"
      for DENSITY in 1 2 3 4;do
218       file=${NFIDIR}/${TYPE}_${DENSITY}.shp;
          COUNT=$(ogrinfo $file $(basename $file .shp)
       → |grep Count|awk -v FS=':' '{print $2}')
220       if [ "${COUNT}" -gt 0 ];then
        RED_V=$(pkstatogr -i $file_v -n B1 -min|awk -v
       → FS="--mean" '{print $2}')
222     RED_S=$(pkstatogr -i $file_s -n B1 -max|awk -v
       → FS="--mean" '{print $2}')
        NIR_V=$(pkstatogr -i $file_v -n B2 -max|awk -v
       → FS="--mean" '{print $2}')
224     NIR_S=$(pkstatogr -i $file_s -n B2 -mean|awk -v
       → FS="--mean" '{print $2}')
        echo "process
       → ${FRACTIONDIR}/${TYPE}_${DENSITY}_sdvi.txt"
226     awk -v RED_V="${RED_V}" -v RED_S="${RED_S}" -v
       → NIR_V="${NIR_V}" -v NIR_S="${NIR_S}" 'BEGIN
       → {DVI_V=NIR_V-RED_V;DVI_S=NIR_S-RED_S};{RED=$3;NIR=$4;
       → DVI=NIR-RED;
       → SDVI=100.0*(DVI-DVI_S)/(DVI_V-DVI_S); print
       → SDVI}' ${FRACTIONDIR}/$(basename ${INPUT}
       → .tif)_${TYPE}_${DENSITY}.txt >
       → ${FRACTIONDIR}/$(basename ${INPUT}
       → .tif)_${TYPE}_${DENSITY}_sdvi.txt
          fi
228   done
        done
230 fi
```

```
232 if [ "${CLASSIFY}" -eq 1 ];then
    #0 0 0 0 255
234 #1 51 153 85 255
    #2 252 251 212 255
236 #3 207 235 252 255
      if [ ! -f ${INPUT} ];then
238 echo "Error: input image $INPUT not found!"
    exit
240   fi
      echo "extract image data for training"
242   if [ "${KML}" -eq 1 ];then
    TRAINING=${SHPDIR}/training_kml_$(basename ${INPUT}
     → .tif).shp
244 if [ -z "$OUTPUT" ];then
        OUTPUT=${MAPDIR}/$(basename ${INPUT}
     → .tif)_kml_map.tif
246 fi
    if [ "${CREATE_TRAINING}" -eq 1 ] ||
     → "${INIT_TRAINING}" -eq 1 ];then
248     rm -f ${TRAINING}
    fi
250 if [ ! -f ${TRAINING} ];then
        pkextract -i ${INPUT} -s
     → ${SHPDIR}/training_kml.shp -o ${TRAINING} -r mean
252 fi
    rm -f ${OUTPUT}
254 echo "classify input image $INPUT"
    if [ "${CLASSIFIER}" == "SVM" ];then
256     pkclassify_svm -t ${TRAINING} -cc 1000 -g 1
     → -bal ${BALANCE} -i ${INPUT} -o ${OUTPUT} -ct
     → ${MAPDIR}/colortable_type.txt -cv 2
    elif [ "${CLASSIFIER}" == "ANN" ];then
258     pkclassify_nn -t ${TRAINING} -bal ${BALANCE} -i
     → ${INPUT} -o ${OUTPUT} -ct
     → ${MAPDIR}/colortable_type.txt -cv 2
    else
260     echo "Error: classifier ${CLASSIFIER} not
     → supported, please us SVM or ANN"
        exit
262 fi
      elif [ "${NFI}" -eq 1 ];then
264 TRAINING=${SHPDIR}/training_nfi_$(basename ${INPUT}
     → .tif).shp
    if [ -z "$OUTPUT" ];then
266     OUTPUT=${MAPDIR}/$(basename ${INPUT}
     → .tif)_nfi_map.tif
```

```
    fi
268 if [ ${CREATE_TRAINING} -eq 1 ] || ${INIT_TRAINING}
     → -eq 1 ];then
        rm -f ${TRAINING}
270 fi
    if [ ! -f ${TRAINING} ];then
272     pkextract -i ${INPUT} -s
     → ${SHPDIR}/training_nfi.shp -o ${TRAINING} -r
     → point -l
        # pkextract -i ${INPUT} -s
     → ${SHPDIR}/training_nfi.shp -o ${TRAINING} -r mean
274 fi
    rm -f ${OUTPUT}
276 echo "classify input image $INPUT"
    if [ "${CLASSIFIER}" == "SVM" ];then
278     pkclassify_svm -t ${TRAINING} -cc 1000 -g 1
     → -bal ${BALANCE} -i ${INPUT} -o ${OUTPUT} -c qs
     → -r 11 -c a -r 12 -c eu -r 13 -c p -r 14 -c NF -r
     → 20 -ct ${MAPDIR}/colortable_type.txt -cv 2
    elif [ "${CLASSIFIER}" == "ANN" ];then
280     pkclassify_nn -t ${TRAINING} -bal ${BALANCE} -i
     → ${INPUT} -o ${OUTPUT} -c qs -r 11 -c a -r 12 -c
     → eu -r 13 -c p -r 14 -c NF -r 20 -ct
     → ${MAPDIR}/colortable_type.txt -cv 2
    else
282     echo "Error: classifier ${CLASSIFIER} not
     → supported, please us SVM or ANN"
        exit
284 fi
      fi
286   if [ "${MASKMAP}" ];then
    EXTENT=${REFDIR}/Maamora_limite_26191.shp
288 pkcrop -i ${OUTPUT} -e ${EXTENT} -o ${OUTPUT}
    gdal_rasterize -i -burn 0 -l $(basename ${EXTENT}
     → .shp) ${EXTENT} ${OUTPUT}
290   fi
fi
292
#kml:
294 #for file in landsat/input/LT52010361987239XXX03.tif
    → landsat/input/LT52010361987255XXX03.tif ; do
    → ./classify.sh -kml --init --training --classify
    → -i $file;done
#pkmosaic -m 6 $(for file in
    → LT52010361987*_kml_map.tif;do echo -i
```

```
→ $file;done) -o 1990_reference_kml_map_type.tif
```

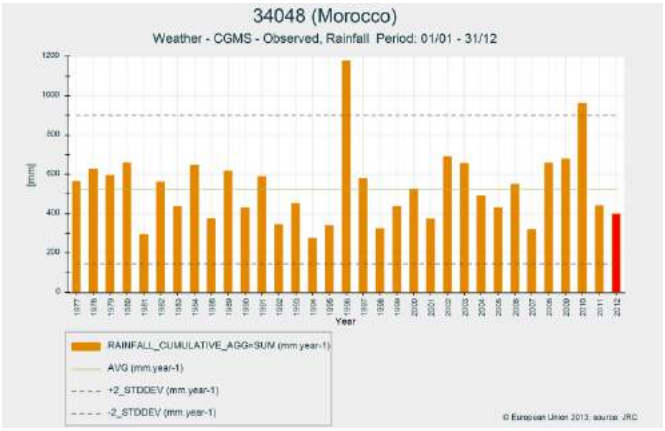
/storage/hyper2/_Projects/NI079_FFEM/classify.sh

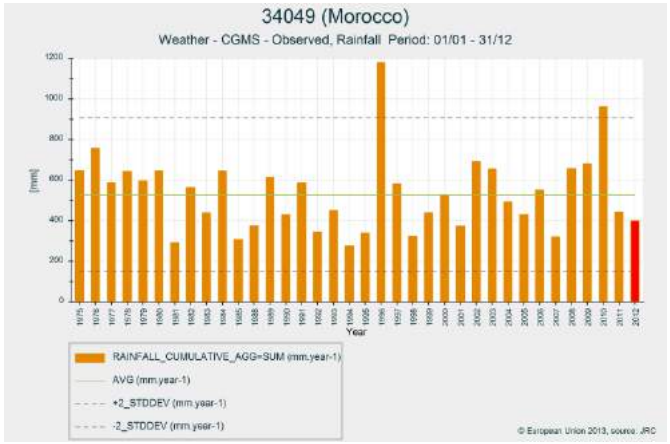
Rainfall panel:

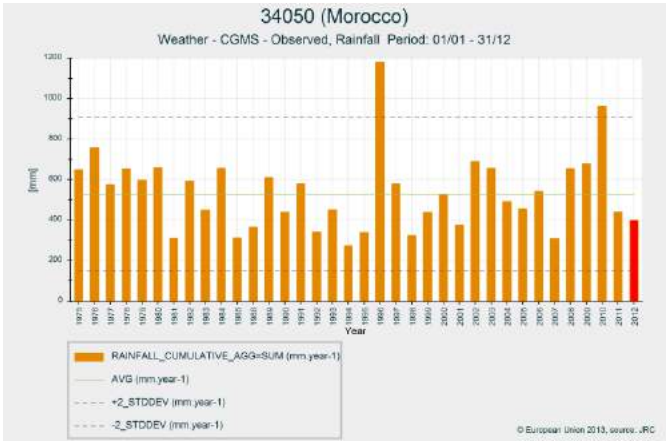
- Yearly rainfall
- Standard Precipitation Index
- Number of days > 5mm
- Number of days > 30mm

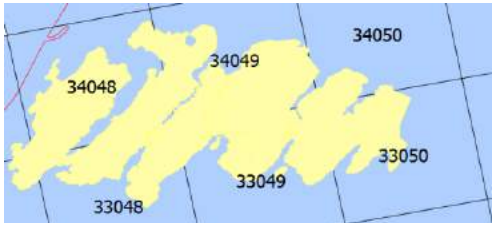
Maamora forest domain

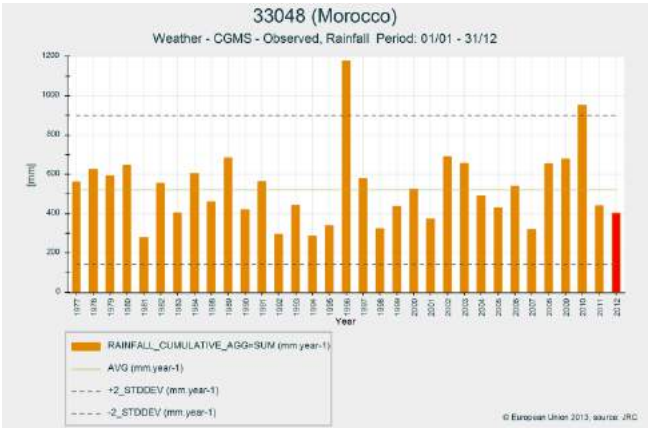
Yearly rainfall

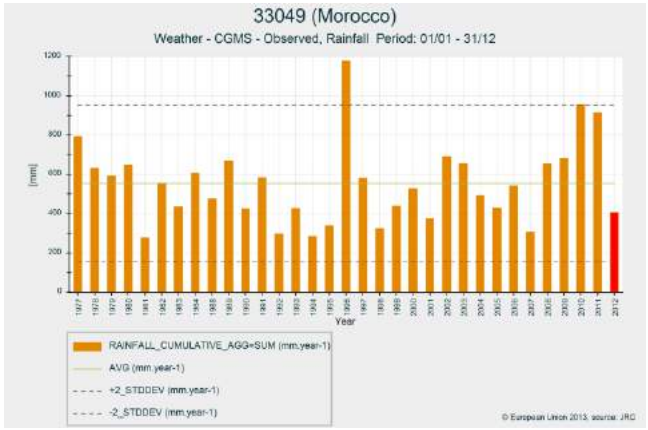


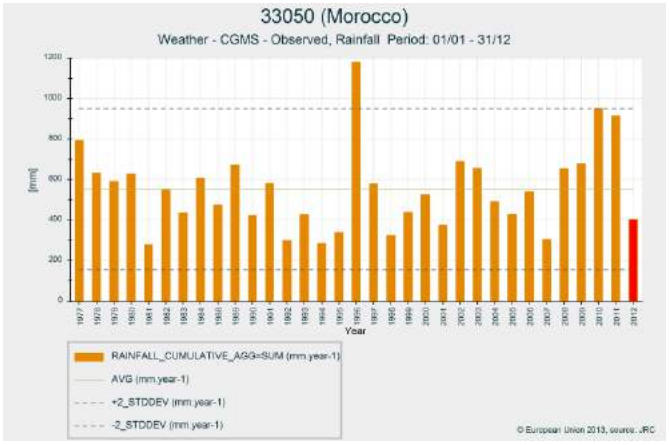






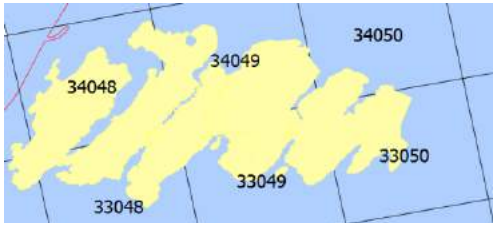


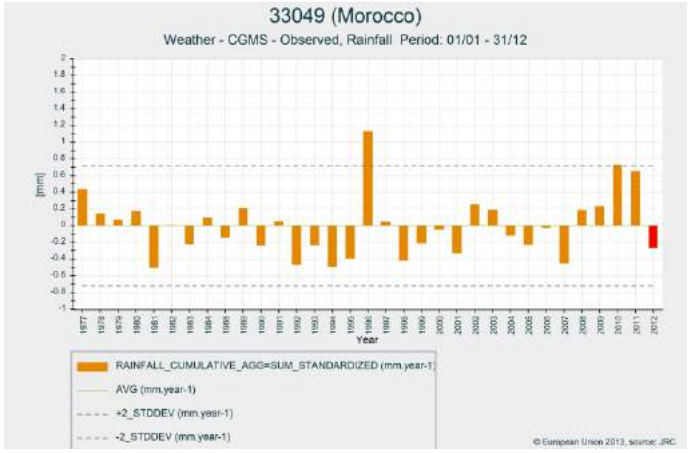




Maamora forest domain

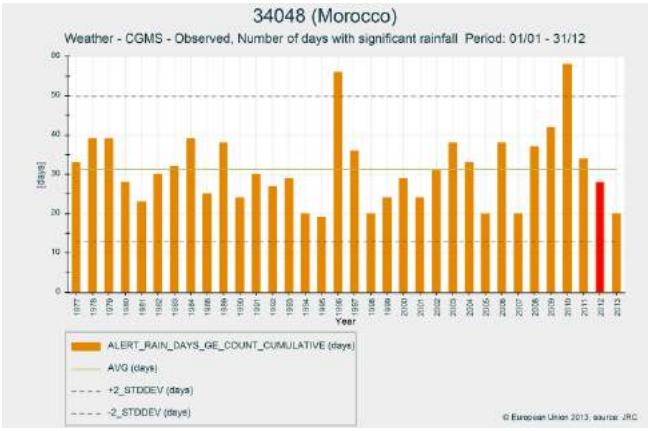
Standard precipitation index

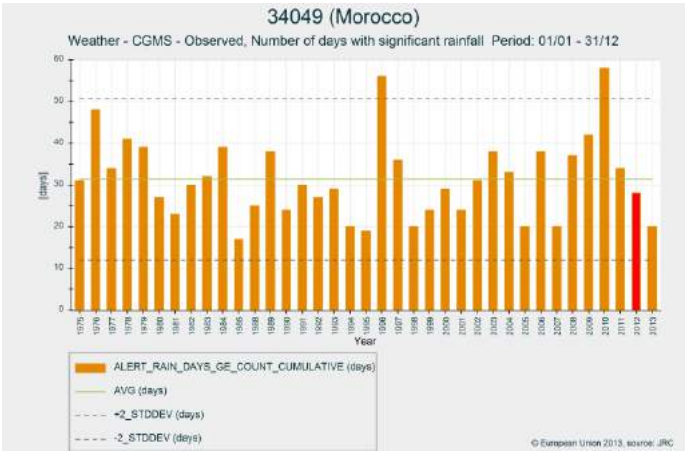


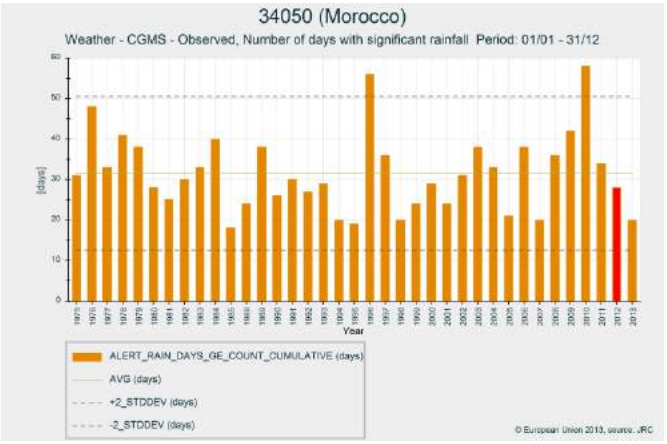


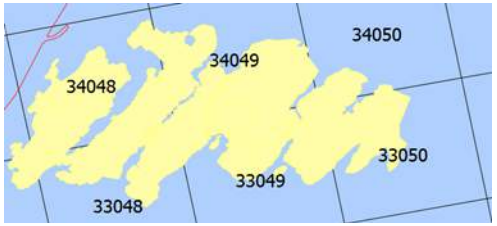
Maamora forest domain

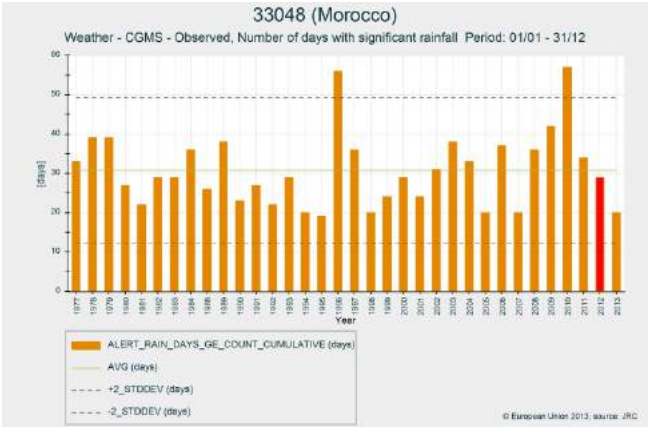
Number of daily rainfall >5mm

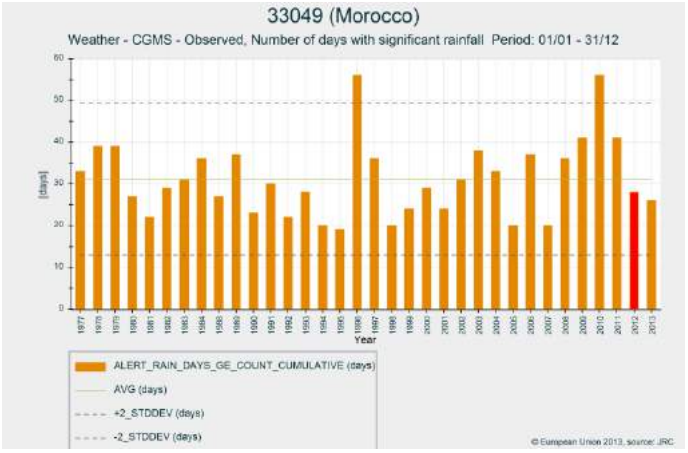


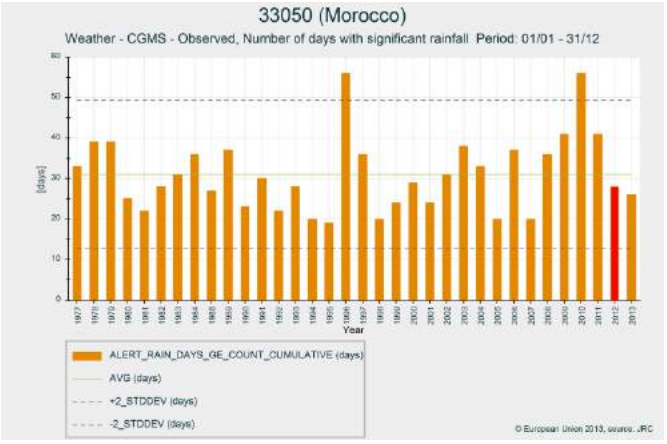






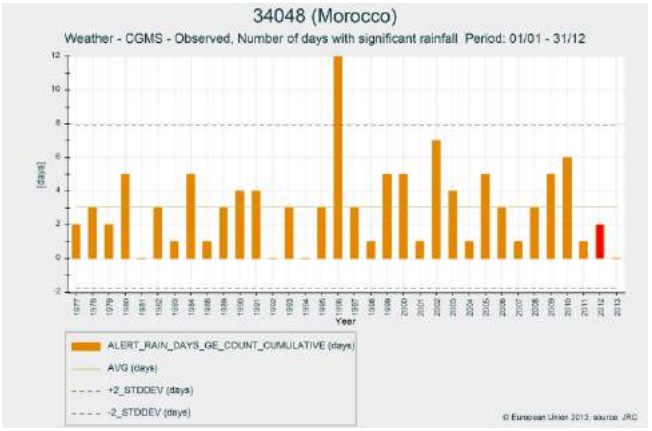


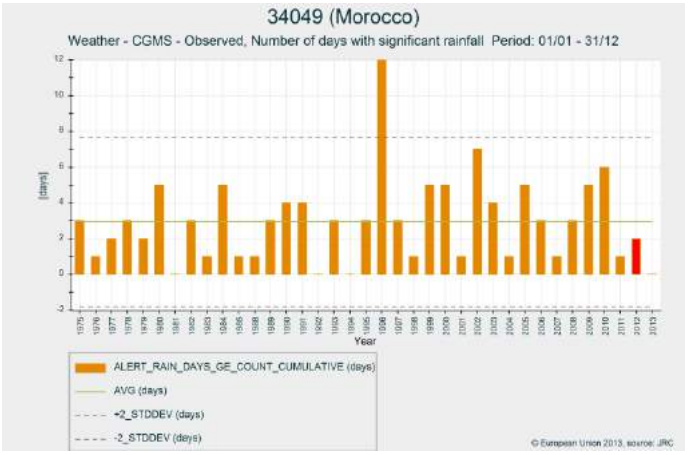


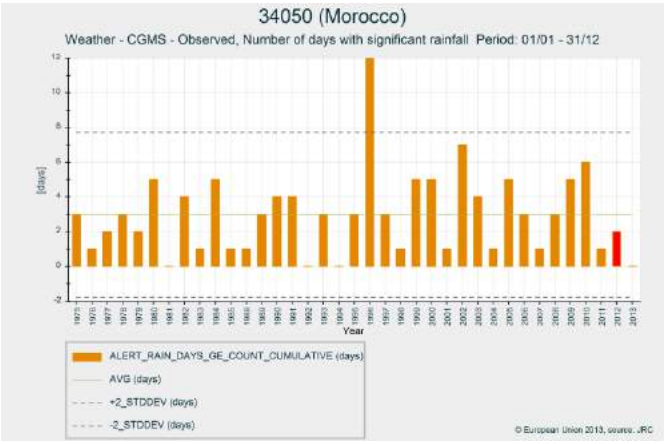


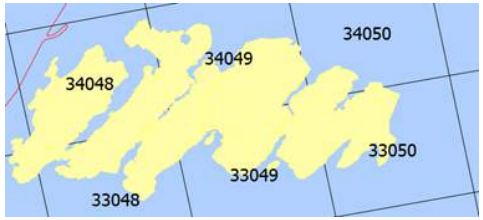
Maamora forest domain

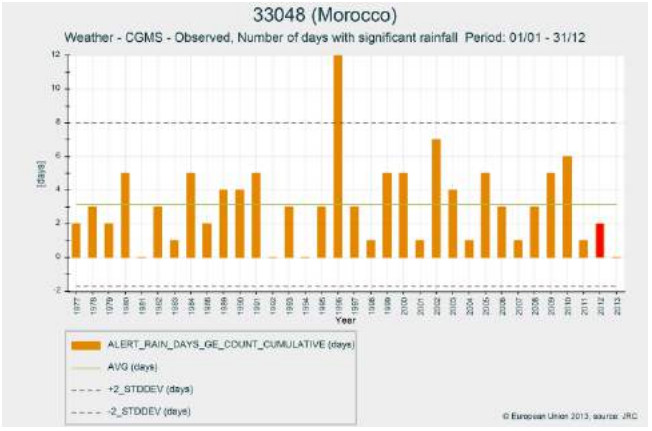
Number of daily rainfall >30mm

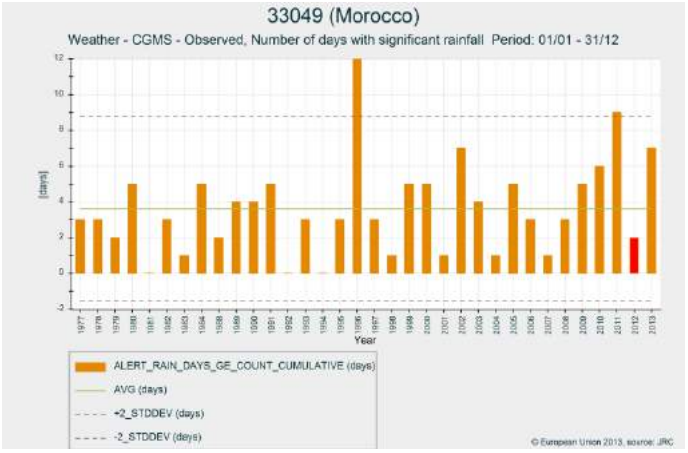


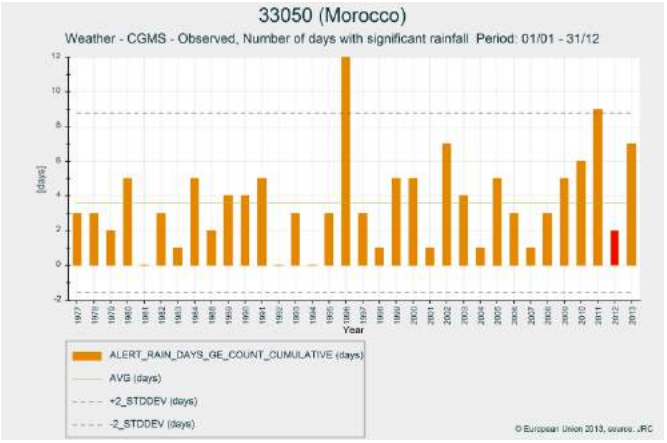










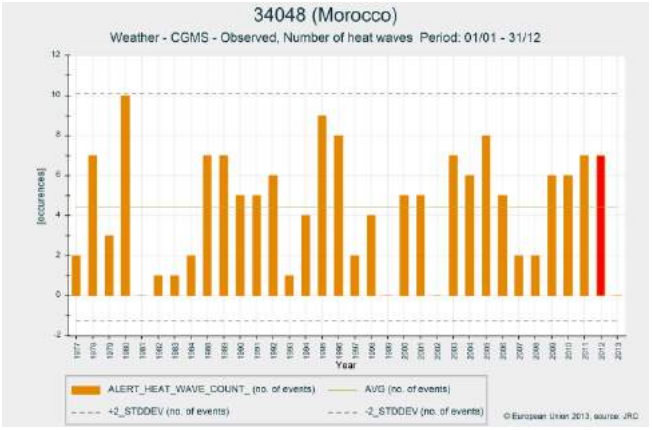


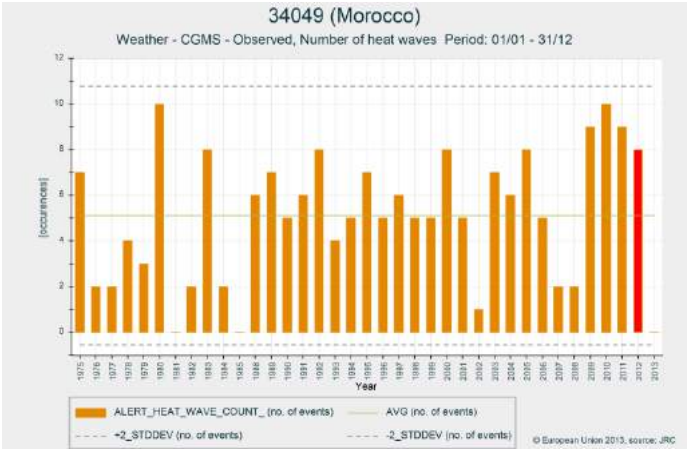
Dash board on the heat wave indicators

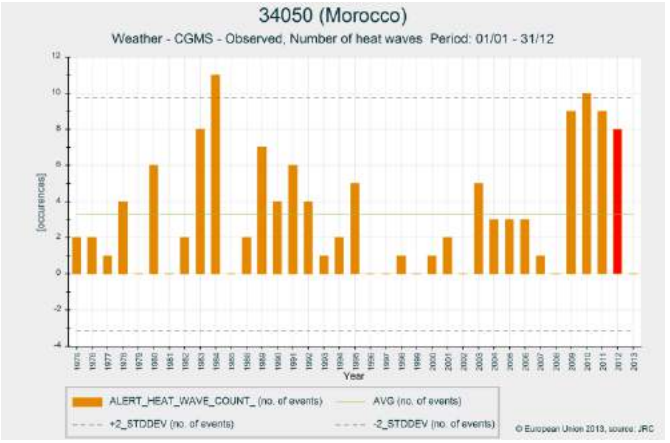
- Number of days >35°C
- Number of days >40°C
- Number of heat waves
- Longest heat wave period

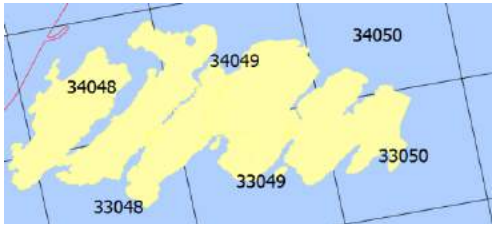
Maamora forest domain

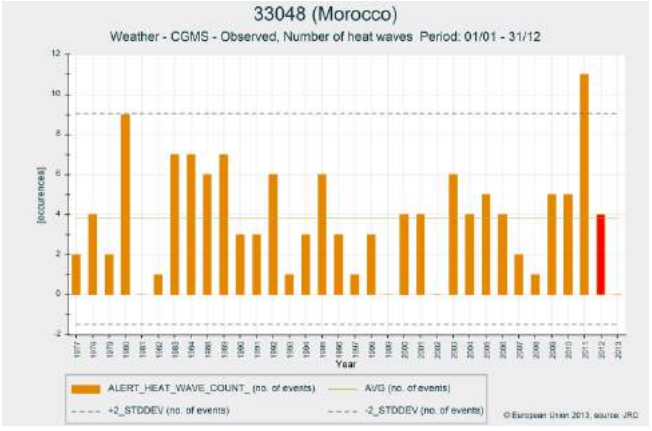
Nb of heat waves (>= 3 consecutive days)

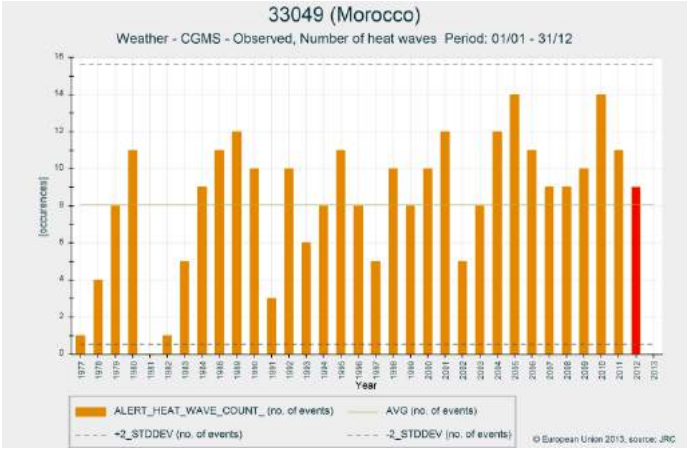


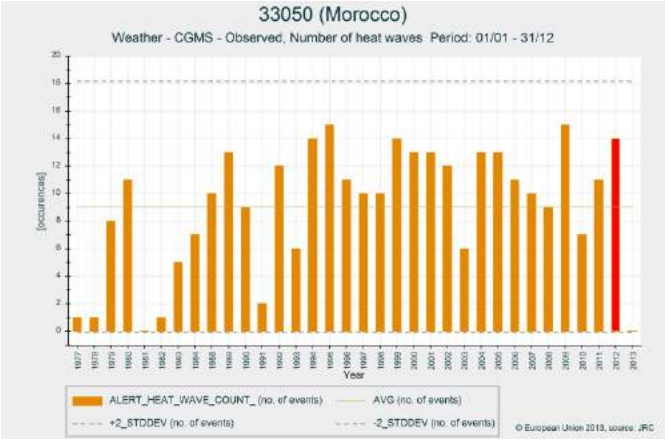






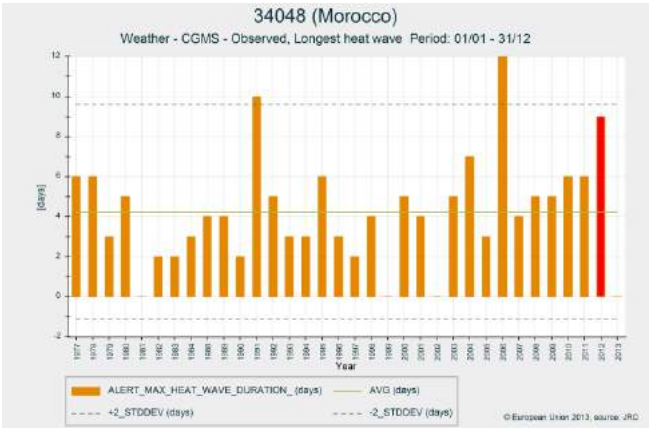


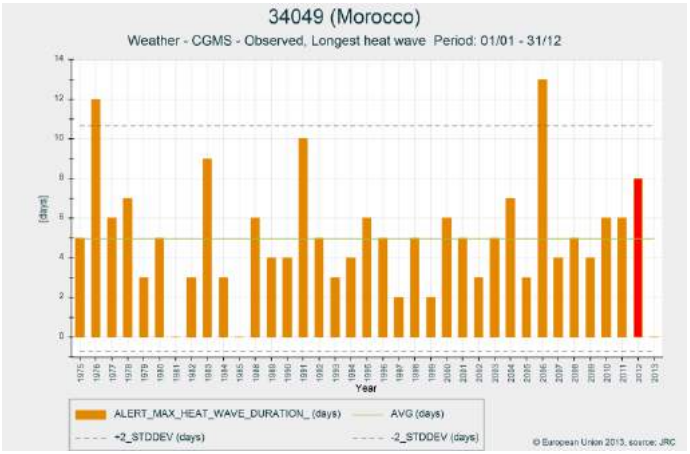


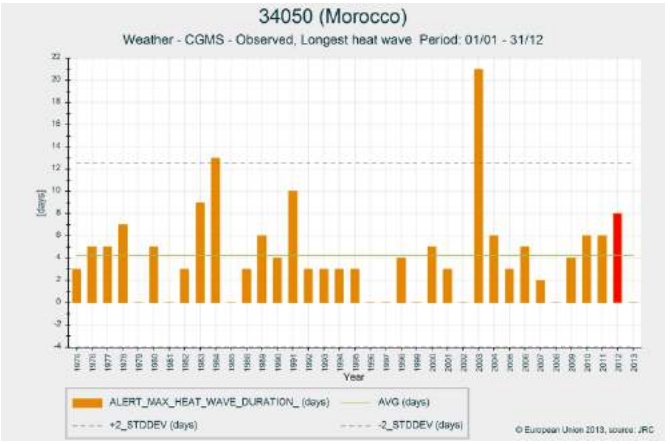


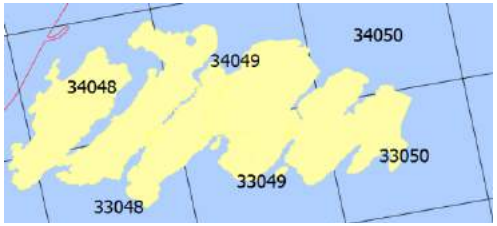
Maamora forest domain

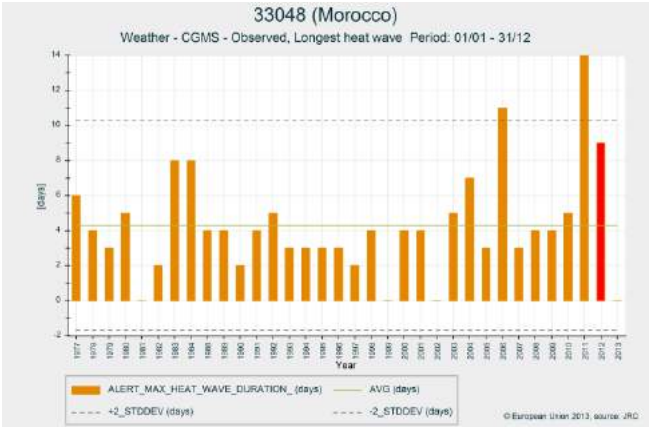
Longest heat wave (consecutive number of heat days)

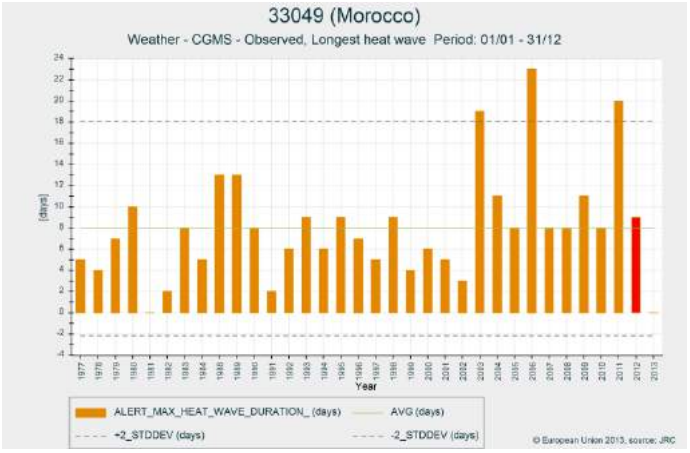


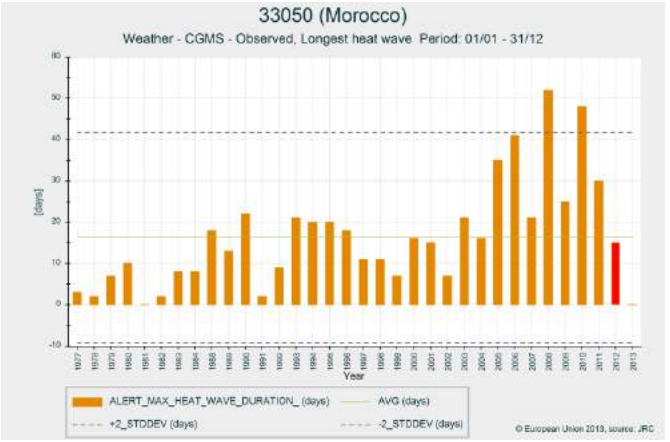






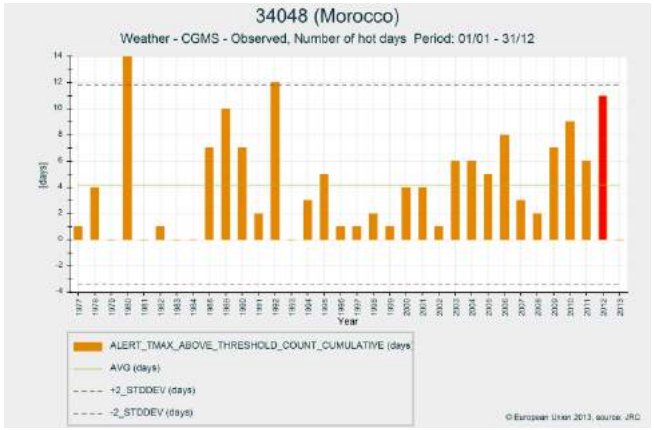


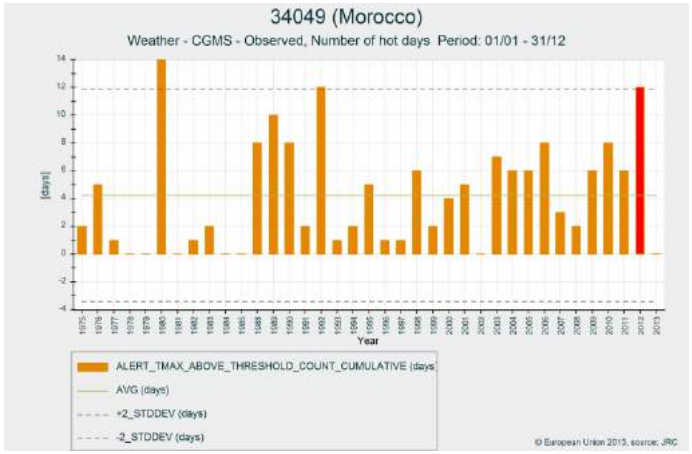


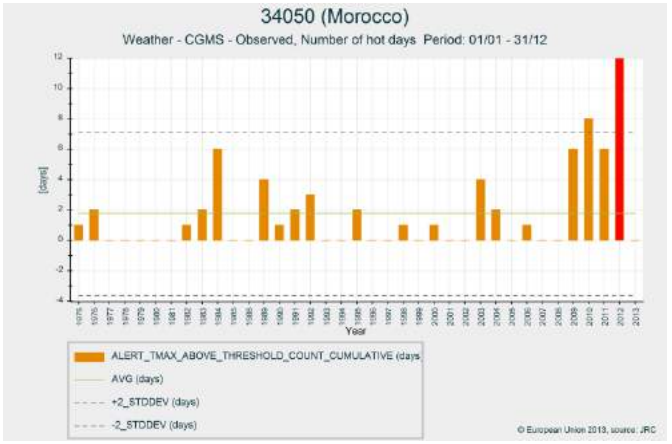


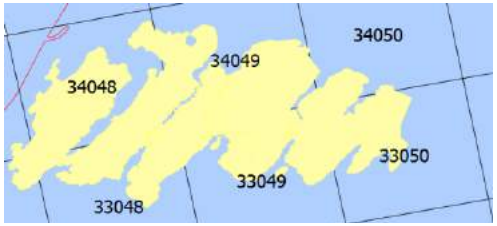
Maamora forest domain

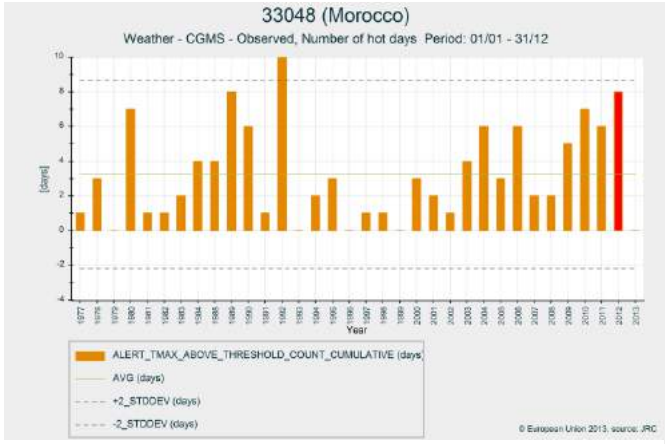
Number of days > 35°C

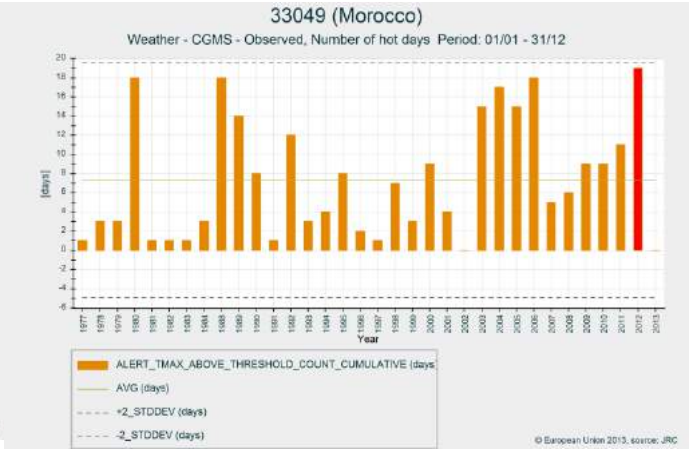


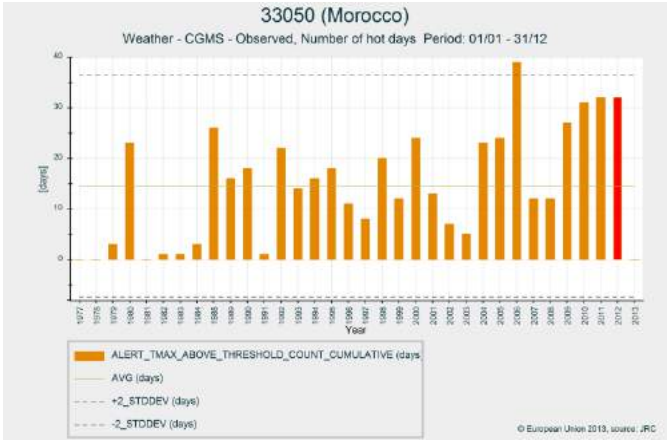






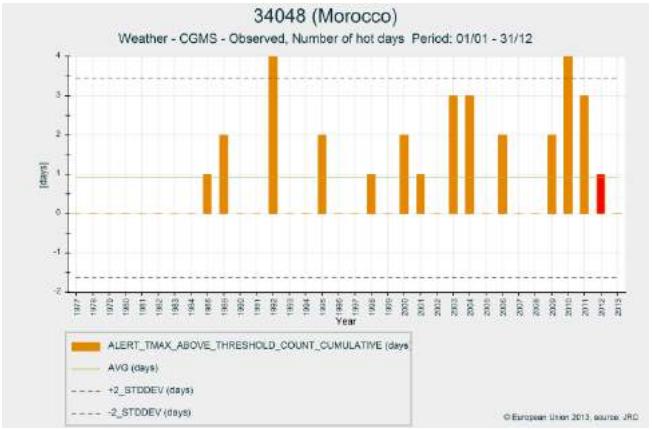


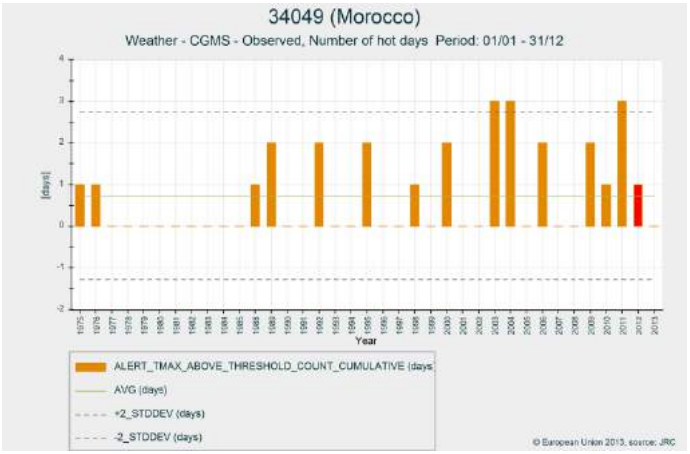


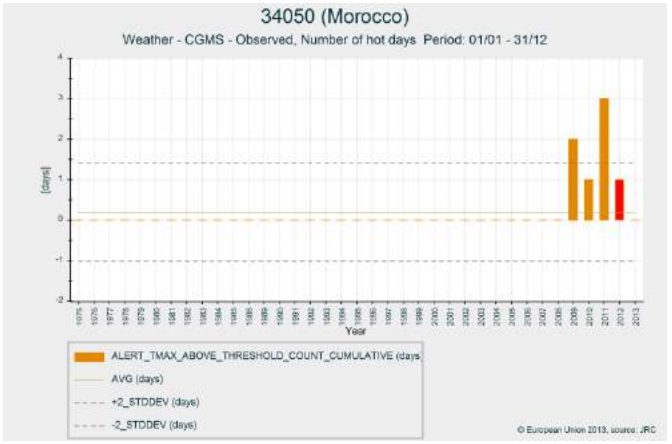


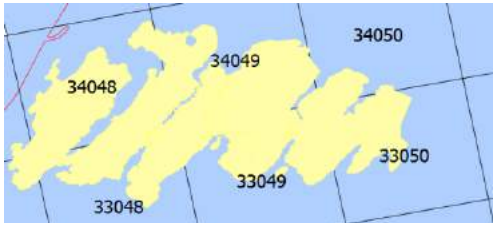
Maamora forest domain

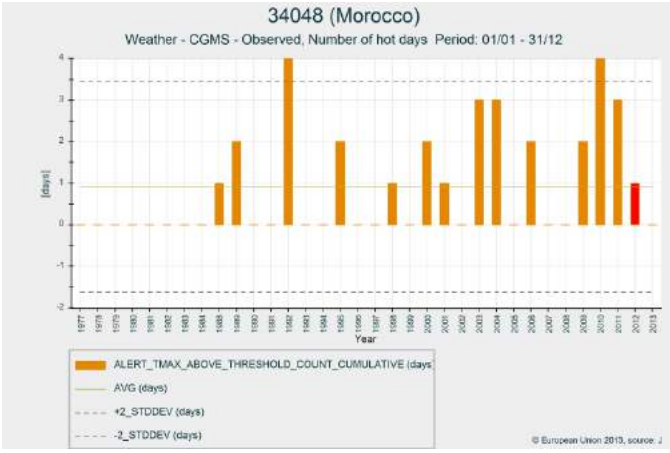
Number of days > 40°C

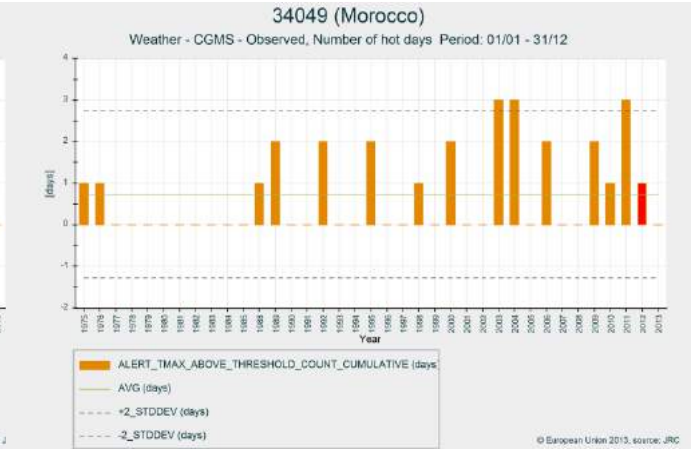


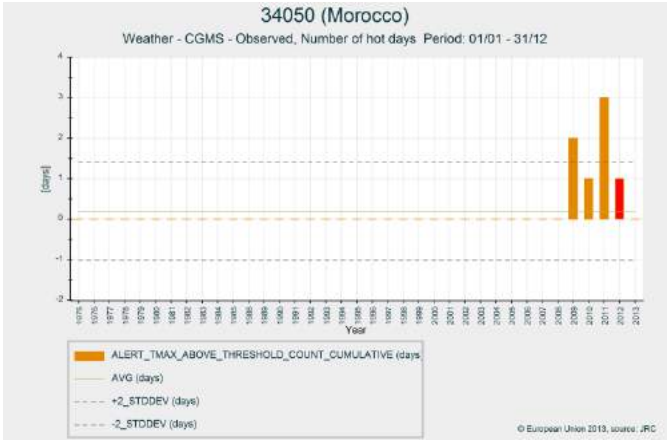










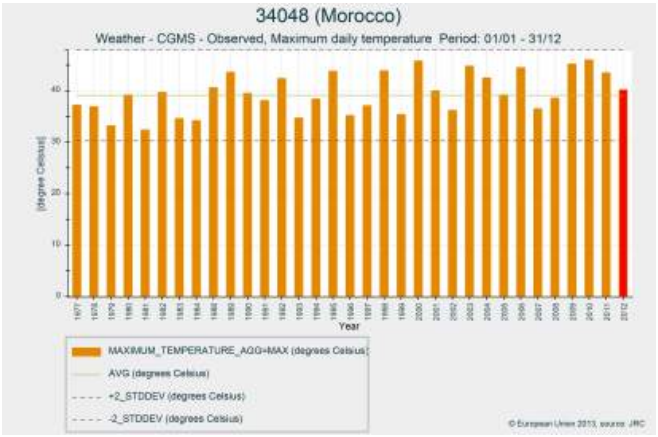


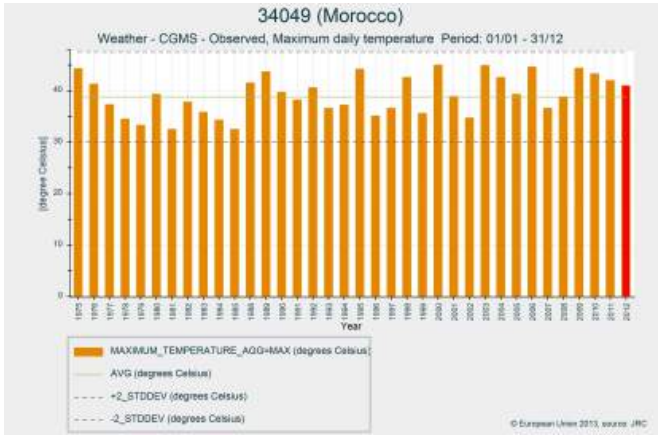
TEMPERATURE panel

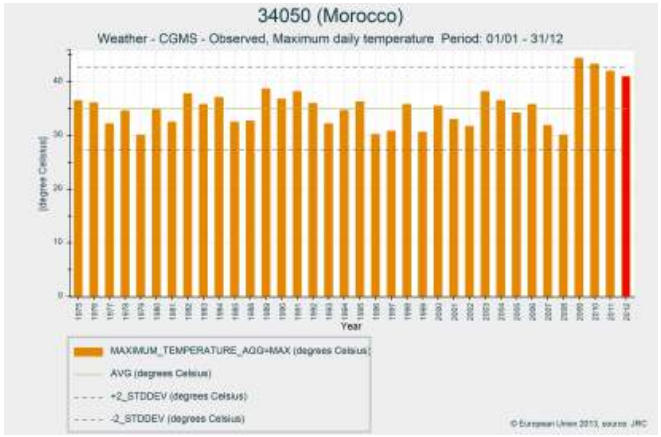
- Maximum temperature (daily average)
- Standard maximum temperature
- Minimum temperature (daily average)
- Standard minimum temperature
- Average temperature (daily average)
- Standard average temperature

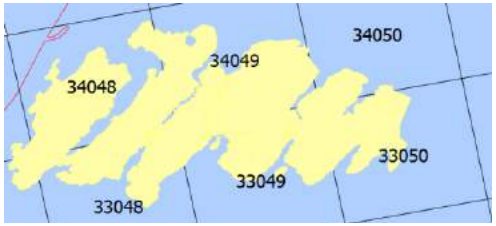
Maamora forest domain

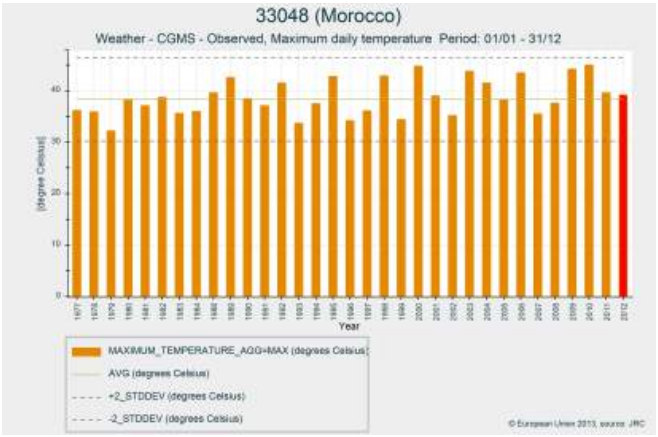
MAXIMUM temperature (daily average)

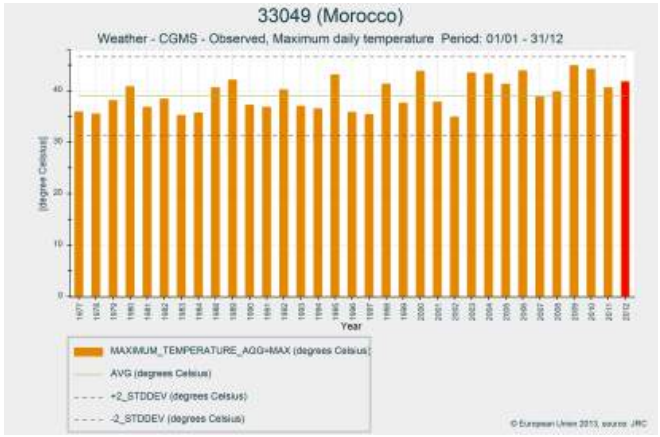


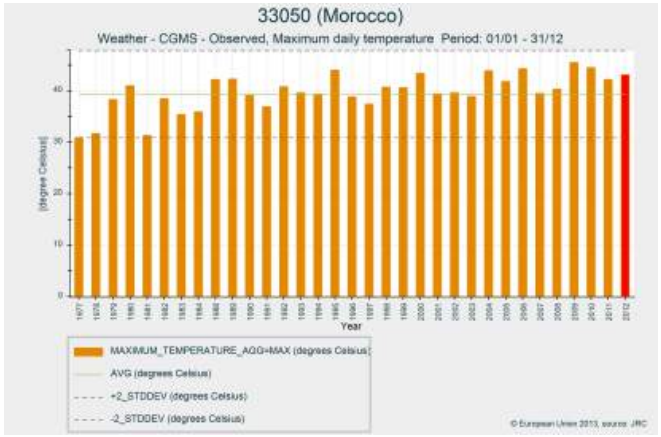






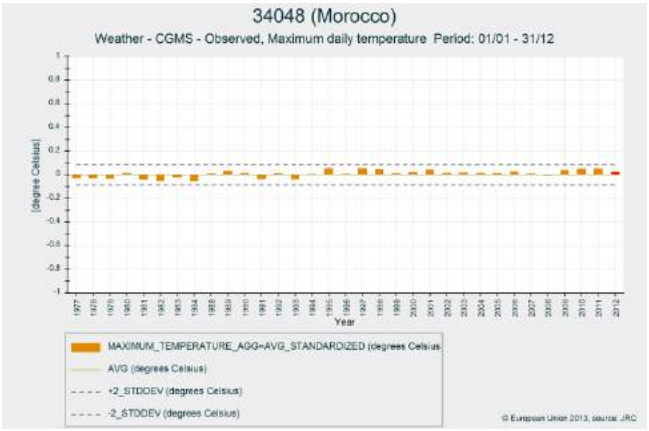


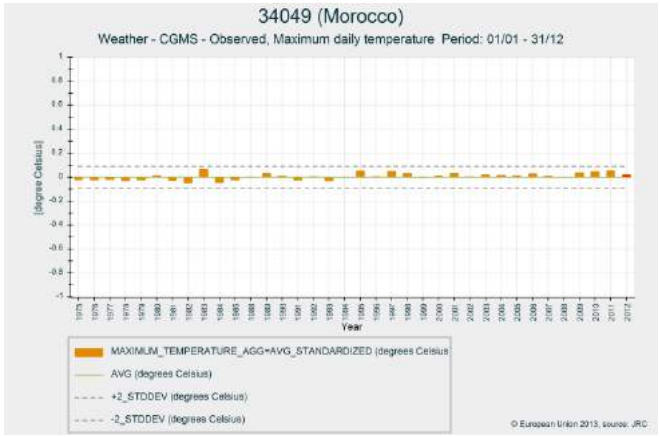


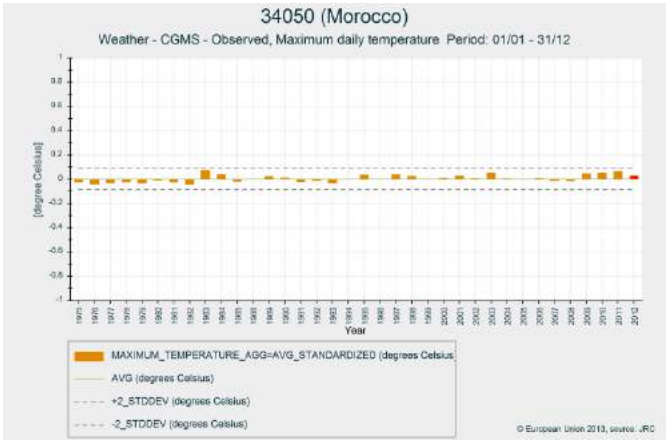


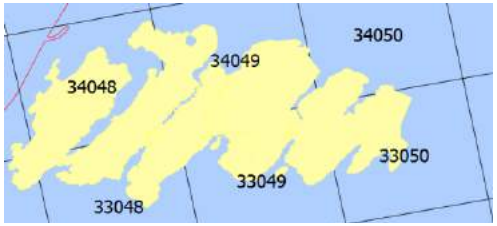
Maamora forest domain

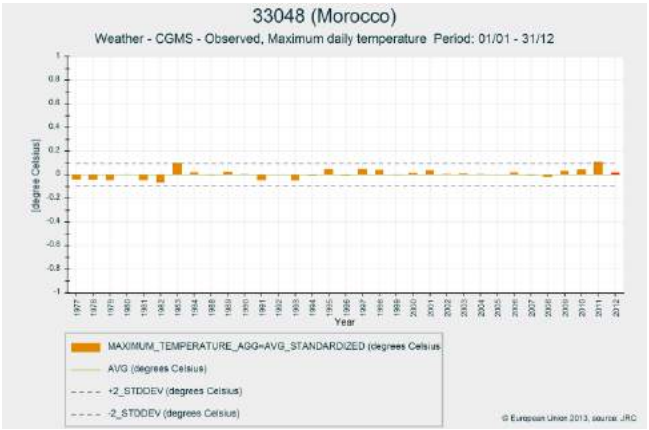
Standard T. MAX index

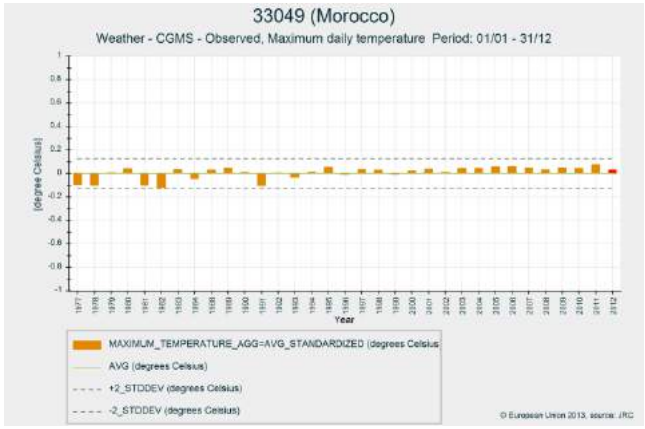


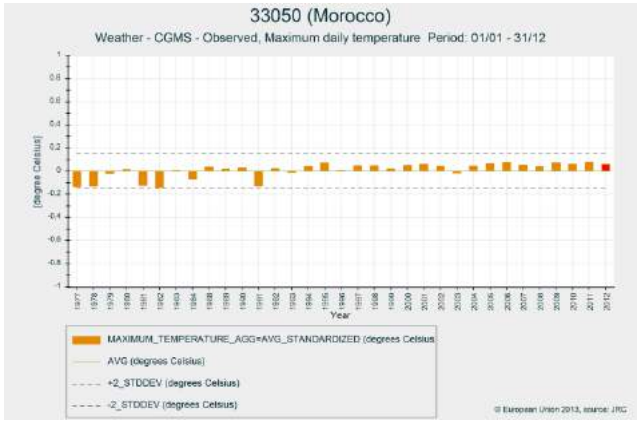






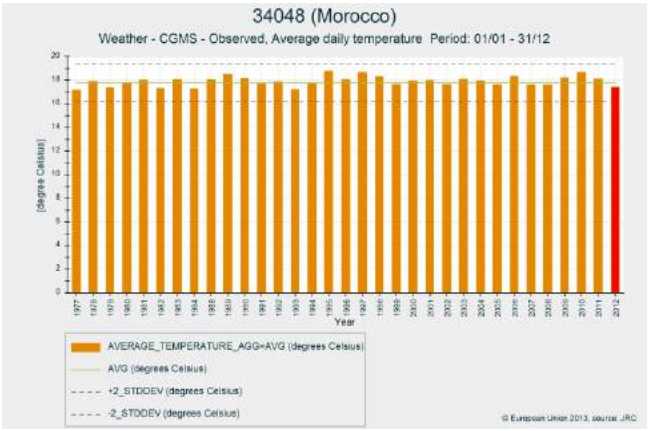


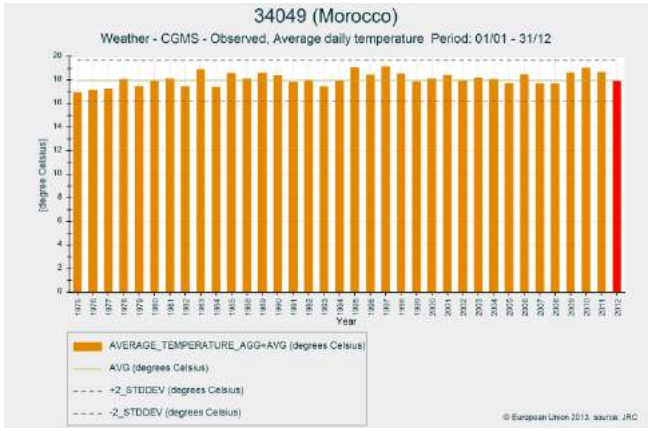


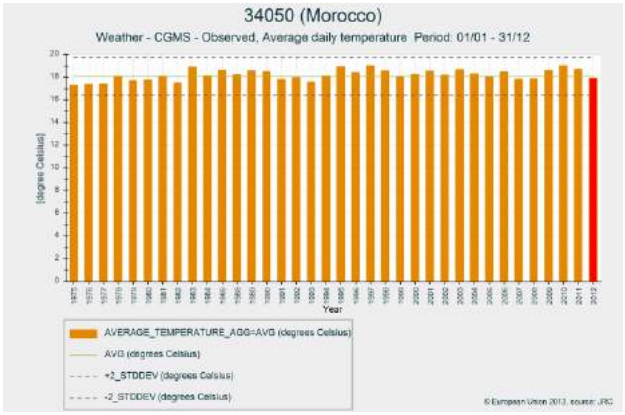


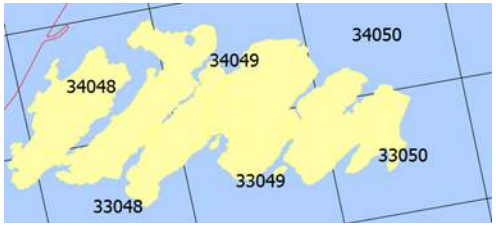
Maamora forest domain

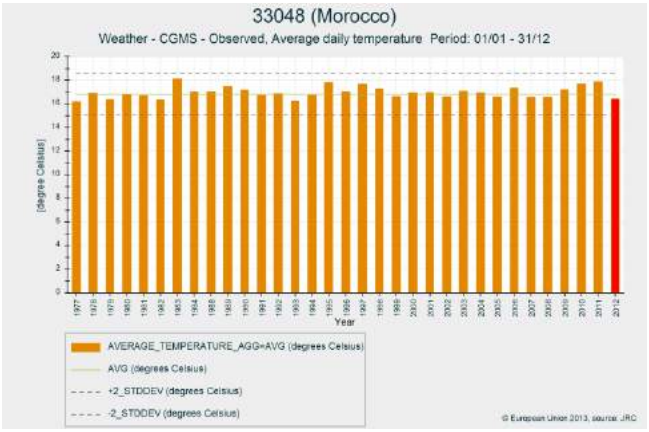
MINIMUM temperature (daily average)

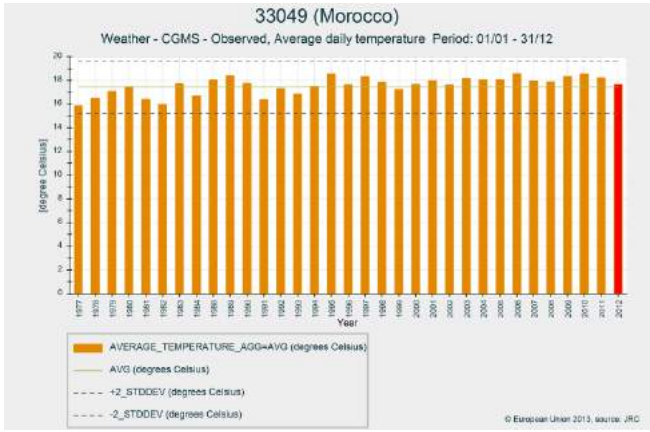


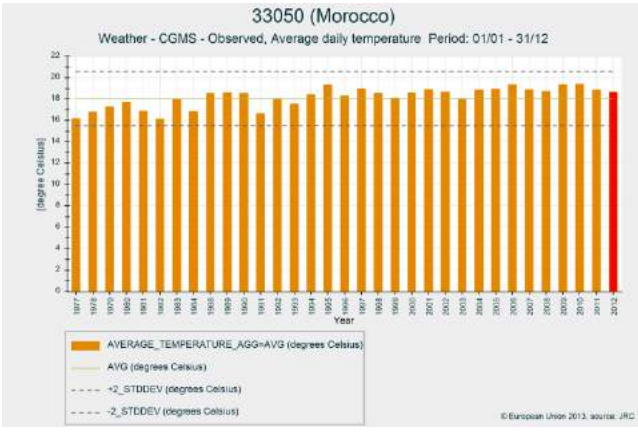






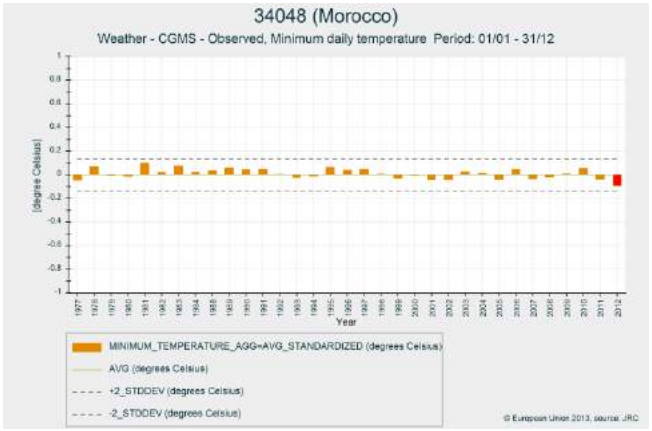


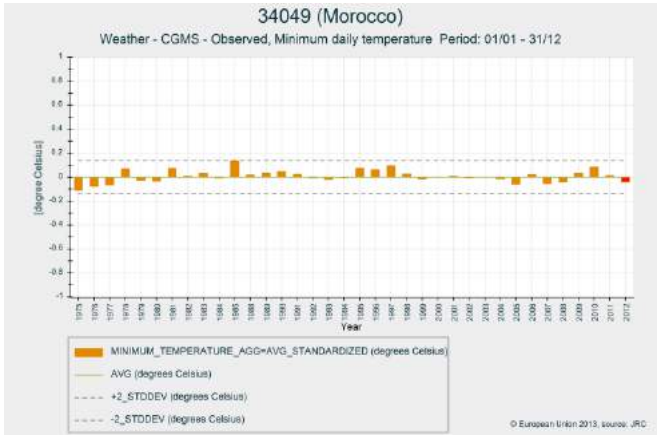


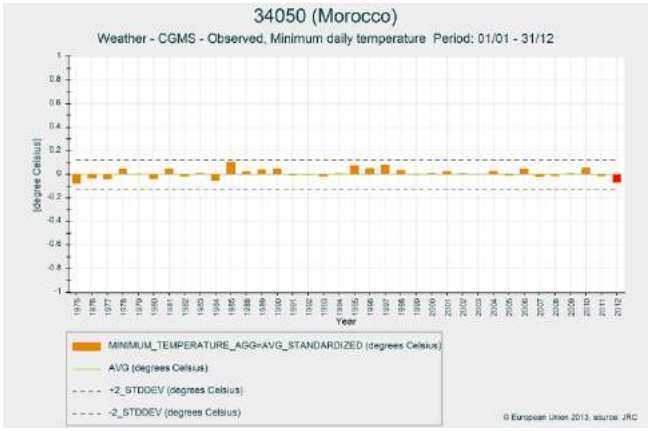


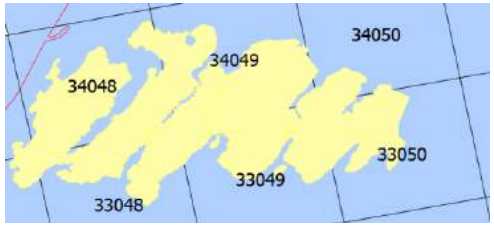
Maamora forest domain

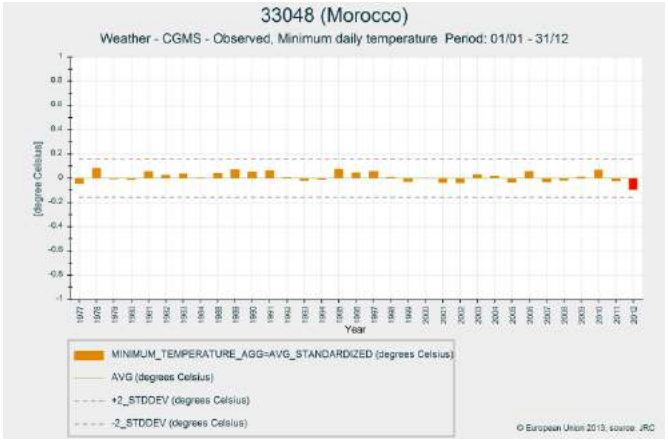
Standard T. MIN index

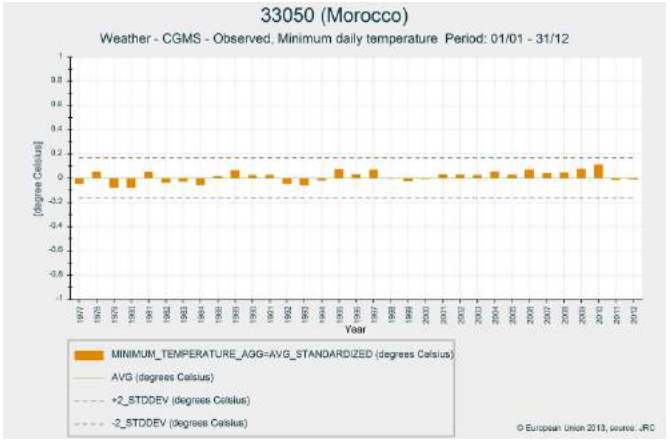






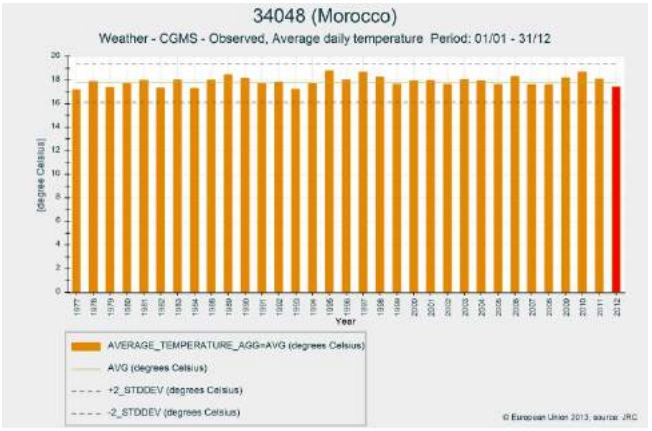


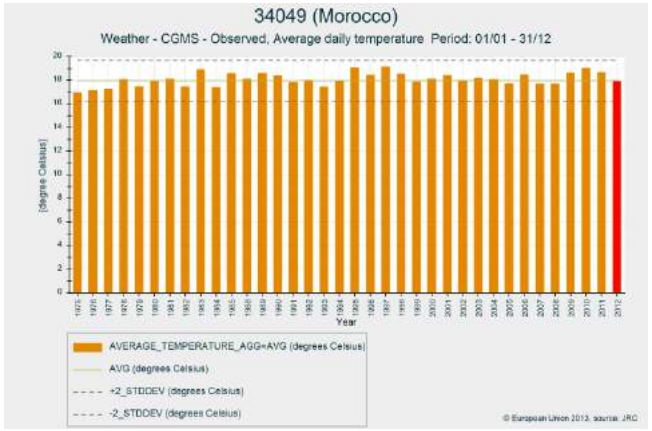


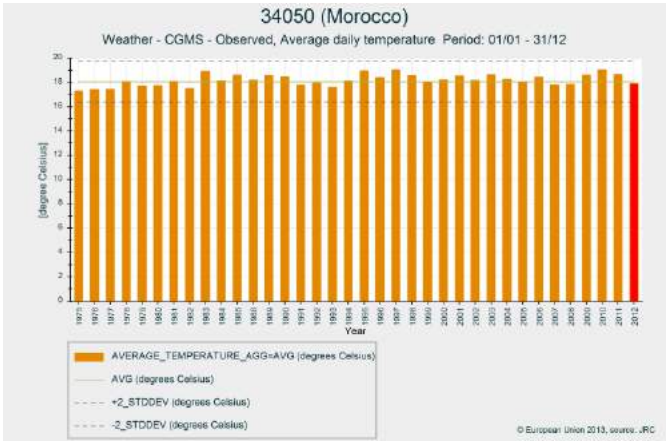


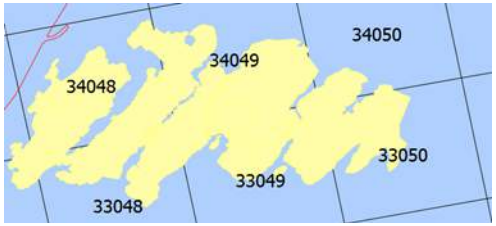
Maamora forest domain

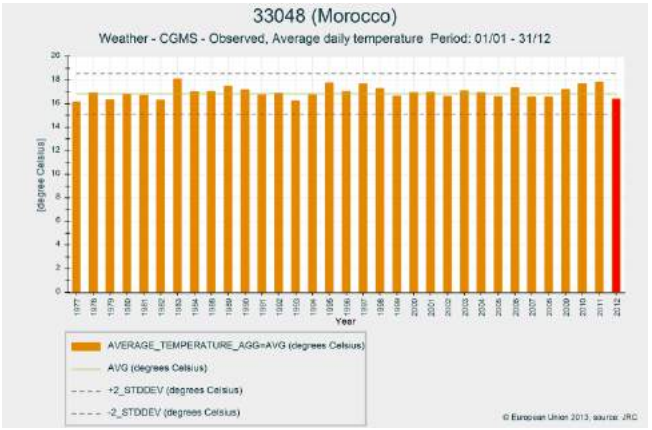
MEAN temperature (daily average)

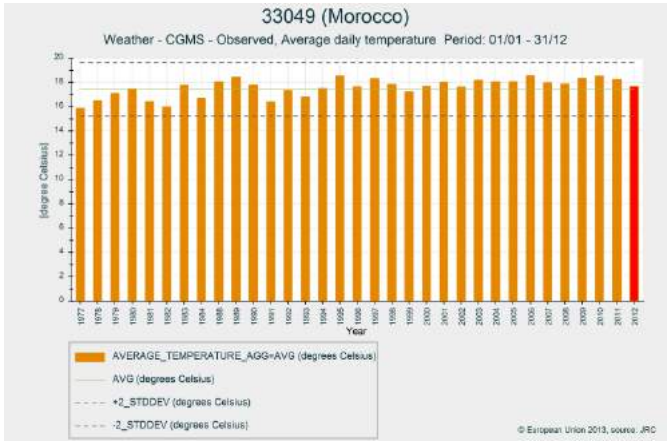


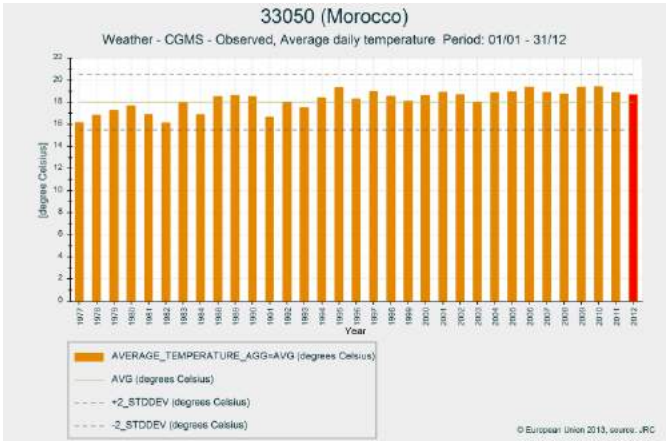












Maamora forest domain
Standard T. MEAN index