

# Automated temporal NDVI analysis over the Middle East for the period 1982 - 2010

**Hristo Dobrev Tomov**

---

2016  
Department of  
Physical Geography and Ecosystem Science  
Centre for Geographical Information Systems  
Lund University  
Sölvegatan 12  
S-223 62 Lund  
Sweden



Hristo Dobrev Tomov (2016). Automated temporal NDVI analysis over the Middle East for the period 1982 – 2010  
Master degree thesis, 30/ credits in Master in Geographical Information Science  
Department of Physical Geography and Ecosystem Science, Lund University

# Automated temporal NDVI analysis over the Middle East for the period 1982 – 2010

---

Hristo Dobrev Tomov

Master Degree Thesis in Geographic Information Systems

Supervisor:

Sadegh Jamali

Lund University

Department of Physical Geography and Ecosystem Science  
Lund University, 2016

## Abstract

The NDVI time-series consist of trend, season and noise. Changes in the season component are related to climate factors and they happen gradually over long period of time. The changes in the trend component are often due to human activities, fires and etc. This paper implements two algorithms (PolyTrend and DBEST) in R language, in order to examine the vegetation changes in the Middle East and to give more possibilities in the hands of the remote sensing communities. DBEST can analyse the gradual and the abrupt changes by decomposing the data, while PolyTrend classifies the inter-annual change between the picks of the green season.

PolyTrend and DBEST were adapted for R language environment. Two additional algorithms were developed to apply both algorithms over NDVI3g data set of the Middle East. A third algorithm discovered the affected land-cover through an overlay analysis by the use of the UMD land-cover classification data set.

PolyTrend showed linear (4%), quadratic (2%) and cubic (3%) trends. The different trend types were often found to be grouped in clusters. The largest clusters of trends were found near the south-eastern corner of the Arabian Peninsula and in the central regions of Saudi Arabia. More than 10% of all mixed forests were affected by these trends, most of which were in negative direction.

DBEST showed that 1% of the vegetation experienced a higher magnitude of change. Clusters of these changes were mainly located in the south-eastern and the western part of Turkey, the northern regions of Iraq and Syria, as well as along the coastlines of the Black Sea and the Caspian Sea. The changes were mainly related to the cropland and the grassland and were more in positive directions.

The study demonstrated the potential of PolyTrend and DBEST in R language for the remote sensing. It concludes that probably climatic factors affected the forests in Turkey and Iran. The high magnitude of changes of the cropland and grassland indicates that in some regions the agriculture expanded, while in others it declined.

**Keywords:** Physical Geography and Ecosystem analysis, GIS, Change detection, Satellite imagery, NDVI Time series, Trend analysis, PolyTrend, DBEST, R language, Middle East.

# Contents

<b>Abstract .....</b>	<b>iv</b>
<b>Contents .....</b>	<b>v</b>
<b>List of Figures .....</b>	<b>vii</b>
<b>List of Tables .....</b>	<b>viii</b>
<b>List of Abbreviated Terms .....</b>	<b>ix</b>
<b>1. Introduction .....</b>	<b>1</b>
1.1 Earth Observations from Space and AVHRR.....	1
1.2 NDVI and Temporal Trend Analysis.....	2
1.3 The R Project for Statistical Computing.....	4
<b>2. Research Objectives.....</b>	<b>5</b>
2.1 Problem statement.....	5
2.2 Specific aims.....	5
<b>3. Study Area.....</b>	<b>7</b>
3.1 The Middle East Conception .....	7
3.2 Geography.....	7
3.3 Climate.....	8
3.4. Land Cover .....	10
3.5 Vegetation change in the Middle East .....	10
<b>4. Literature review of common methods in NDVI time series analysis.....</b>	<b>13</b>
4.1 PolyTrend .....	13
4.2 DBEST.....	13
4.3 BFAST.....	15
4.4 LandTrendr .....	15
4.5 TIMESAT .....	16
<b>5. Data .....</b>	<b>17</b>
5.1 GIMMS NDVI3g.....	17
5.2 GIMMS NDVI3g Data Description.....	17
5.3 UMD Land Cover Classification .....	18
5.4. Global Administrative Areas .....	19
<b>6. Methods .....</b>	<b>21</b>
6.1 Overview.....	21
6.2 Data preparation.....	23
6.3 Data processing.....	24
6.4 PolyTrend in R language .....	25
6.5 DBEST in R language.....	27
6.6 Outputs.....	30
6.7 Land-cover/use overlay analysis.....	30
<b>7. Results.....</b>	<b>33</b>
7.1 Results computed by PolyTrend.....	33

7.1.1	<i>Trend types of the green season in the Middle East</i> .....	33
7.1.1.1	Linear trends .....	34
7.1.1.2	Quadratic trends .....	34
7.1.1.3	Cubic trends .....	35
7.1.2	<i>Direction of the growing season in the Middle East</i> .....	36
7.2	Results computed by DBEST .....	39
7.2.1	<i>Magnitudes and types of the greatest temporal NDVI changes</i> .....	39
7.2.2	<i>Beginning of the greatest temporal NDVI changes</i> .....	42
7.2.3	<i>Duration of the greatest temporal NDVI changes</i> .....	44
<b>8.</b>	<b>Discussion</b> .....	<b>47</b>
8.1	Temporal NDVI trends without vegetation .....	47
8.2	Major patterns and relationships .....	47
8.3	Automated temporal NDVI analysis .....	49
8.4	Sources of errors .....	49
8.5	Strengths and weaknesses of the study .....	50
8.6	Future work .....	50
<b>9.</b>	<b>Conclusions</b> .....	<b>51</b>
<b>10.</b>	<b>References</b> .....	<b>53</b>
<b>Appendix A.</b>	<b>Selected outputs from the MELCA algorithm.</b> .....	<b>59</b>
<b>Appendix B.</b>	<b>Load region function.</b> .....	<b>67</b>
<b>Appendix C.</b>	<b>Read NDVI3g function.</b> .....	<b>69</b>
<b>Appendix D.</b>	<b>META PolyTrend algorithm.</b> .....	<b>71</b>
<b>Appendix E.</b>	<b>META DBEST algorithm.</b> .....	<b>77</b>
<b>Appendix F.</b>	<b>MELCA algorithm.</b> .....	<b>83</b>
<b>Appendix G.</b>	<b>PolyTrend algorithm.</b> .....	<b>91</b>
<b>Appendix H.</b>	<b>DBEST algorithm.</b> .....	<b>95</b>

## List of Figures

Figure 3.1: Common western delineation of the “Middle East” (Culcasi, 2012). .....	7
Figure 3.2: Mean annual rainfall (mm) in the countries in the Middle East and North Africa, Source: AQUASTAT-FAO, 2008. ....	9
Figure 3.3: Land cover in the “Middle East” 1981 - 1994. The land cover data used in the map is created from is generated from University of Maryland Department of Geography (UMD) by the use of imagery from the AVHRR satellites acquired between 1981 and 1994 (Hansen, 1998). ....	10
Figure 6.1: Flow chart of META PolyTrend (implemented in R). The dashed rectangle indicates the wrapper function. ....	22
Figure 6.2: Flow chart of META DBEST (implemented in R). The dashed rectangle indicates the wrapper function. ....	22
Figure 6.3: Flow chart of PolyTrend. Source: Jamali et al., 2014. ....	26
Figure 6.4: Example of a graphical output by PolyTrend algorithm (insignificant cubic trend). ....	27
Figure 6.5: Flow chart of DBEST. Source: Jamali et al., 2015. ....	28
Figure 6.6: An example of two graphical outputs by DBEST algorithm in a change detection mode. ....	29
Figure 6.7: Flow chart of MELCA algorithm (implemented in R). ....	31
Figure 7.1: NDVI trend types in the Middle East (1982 – 2010). ....	33
Figure 7.2: NDVI trend slopes in Middle East (1982 – 2010). ....	37
Figure 7.3: Magnitude of the greatest NDVI change in the Middle East (1982 – 2010). ....	39
Figure 7.4: Type of the greatest NDVI change in the Middle East (1982 – 2010). ....	42
Figure 7.5: Start of the greatest NDVI change in the Middle East (1982 – 2010). ....	43
Figure 7.6: Duration of the greatest NDVI change in the Middle East (1982 – 2010). ....	44
Figure A.1: Linear NDVI trends and land cover in the Middle East (1982 – 2010). ....	59
Figure A.2: Quadratic NDVI trends and land cover in the Middle East (1982 – 2010). ....	60
Figure A.3: Cubic NDVI trends and land cover in the Middle East (1982 – 2010). ....	61
Figure A.4: Positive NDVI trends and land cover in the Middle East (1982 - 2010). ....	62
Figure A.5: Negative NDVI trends and land cover in the Middle East (1982 - 2010). ....	63
Figure A.6: Magnitude of NDVI change value -0.2 - -0.1 and land cover in the Middle East (1982 – 2010). ....	64
Figure A.7: Magnitude of NDVI change value 0.1 – 0.2 and land cover in the Middle East (1982 – 2010). ....	65

## List of Tables

Table 1.1: AVHRR/3 Channel Characteristics, Source: NOAA/SIS. ....	2
Table 5.1: Geographical coordinates of the grid, where 1/24 is half of the pixel size.....	18
Table 5.2: GIMMS dataset code values. ....	18
Table 5.3: GIMMS NDVI3g flag values and their descriptions.....	18
Table 5.4: Code values of the UMD land-cover data set (8 km pixel resolution) and class-specific accuracies (De Fries et al., 1998). ....	19
Table 7.1: Affected land-cover by linear NDVI trends in the Middle East (1982 – 2010). ....	34
Table 7.2: Affected land-cover by quadratic NDVI trends in the Middle East (1982 – 2010).....	35
Table 7.3: Affected land-cover by cubic NDVI trends in the Middle East (1982 – 2010).....	36
Table 7.4: Affected land-cover by positive NDVI trends in the Middle East (1982 – 2010).....	38
Table 7.5: Affected land-cover by negative NDVI trends in the Middle East (1982 – 2010).....	38
Table 7.6: Affected land-cover by NDVI change between -0.2 and -0.1 units in the Middle East (1982 – 2010). ....	40
Table 7.7: Affected land-cover by NDVI change between 0.1 and 0.2 units in the Middle East (1982 – 2010). ....	40
Table 7.8: Affected land-cover by NDVI change between 0.2 and 1 unit in the Middle East (1982 – 2010). ....	41
Table A.1: Linear NDVI trends and land cover in the Middle East (1982 – 2010).....	59
Table C.2: Quadratic NDVI trends and land cover in the Middle East (1982 – 2010). ....	60
Table A.3: Cubic NDVI trends and land cover in the Middle East (1982 – 2010). ....	61
Table A.4: Positive NDVI trends and land cover in the Middle East (1982 - 2010).....	62
Table A.5: Negative NDVI trends and land cover in the Middle East (1982 - 2010). ....	63
Table A.6: Magnitude of NDVI change value -0.2 - -0.1 and land cover in the Middle East (1982 – 2010). ....	64
Table A.7: Magnitude of NDVI change value 0.1 – 0.2 and land cover in the Middle East (1982 – 2010). ....	65



## List of Abbreviated Terms

<b>AVHRR</b>	Advanced Very High Resolution Radiometer
<b>BFAST</b>	Breaks For Additive Seasonal and Trend
<b>BIC</b>	Bayesian Information Criterion
<b>CRAN</b>	Comprehensive R Archive Network
<b>DBEST</b>	Detecting Breakpoints and Estimating Segments in Trend
<b>DMSP</b>	Defense Meteorological Satellite Program
<b>ESA</b>	European Space Agency
<b>ESSA</b>	Environmental Science Services Administration
<b>EUMETSAT</b>	European Organisation for the Exploitation of Meteorological Satellites
<b>EVI</b>	Enhanced Vegetation Index
<b>GADM</b>	Global Administrative Areas
<b>GIMMS</b>	Global Inventory Modeling and Mapping Studies
<b>ITOS</b>	Improved TOS
<b>LandTrendr</b>	Landsat-based detection of Trends in Disturbance and Recovery
<b>MAD</b>	Mean Absolute Deviation
<b>MELCA</b>	Middle East Land-Cover Analysis
<b>META DBEST</b>	Middle East Temporal Analysis with DBEST
<b>META PolyTrend</b>	Middle East Temporal Analysis with PolyTrend
<b>MLC</b>	Maximum Likelihood Classifier
<b>MODIS</b>	Moderate Resolution Imaging Spectroradiometer
<b>MOSUM</b>	Moving Sum
<b>NASA</b>	National Aeronautics and Space Administration
<b>NDVI3g</b>	third generation GIMMS NDVI
<b>NDVI</b>	Normalized Difference Vegetation Index
<b>NOAA</b>	National Oceanic and Atmospheric Administration
<b>NOAASIS</b>	NOAA Satellite Information System
<b>OLS</b>	Ordinary Least-Squares
<b>POES</b>	Polar Operational Environmental Satellites
<b>POES</b>	Polar Orbiting Environmental Satellites
<b>RMSE</b>	Root Mean Squared Error
<b>SPDF</b>	Spatial Polygon Data Frame
<b>SSM/I</b>	Special Sensor Microwave /Imager
<b>STL</b>	Seasonal-Trend Decomposition Procedure Based on Loess
<b>SVM</b>	Support Vector Machine
<b>TIROS</b>	Television Infrared Observation Satellite Program
<b>TOS</b>	TIROS Operational System
<b>UMD</b>	University of Maryland Department of Geography
<b>UN</b>	United Nations
<b>VHRR</b>	Very High Resolution Radiometer



# 1. Introduction

## *1.1 Earth Observations from Space and AVHRR*

On August 14, 1959, the U.S. Explorer VI Earth satellite took the first orbital images of the Earth (ESA, 2009). These crude images of our planet gave the start of constant satellite programs for Earth observation that continue to extend nowadays. At the beginning, sophisticated high-resolution camera systems were used for a first time by the CORONA Program (1958–1972) to map the U.S. and some other places. In the very same year that the CORONA program ended, NASA launched the first satellite dedicated to the remote sensing of the Earth's environment, which later was renamed to Landsat-1 (Khorram et al., 2005).

One of the longest operating satellite systems for Earth observations is the Advanced Very High Resolution Radiometer (AVHRR) that is mostly carried on the Polar Orbiting Environmental Satellites (POES). The beginning of this program can be traced back to 1960, when a series of experimental satellites were launched by NASA and the Environmental Science Services Administration (ESSA). In 1966, both agencies were able to provide regular global meteorological data from the TIROS Operational System (TOS) for a first time.

Later, in the 70s, ESSA merged into the National Oceanic and Atmospheric Administration (NOAA), which developed the Improved TOS (ITOS) and the Very High Resolution Radiometer (VHRR). The last satellite from the ITOS series was launched in 1976. TIROS-N, a satellite from the next generation, was equipped with the Advanced Very High Resolution Radiometer (AVHRR) and changed from analog data to a fully digital system (Hastings et al., 1992).

Nowadays, from TIROS-N satellite series NOAA-15, NOAA-18 and NOAA-19 are remaining operational (NOAA, 2015). Up-to-date, in addition to those satellites the POES constellation includes METOP-A and METOP-B. The last two are the result of a joint initiative between NOAA and the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT). The Initial Joint Polar System (IJPS) aims to provide a global climate monitoring services, as well as meteorological and environmental data (EUMETSAT, 2015). As part of this initiative NOAA has provided AVHRR instruments for each of the METOP satellites. The third spacecraft (METOP-C) from this initiative is expected to go in orbit in 2017.

During the evolution of POES, AVHRR went through two major changes. In the beginning, during the TIROS-N series, AVHRR was a 4-channel radiometer. Later, with the launch of NOAA-7 satellite in 1981, an additional channel was added to the radiometer (AVHRR/2). Nowadays, each of the satellites from the POES constellation carries a 6 channels AVHRR/3 sensor, which was used for the first time on NOAA-15 satellite in 1998 (NOAASIS, 2015).

The advanced optical subsystem of AVHRR/3 sensor splits the light into 6 spectral bands. The first three are respectively in the red, near-infrared and mid-infrared spectrum, while the other three are in the thermal infrared region (NOAA, 2007).

**Table 1.1: AVHRR/3 Channel Characteristics, Source: NOAA/SIS.**

Channel	Resolution (km)	Wavelength (um)	Typical Use
1	1.09	0.58 - 0.68	Daytime cloud and surface mapping
2	1.09	0.725 - 1.00	Land-water boundaries
3A	1.09	1.58 - 1.64	Snow and ice detection
3B	1.09	3.55 - 3.93	Night cloud mapping
4	1.09	10.30 - 11.30	Night cloud mapping, sea surface temperature
5	1.09	11.50 - 12.50	Sea surface temperature

As visible in Table 1.1, there are two channels pointed as number 3 (A or B respectively). At a given time only one of the last two can be switched on, so in reality only 5 channels are transmitting.

## ***1.2 NDVI and Temporal Trend Analysis***

One of the studies that NASA initiated during the Landsat-1 program was to examine the vegetation seasonal greening or green wave effect in the Great Plains. This is a flat land region that virtually divides the entire North America from North to South. Due to the size and the location of the Great Plains, the data derived from Landsat-1 were with a wide range of solar zenith angle. In order to reduce the influence of the zenith angle Kriegler et al. (1969) and Rouse et al. (1974) introduced the Normalized Difference Vegetation Index (NDVI).

NDVI is the interaction between the visible red and near-infrared spectrum as reflected by vegetation. Those two spectra are involved in the unique properties of the healthy plants and the photosynthesis. Because of the process of photosynthesis, the plants strongly absorb the visible light, while at the same time they strongly reflect the near-infrared light in order to avoid overheating.

$$NDVI = (NIR - RED) / (NIR + RED) \quad (1)$$

NDVI can take values between -1.0 and +1.0, where the healthy vegetation is mostly in the diapason mainly from 0.3 to 0.8, and sometimes above when the density of green leaves is high. In contrast, the barren and the snow covered surfaces are represented by low or negative values. Another interesting aspect of the ratio that is used in the above formula (1) is that it reduces the noise in the imagery from variations not only in the solar illumination, but also from the cloud shadows.

The Enhanced Vegetation Index (EVI) is another vegetation index that improves NDVI by being more sensitive toward canopy variations (Huete et al., 2002). However, the theoretical version of EVI uses three bands – red, near-infrared and the blue band in order to correct for aerosol influences. This limits the usage of EVI on data derived from

AVHRR, since this sensor doesn't measure the blue band. Moreover, AVHRR is recording data continuously since 1981, a decade more than the Moderate-resolution Imaging Spectroradiometer (MODIS). A result of the efforts to extend the current MODIS EVI time-series to the historical AVHRR data is the development of the EVI2, which uses only two bands (Zhangyan et al., 2007). However, despite the creation of EVI2 and other indexes, NDVI remains an essential index for monitoring the vegetation, especially when it comes to long time-series studies on global or large spatial scale (Wang et al., 2004).

A number of studies have shown that the data collected from AVHRR contain useful information about the seasonal vegetation development. Analyses of long time-series of NDVI data, which is measured from AVHRR, are being used to find shifts in the spatial distribution of bio-climatic zones, land-use changes, disturbance in global cycles or other large-scale circulation patterns (Jönsson et al, 2002). Besides, many scientific studies point out AVHRR as the best data source for long-term NDVI studies.

The NDVI time-series can be decomposed in several components in order to be studied and analysed. Similar to other type of time-series data, as well as in NDVI time-series, these components are seasonal, trend and remainder. The trend itself can be described as linear or non-linear. One method that is often used in the analysis of a linear type of change is the Ordinary Least-Squares (OLS) regression. However, a number of studies suggest that OLS regression may not always be appropriate for analysing changes in NDVI time-series (Foody, 2003). Moreover, in reality the changes that can be described as linear, and therefore to be subject to OLS regression, are just a fraction compared to the non-linear ones (Lambin et al., 2007).

A study that maps the vegetation trends over the Sahel region by Jamali et al. (2014) presents a method, that addresses the linear and the non-linear changes. The algorithm (PolyTrend) used in the study tests and applies a polynomial fitting to the annual NDVI time-series. It classifies the trends behaviour by dividing the NDVI changes to linear, quadratic, cubic or concealed. Besides the trend type, the algorithm also outputs the trend direction, slop and significance (Jamali et al., 2014).

Another interesting aspect of the NDVI time-series analysis, is the detection of the temporal moment and the magnitude of the change in the trend component. Such changes often indicate some kind of vegetation disturbance due to disease, insect attack or other. Likewise, the abrupt changes in the time-series can be due to fires, deforestation or other critical event (Verbesselt et al., 2010). Lately some researches present methods that automatically detect the NDVI change types. However, most of them can be reproduced only for a specific territory, NDVI data or situation. The main reason for this obstacle is due to the limited number of settings that the user can control.

A study that improves the detection of gradual and abrupt changes in vegetation time series data by Jamali et al. (2015) presents a new method which is named Detecting Breakpoints and Estimating Segments in Trend (DBEST). The study applies the DBEST algorithm over Iraq in order to automatically map the change types, magnitude, beginning

and duration. DBEST gives the flexibility to be configured for a number of situations. Likewise, it allows the users to define the boundary between the signal and the noise.

Among the other known software solutions for temporal NDVI analysis are BFAST (chapter 4.3), LandTrendr (chapter 4.4) and TIMESAT (chapter 4.5). Each of these methods has their own approach and does not provide a universal solution for vegetation trend detection. TIMESAT extracts seasonal parameters related to the growing seasons, such as the time of the growing season and the seasonal amplitudes. LandTrendr is designed to capture short-duration events and to smooth long-term trends, however, it is designed only for forest monitoring from Landsat data. PolyTrend is most suitable for mapping long gradual changes and is especially useful for large data volumes. DBEST and BFAST have more generic change detection approach. Both algorithms are useful for estimating the characteristics of abrupt and long-duration changes, however, each has its own strengths. BFAST computes the confidence interval for the time of change and detects changes in the seasonal component, while with DBEST the user can control the process trend generalisation and change detection through a number of parameters.

### ***1.3 The R Project for Statistical Computing***

For the first time R language was presented in 1993 on the mailing news of the S language - a statistical programming language. Moreover, R is an implementation of S and it has later evolved to resemble it closely. It is freely available under the GNU General Public License for UNIX platforms, Windows and MacOS (R Foundation, 2015).

The R language development accelerated and became notable just five years later, mainly due to its active community (Ihaka, 1998). The Comprehensive R Archive Network (CRAN) is just one of the repositories that extend R through user-created packages. Moreover, the number of R user groups has dramatically increased since 2010 with the rise of the so called 'big data' applications. R-sig-geo is only one of these active groups, which is developing R functions and packages especially for analysis of geospatial data (Bivand et al., 2013). Up to the date of this text (28.03.2015), the CRAN package repository features 6455 available packages, which encompasses a wide variety of statistical techniques and graphical methods.

The interpreted nature of R language makes it easy to implement, test and modify algorithms. At the same time, besides the statistical tools, the R packages include spatial data, raster and satellite image processing algorithms (Goslee, 2011). R language allows the users to know what exactly the software is doing, but also to modify and extend it. Finally, the open source model of R allows the analysis of empirical and simulated data to be reproducible (Bivand et al., 2013).

## 2. Research Objectives

### 2.1 Problem statement

There is a limited number of open-source software for temporal NDVI analysis. Most of this software uses different techniques for trend analysis and change detection or is designed to be suitable for specific situations. The implementation of PolyTrend and DBEST in R language will provide alternative open-source tools for NDVI trend analysis to the remote sensing community. The combination of specific tools for analysis of long gradual changes and disturbances, the number of parameters that the user can control in DBEST, plus the interpreted nature of R language, and the possibility easy to implement or modify source code should give sufficient flexibility to the researchers.

In most of the Middle East the climate is arid. According to IPCC (2001) the vegetation in the arid zones is the most stable community among all, and the impact from the climate change may not be apparent for several decades. However, some of the countries in the region were subject to socio-political changes in the last decades. Between 1982 and 2010, different parts of the Middles East were constantly in armed conflicts, among which are: the Iran–Iraq War (1980–1988), the Gulf War (1990–1991), the civil wars in Yemen (1986, 1994, 2004–2014), the Iraq War (2003–2011), the conflicts in Lebanon (2006, 2007, 2008), the insurgencies in Saudi Arabia (1987, 1995, 2004–2014), the massacres in Syria (1982, 1986, 2004) and etc. It is interesting to see how all these instabilities affect the vegetation in the arid zones in the Middle East. In what directions were the long gradual and the more sudden changes in the region? The northern regions are subject to different climate zones and between 1982 and 2010 there were no wars in Turkey, as well as in other countries from the region. What were the changes there?

### 2.2 Specific aims

The current research has two main aims - to develop algorithms in R language for automated NDVI trend analysis and to present analysis of the NDVI trends from 1982 to 2010 in the Middle East.

The first aim includes the implementation of PolyTrend and DBEST algorithms in R language, as well as the development of three additional algorithms. Two of these additional algorithms apply PolyTrend and DBEST over the GIMMS NDVI3g data set, while the third algorithm performs an overlay analysis with the UMD land-cover classification data set. Each of the three algorithms stores the results as raster data sets in GeoTIFF format, maps in PDF format and statistical data in text files.

The second aim of the research is to summarize the obtained results and deliver information regarding the trends of the vegetation in the Middle East. This includes

information related to the trend types of the green season, the directions of these trends and their slope values. Likewise, information regarding the greatest temporal NDVI changes, such as: the magnitude of the changes, whether they were abrupt or gradual, the time in which the changes started and their duration. The summarized results will also show what proportion from each land-cover class was affected.



### 3. Study Area

#### 3.1 The Middle East Conception

The Middle East region includes Western Asia and the northeastern corner of Africa - the country Egypt. The other countries that are considered to be part of the region are: Bahrain, Cyprus, Egypt, Iran, Iraq, Israel, Jordan, Kuwait, Lebanon, Oman, Palestine, Qatar, Saudi Arabia, Syria, Turkey, United Arab Emirates and Yemen (Culcasi, 2012). Some countries and occasionally the United Nations (UN) are using the term Near East for the same region (Davison, 1960).



Figure 3.1: Common western delineation of the “Middle East” (Culcasi, 2012).

However, there is no single agreed description that gives exact definition of the region. The term was invented outside the region for political, economical and other reasons (Koppes, 1976). Moreover, sometimes only Egypt's Sinai region is considered a part of the Middle East, and sometimes the entire country Egypt. Likewise, some include Armenia, Azerbaijan and Georgia as part of the Middle East (Novikova, 2000). Furthermore, other countries that can be related to the Middle East are Libya, Tunisia, Algeria, Morocco from North Africa and Afghanistan and Pakistan from Asia (Özalp, 2011).

#### 3.2 Geography

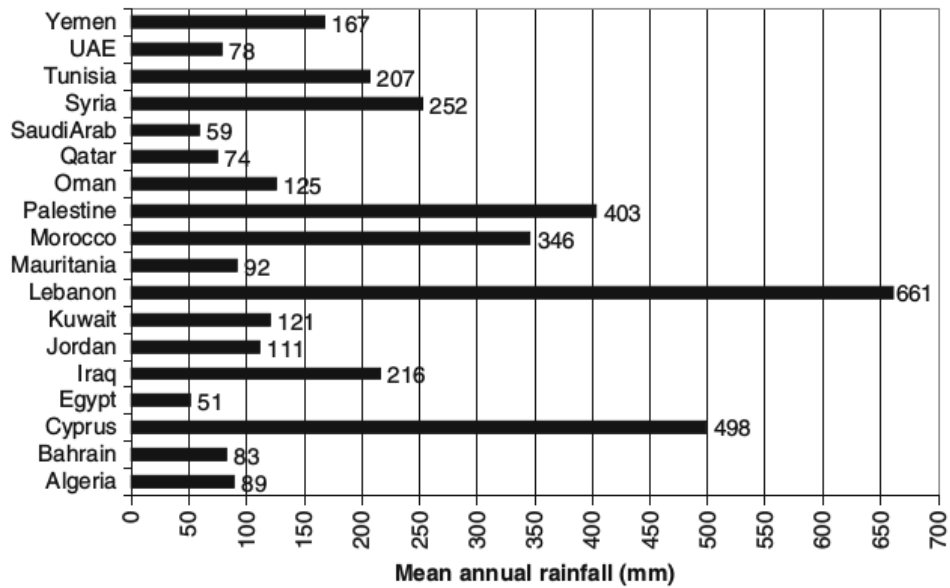
The Middle East region borders to the north with Eastern Europe and the Caucasus Mountains. In the east, it spreads to the Dasht-e Kavir and Dasht-e Lut deserts which connect it with Central Asia and further from there it borders South Asia. In the southwest, it is delimited by or sometimes includes the region of North Africa. Seven seas surround the Middle East region - the Black Sea, Aegean Sea, the Mediterranean Sea, the Red Sea, the Persian Gulf, and the Caspian Sea.

In the Middle East are located some of the largest sand deserts in the world. Near the central part of the region, over most of the Arabian Peninsula is located the Arabian Desert. It is the second largest desert in the world and it consists of several other deserts, among which are the Rub' al Khali sand desert. It is one of the largest continuous sand areas in the world and that is why it is also known as the Empty Quarter. The Arabian Desert includes also two other desert regions - the Syrian Desert and the An Nafud Desert. In addition to these deserts and the Iranian deserts in the north, the region adjoins even larger desert bodies. In the south, the Middle East meets and includes parts of the largest desert on the planet – the Sahara desert, as well as the Libyan Desert.

Despite this dominant desert flat topography, located in the region are also large areas of mountainous terrain. In the northern part of the region, in Turkey, the Pontus Mountains, the Taurus Mountains and Mount Ararat are located. Not so far from there, on the Europe-Asia boundary, the region meets another significant mountain range - the Caucasus Mountains. Then, in the south from the Caspian Sea it includes the Zagros Mountains, a mountain range with length of 1,500 km. The Asir Mountains and Hejaz Mountains are two other ranges of lower mountains that are located along the Red Sea. Besides, in North Africa, the region adjoins the Atlas Mountains, stretching over a length of over 2,500 km.

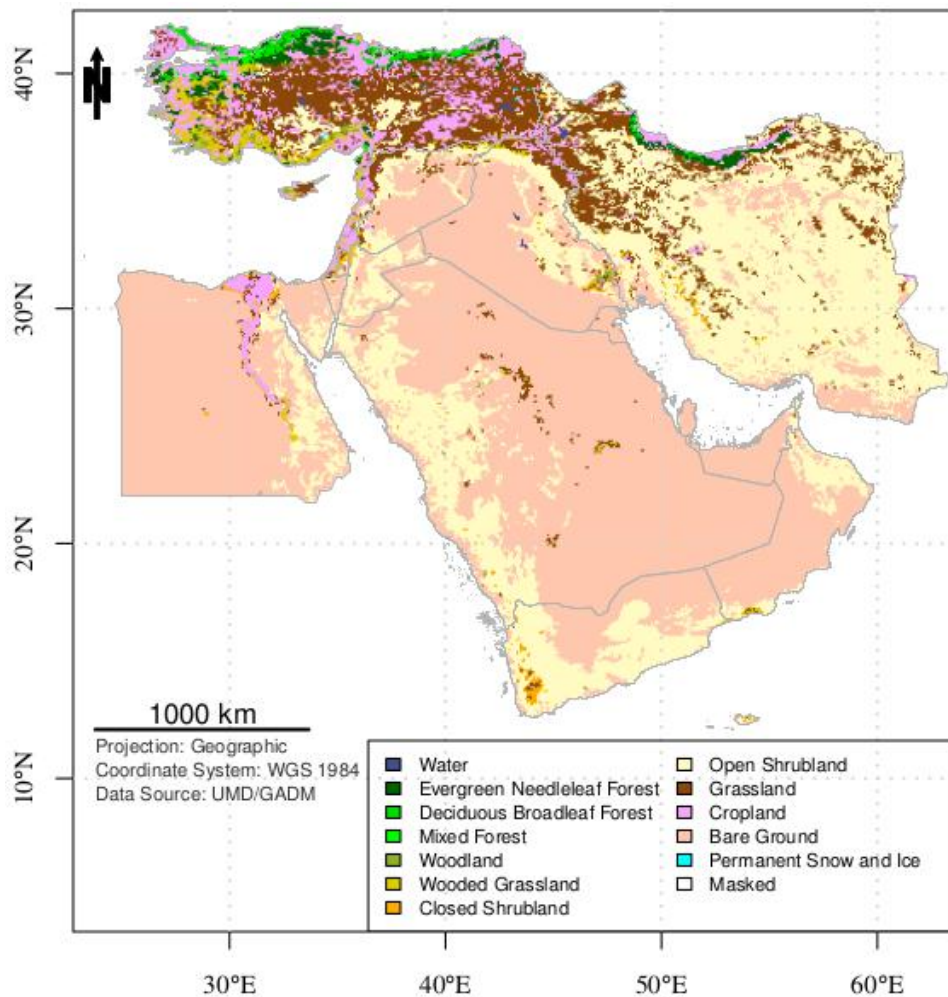
### ***3.3 Climate***

The Middle East region includes some of the hottest, driest and sunniest places on Earth. The average temperatures in the region in January vary from -10°C to 25°C, while in July they are from 20°C to more than 35°C (IPCC, 2001). Around two-thirds of the region is classified as hot arid climate. This climate zone spreads almost entirely uninterrupted in the south and central part of the Middle East. In the northern part of the Middle East, near the mountain ranges the climate is semi-arid. Permafrost zones can be seen in the high mountain ranges. This type of zones can also be found in the southeast parts of the region. Along the coast of the Mediterranean Sea the climate is classified as Mediterranean. This narrow zone has warm to hot, dry summers and mild to cool, wet winters.



**Figure 3.2: Mean annual rainfall (mm) in the countries in the Middle East and North Africa, Source: AQUASTAT-FAO, 2008**

Generally, the rainfalls in the Middle East are low, with relatively small exceptions of aridity. It is considered to be one of the driest regions in the world. As visible in Figure 4, among the shown countries Lebanon and Cyprus are exceptions, as well as other regions that are near the mountain ranges – like for example Morocco, where the higher rainfalls are due to the effect of the Atlas Mountains (Sivakumar, 2013). However, in most of the territories on the Arabian Peninsula and the countries in North Africa, the annual rainfall is less than 100 mm.



**Figure 3.3: Land cover in the “Middle East” 1981 - 1994. The land cover data used in the map is created from is generated from University of Maryland Department of Geography (UMD) by the use of imagery from the AVHRR satellites acquired between 1981 and 1994 (Hansen, 1998).**

### ***3.4. Land Cover***

Despite the absence of water in 75% of the region and the sand deserts, there are fertile valleys located near the major rivers. These are Tigris and Euphrates river systems in Western Asia, and the longest river in the world - Nile in northeastern Africa. They are important source for irrigation water in the region and the agricultural activity along them is well developed. Besides the agricultural land, in the vast floodplain of the Euphrates and Tigris are located huge permanent lakes, marshes, and forests. More forest areas are also located in the northern parts of the Middle East and in the mountain ranges. Moreover, 37.5% of all forest is located in Turkey (FAO, 2000). Likewise, grasslands and rangelands fields are located near the same regions.

### ***3.5 Vegetation change in the Middle East***

Al-Bakri et al. (2001) investigated the land-use change over the Badia Transition Zone in Jordan for the period from year 1953 to 1992. To do that, the authors of the study

use panchromatic aerial photography and merged satellite imagery from SPOT PAN and Landsat TM. The results from the research are land-use change analysis and maps for four sites in the transition zone. Their study concludes that the observed trends of land-use change are mainly: expansion of the urbanized area, a shift from rangeland into rain-fed cultivation, and an incremental transition of rain-fed cultivation and rangeland into irrigated fields.

Bitter & Sofer (2013) monitor the changing land-use patterns in three rural settlements along the coastal area in Israel. The data used in the study was mainly from historical air photographs as well as from recent field mapping survey. Based on data, the authors of the study created four land use maps for each settlement for the 1950s, 1970s, 1990s and 2009 periods. The land-use change analysis in the study points out two main trends: intensification of the agricultural land and expansion of built up areas. In the conclusions of this research the authors look for a link between the result, the socio-economic development and the national rural policy.

A study of the Euphrates river basin in Syria by Wafi Al-Fares (2013) investigates the historical land use/land cover by the use of remote sensing data. The study focuses on the agricultural irrigation projects during the last four decades, and maps the development of the irrigation areas. In order to do that, the author examines various remotely sensed data and searches for an effective approach to classify the historical and current land use/land cover. The main data used in the final results are derived from the LANDSAT missions as well as from the Terra satellite.

The study concludes that the change of the vegetation trends is due to an intensive agricultural expansion in the area. It presents a land cover change analysis that shows a dramatic increase by 148 % of the cultivated land from year 1975 to 2007, but also a decrease in the natural vegetation area and an increase in bare areas. Likewise, during the period of 32 years, 4.74 % of the land was changed from natural vegetated areas to bare areas, while 7.19% of the land was changed from bare areas to cultivated areas.

Guneroglu (2015) investigates the coastal changes and land use alteration on Northeastern part of Turkey in the period from 1984 to 2011. The author applies SVM algorithm over Landsat TM/ETM images in order to locate and classify the different land-use/land cover types in a coastal zone with length of 110 km. The four different classes that are defined in the study are: impervious surface, vegetation surface, soil-agriculture and water. Among the various conclusions from the research, the author points that for the study period of 27 years the impervious surface changed from 5.45% to 13.19% of the total area, while at same time the agricultural and soil areas proportionally decreased from 21% to 5.19 % . Guneroglu (2015) concludes that the process of urbanization has altered the land-use/land cover from soil and agriculture type to impervious surface.

A study by Gibson et al. (2015) investigates changes in agricultural land use/land cover in Iraq in the period between 2001 and 2012. The authors use the relation between the annual MODIS Land Cover Product and the years of drought in order to analyze how exactly the cropland changed over time. Likewise, the study connects the land use/land

cover changes with socio-political factors, the 2003 US-led military invasion and other ongoing domestic conflicts. Gibson et al. (2015) concludes that there is a significant impact on the cropland area and this change is related to the socio-political turmoil in the period. The authors point out two of the most affected regions: the Kurdish Autonomous Region, where the cropland area decreased an average of 16,479 ha per year since the Gulf War of 1991, and Central Iraq, where a decade later the cropland began to decrease by 18,581 ha per year.

## 4. Literature review of common methods in NDVI time series analysis

### 4.1 *PolyTrend*

The vegetation properties at a particular location are not always changing in gradual and linear manner with the time. That is why the simple linear regression is not sufficient to describe the changes in the temporal trends of NDVI data. In a study over the region of Sahel, Jamali et al. (2014) test and apply a polynomial fitting-based scheme to the annual GIMMS NDVI data for the period between the years 1982 and 2006.

Jamali et al. (2014) research method classifies the annual NDVI time-series' trends as linear, cubic, quadratic, concealed or no-trend. The difference between the last two is that the concealed trend can be either cubic or quadratic polynomial, but however the annual NDVI net change hasn't been significant at the end of the period. Whereas, the no-trend category includes simple linear trends that are below the significant slope coefficient.

Generally, the classification process of the PolyTrend scheme can be divided into 3 phases. Each phase tests a polynomial fitting of a lower-degree than the previous one and each phase evaluates if the linear coefficient is statistically significant. In the initial phase, a given temporal NDVI trend is tested and evaluated for a cubic polynomial fit. If the tests fail, a second phase begins, where the data is tested and evaluated for a quadratic polynomial fit. If the tests of any of the first two are passed, then the linear coefficient significance determines if the trend will be classified as concealed. Then, if the quadratic model fails to describe the trend, a linear regression model is tested. Finally, an evaluation of the linear coefficient determines if the trend will remain linear or will be classified as no-trend.

The results from the automated mapping of vegetation trends over the Sahel region show that 51.9% of the trends have statistically significant linear coefficient, and out of them the majority are linear trends (27.1%), then are the quadratic (3.4%) and cubic trends (3.2%). In each trend class, the positive trends are much more than the negative ones. Moreover, the study finds an interesting relation between the ratio of the linear and non-linear trends, and the different land cover types. Finally, the study notes that while PolyTrend has some advantages over other known methods, in some cases it may oversimplify the trend or miss short-term variations.

### 4.2 *DBEST*

Nowadays, a number of studies are focused over trend detection algorithms that are able to discriminate the abrupt and the non-abrupt changes in a time series data. Jamali et al. (2015) present a method for generalizing the vegetation trends or

distinguishing the type of changes in the vegetation trends. In the study, the Detecting Breakpoints and Estimating Segments in Trend (DBEST) algorithm is used to analyze GIMMS data set in order to map the types, the time and the magnitude of the vegetation changes in Iraq for the period from year 1982 to 2006. Likewise, it is tested against the simulated NDVI data from two different sites for a period of 300 months. The introduced simulated NDVI data is with and without seasonal cycles respectively, and is generated by the use of artificial events and noise.

The DBEST algorithm consists of two parts - a trend estimation method and segmentation method. The trend estimation method calculates for each data point the absolute difference with the next one and the differences with the mean values for the measures before, and after the current point. The results from those calculations are filtered by user defined thresholds and the remaining points are added to a sorted list in descending order according to the absolute value of the shift. Then, the spacing between the candidate points is compared with a user defined threshold. The points that pass the last criteria are used as braking points for a given time series. If the data is not already deseasonalized, a Seasonal-Trend Decomposition Procedure Based on Loess (STL) is performed for each part defined by the braking points.

The segmentation part of the algorithm begins with peak/valley detector function, which detects the points where the direction of the trend changes. The straight lines that are connecting the last points are then used in turning point detector function. In this function, the perpendicular distance between each point and the pick/valley line is compared with a threshold value. The algorithm continues by assuming that all detected points, including the peak/valley points and the level shift points, are turning points. All these points are used in a trend local change function, which calculates the absolute difference in each pair and creates a list of points sorted in descending order by the magnitude of change. Then, the least-squares fit and the Bayesian Information Criterion (BIC) are calculated for the time series by the use of a different number of points from the last sorted list. The calculations with the minimum BIC value define the valid turning points from the list and are assumed to be the trend breakpoints.

Then, the user can choose to use all breakpoints in order to detect the time points with abrupt or gradual changes, or a subset of points based on different criteria in order to generalise the trend. Finally, depending on the selected options, the algorithm outputs the change type, value, direction, significance, starting and ending time or the generalised trend, the number of segments, and fit quality measures.

The simulated testing over the two sites showed that DBEST detects both abrupt and non-abrupt changes, and accurately estimates the trend's time of change and its magnitude. The study showed that most of the greening (46%) and browning (41%) in Iraq is gradual. The larger changes for both, greening (5%) and browning (5%), are in northern and eastern Iraq. In the study, DBEST detects a small number (~1%) of abrupt change in the southeast part of the country.



### **4.3 BFAST**

A study by Verbesselt et al. (2010) focuses on the trend change detection and seasonal components and proposes a generic method called Breaks For Additive Seasonal and Trend (BFAST). In the study, BFAST is tested over simulated NDVI time series with varying amounts of seasonality, noise and abrupt changes, as well as over 16-day NDVI MODIS composites in South Eastern Australia. The two tests in the study show that BFAST is robust against noise and is not influenced by changes in amplitude of the seasonal component, as well as that it correctly detects the spatial and temporal changes in the real testing site.

BFAST consists of two parts - in the first the algorithm decomposes a time series into a trend, seasonal and noise component, then in the second part it detects and characterizes the trend changes. To execute those two phases the algorithm doesn't require from the user to select a reference period, set a threshold or define a change trajectory. In the first phase, BFAST uses the STL method to calculate the initial seasonal component. Then, it uses a piecewise linear approximation to calculate the trend parameters and the length of season. The last are used to adjust the seasonal braking points by assuming that the sum of the seasonal component in one cycle should be zero. BFAST applies the Ordinary Least Squares (OLS) residuals-based Moving Sum (MOSUM) test to find the breaking points before the piecewise linear regression and BIC to determine their number. The change characteristics are derived from the differences between the intercepts and the slopes of the neighbouring linear models.

### **4.4 LandTrendr**

In a study by Kennedy et al. (2010), the Landsat-based detection of Trends in Disturbance and Recovery (LandTrendr) algorithm extracts the spectral trajectories from several test sites in USA. The results from the tests prove that LandTrendr can capture a wide range of disturbances, recovery phenomena and stable longer-duration processes.

To achieve this, the LandTrendr goes through three main steps – pre-processing, trajectory extraction, trajectory segmentation. In the pre-processing phase the algorithm ensures the quality of the images from time-series by performing geometric corrections, radiometric normalizations and transformations, as well as it detects and eliminates various types of noise from clouds, snow, smoke and shadows. Then, the algorithm selects the clear (cloudless) pixel values from the images closest to the median Julian day for each year. From the last selection, the algorithm extracts the spectral trajectories by taking the mean pixel values from user defined kernel.

Later, LandTrendr selects good measures from the annual Landsat time-series and it passes them to the next essential phase – the trajectory segmentation. In this phase, the algorithm iterates through the time-series and removes all spikes where the spectral value before and after the spike is similar. Then, it uses regression and angle criteria to identify the potential breaking points. A second set of fitting algorithms determine the spectral values for each potential point that would result in the best continuous trajectory through

the time series. The segments from the last steps are analysed and simplified. At the end, the algorithm determines the best segmentation model of the trajectory and evaluates the result.

#### ***4.5 TIMESAT***

Jönsson and Eklundh (2004) present a program for analysing time-series with three different least-squares methods. In the study, the TIMESAT algorithms are tested over 10-days NDVI time-series for Africa in the period between year 1998 and 2000. The results from the tests show that the extracted seasonal data from the different TIMESAT modes are in general agreement with the climatic patterns for the region. Besides, the study also examines and compares the different methods in few selected cases.

Initially, the TIMESAT algorithm allows the users to improve the data by the use of additional cloud data, which are translated into weights and used to determine the uncertainty of the NDVI values. Then, a second step reduces the noise by a process of adaption to the upper envelope which implies that the noise is with smaller NDVI value. After the data is cleared, the algorithm calculates the number of seasons by comparing the amplitudes of the maxima from fitting procedures. In this part, the algorithms imply maximum of three annual growing seasons. So, if the amplitude of the secondary maxima exceeds the first maxima, then the annual seasons are two; if it is below the first maxima, the season is one. The algorithm continues with a method that uses local polynomial functions, which is also described as adaptive Savitzky–Golay filter. The other two optional least-squares methods that TIMESAT allows are - to a polynomial and harmonic basis, and to asymmetric Gaussian functions.

## 5. Data

### 5.1 GIMMS NDVI3g

The GIMMS (Global Inventory Modeling and Mapping Studies) NDVI data product is one of the longest and most comprehensive NDVI data series that is available for global or large-area phenology studies (USGS, 2015). It is measured with AVHRR instruments that are carried by the NOAA POES satellite series 7, 9, 11, 14, 16 – 19, and EUMETSAT MetOp-A and MetOp-B (Pinzon et al., 2005). The daily daytime AVHRR global coverage dataset with 4 km resolution is processed to bimonthly NDVI dataset in Albers Conical Equal Area projection with 8 km resolution. The dataset time-series start from July 1981 to the present and cover all continents except Antarctica (Tucker et al., 2005).

The dataset is calibrated and corrected to minimize various effects that are not related to the vegetation, among which are the NDVI variations due to the changes in the solar zenith angle that are associated with the satellite's orbital drifts (Pinzon et al., 2005), as well as, an atmospheric correction for the El Chichon and Mt Pinatubo volcanic stratospheric aerosol periods. In the final assessment of the dataset more corrections are introduced that are related to the data drop outs, composite discontinuities, navigation and other errors (Tucker et al., 2005).

Up to the date of this text, the latest version of the dataset is termed GIMMS NDVI3g or third generation GIMMS NDVI from AVHRR sensors (Pinzon et al., 2014). It extends the previous GIMMS generation to December 2011 by including the NOAA POES satellite series 18 and 19, and EUMETSAT MetOp-A and MetOp-B. Other significant changes in the third generation are the improved calibration between AVHRR/2 (July 1981 to November 2000) and AVHRR/3 (November 2000 to the present), as well as the incorporation of improved snowmelt detection algorithm with calibration for the arctic growing season (Pinzon et al., 2010).

### 5.2 GIMMS NDVI3g Data Description

Each file name from the dataset carries a unique name and information. They are constructed from the five variables, which are formatted as presented below:

*geo[year][month][period].n[sat]-VI[version]g*

The temporal resolution of the dataset is 15 days, so for each month there are two periodic files. The first is the maximum value composite from the first 15 days of the month (15a) and the second file with the values from the 16th day until the end of the month (15b).

Each file from the dataset stores 16-bit signed integer values that are saved as big endian byte order. The values are in grid with dimensions 4320 by 2160 cells, where each cell is with size 0.0833 degrees.

**Table 5.1: Geographical coordinates of the grid, where 1/24 is half of the pixel size.**

Grid corner	Latitude	Longitude
Upper left	$90.0 - 1/24$	$-180.0 + 1/24$
Lower right	$-90.0 + 1/24$	$180.0 - 1/24$

**Table 5.2: GIMMS dataset code values.**

Value	Label
0.0 - 1.0	NDVI
-0.1	Water
-0.05	No Data

All of the GIMMS NDVI values are subject to a scale factor of 10000. In addition, a flag value is added to each NDVI value from the dataset:

$$\text{ndvi3g} = \text{round}(\text{ndvi} * 10000) + \text{flagW} - 1, \quad (2)$$

and in order to retrieve the original ndvi and the flag value:

$$\text{flagW} = \text{ndvi3g} - \text{floor}(\text{ndvi3g}/10) * 10 + 1 \quad (3)$$

$$\text{ndvi} = \text{floor}(\text{ndvi3g}/10)/1000 \quad (4)$$

**Table 5.3: GIMMS NDVI3g flag values and their descriptions.**

Flag Value	Description
1	Good value
2	Good value
3	NDVI retrieved from spline interpolation
4	NDVI retrieved from spline interpolation, possibly snow
5	NDVI retrieved from average seasonal profile
6	NDVI retrieved from average seasonal profile, possibly snow
7	Missing data

### 5.3 UMD Land Cover Classification

The University of Maryland Department of Geography (UMD) created a global land cover classification collection in 1998 from AVHRR data collected from year 1981 to 1994. Their product is available for free on-line in three different resolutions: 1 degree, 8 kilometre and 1 kilometre pixel resolutions. The data can be downloaded in Geographic projection as a GeoTIFF file or in Goodes projection as a binary BSQ file. The data-set with 8 km pixel resolution was used in this work. This version has overall accuracy between 81.4% and 90.3% (De Fries et al., 1998).

**Table 5.4: Code values of the UMD land-cover data set (8 km pixel resolution) and class-specific accuracies (De Fries et al., 1998).**

Value	Label	Lower Bound	Upper Bound
0	Water	-	-
1	Evergreen Needleleaf Forest	86.4%	90.0%
2	Evergreen Broadleaf Forest	87.5%	95.2%
3	Deciduous Needleleaf Forest	64.3%	64.3%
4	Deciduous Broadleaf Forest	57.4%	82.9%
5	Mixed Forest	68.1%	81.0%
6	Woodland	74.2%	77.9%
7	Wooded Grassland	87.0%	90.0%
8	Closed Shrubland	71.9%	83.3%
9	Open Shrubland	93.2%	95.4%
10	Grassland	82.4%	90.9%
11	Cropland	77.4%	88.1%
12	Bare Ground	97.9%	99.2%
13	Urban and Built	83.3%	90.9%

The following note in relation to Table 5.4 is published at the UMD website:

“Please note the following errors while trying to match the data and the legend for the 1km and 8km products. In the 1 km AVHRR land cover classification product, Code value 14 is used to represent Urban and Built class instead of Code value 13 as in the legend above. In the 8km AVHRR land cover classification product, code 13 represents regions with "permanent snow and ice" and not "urban and built". While code 14 shouldn't exist, it appears erroneously in the dataset over areas with bare ground and water. If code 14 appears over Greenland, Antarctica or similar icy areas, please recode the data to code 12. If some code 14 values appear along the image borders in the ocean, please recode these to code 0 which is water. “

## **5.4. Global Administrative Areas**

Global Administrative Areas (GADM) is a free spatial database of the locations of the global administrative boundaries, such as countries and their lower level subdivisions. Besides the spatial data, the database contains attribute information, such as names, variants of the names and other. The current version of GADM is 2.5 and it includes 237 726 administrative areas around the world.

The data can be downloaded as shapefile, ESRI geodatabase, RData, and Google Earth kmz format. The spatial data is stored in longitude/latitude and uses WGS84 datum. In the current work, the GADM data for the Middle East will be accessed, downloaded and used through the 'sp' R package (Pebesma & Bivand, 2005).



## 6. Methods

### 6.1 Overview

Three different algorithms were developed in R Studio in compliance with this project aims. The first two performed a temporal trend analysis over the Middle East by the use of NDVI 3G dataset. Both algorithms produced spatial data and maps, which were used by the third algorithm. This third algorithm, also used the land-cover dataset developed by the UMD and produced an overlay analysis, maps and statistic data.

For clarification, the first algorithm will be named Middle East Temporal Analysis with PolyTrend (META PolyTrend). The second algorithm will be named Middle East Temporal Analysis with DBEST (META DBEST). The third algorithm, related to the Middle East Land-Cover Analysis will be named MELCA.

In general, META PolyTrend (Appendix D) and META DBEST (Appendix E) share some common steps that can be reused in each of them. These common functions are related to the data reading, the data preparation and partly to the final outputs. However, PolyTrend and DBEST have different input interface and outputs. Besides, in META PolyTrend the annual NDVI data is generalized by a specific rule (discussed further). That is why after PolyTrend and DBEST were coded in R language, a wrapper function was developed in order to integrate each of the algorithms. This function handles the processing of the raw NDVI time-series data and the execution of PolyTrend or DBEST. The wrapper function and part of the functions related to the final outputs, differs in each of the analysis over the Middle East.

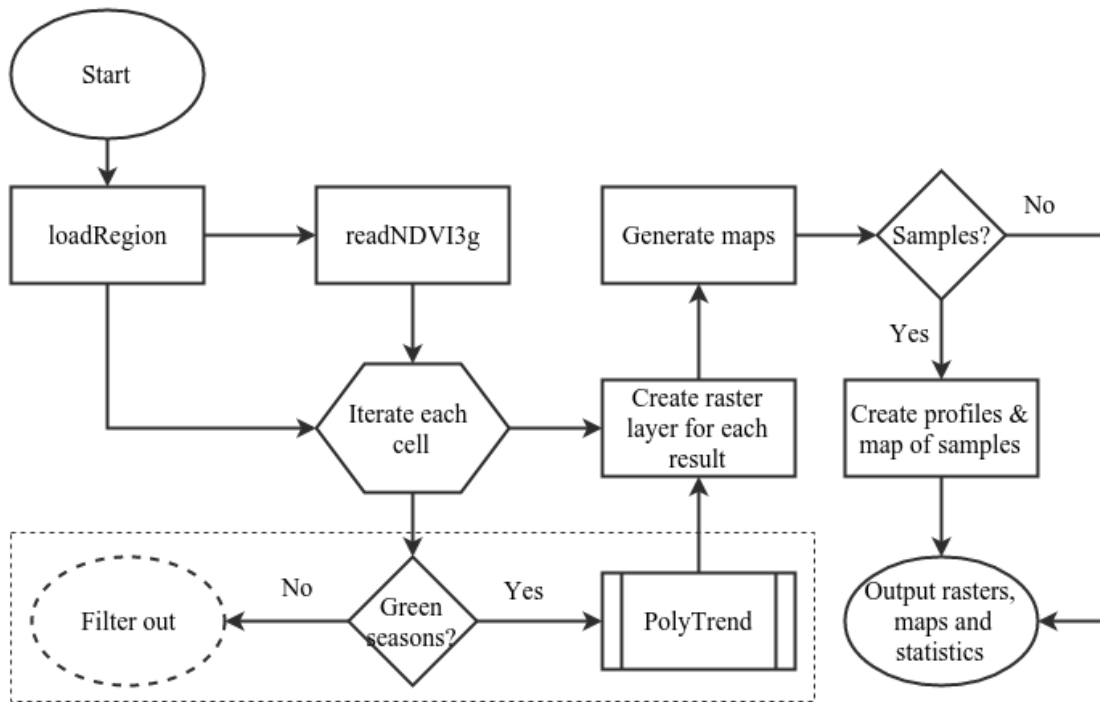


Figure 6.1: Flow chart of META PolyTrend (implemented in R). The dashed rectangle indicates the wrapper function.

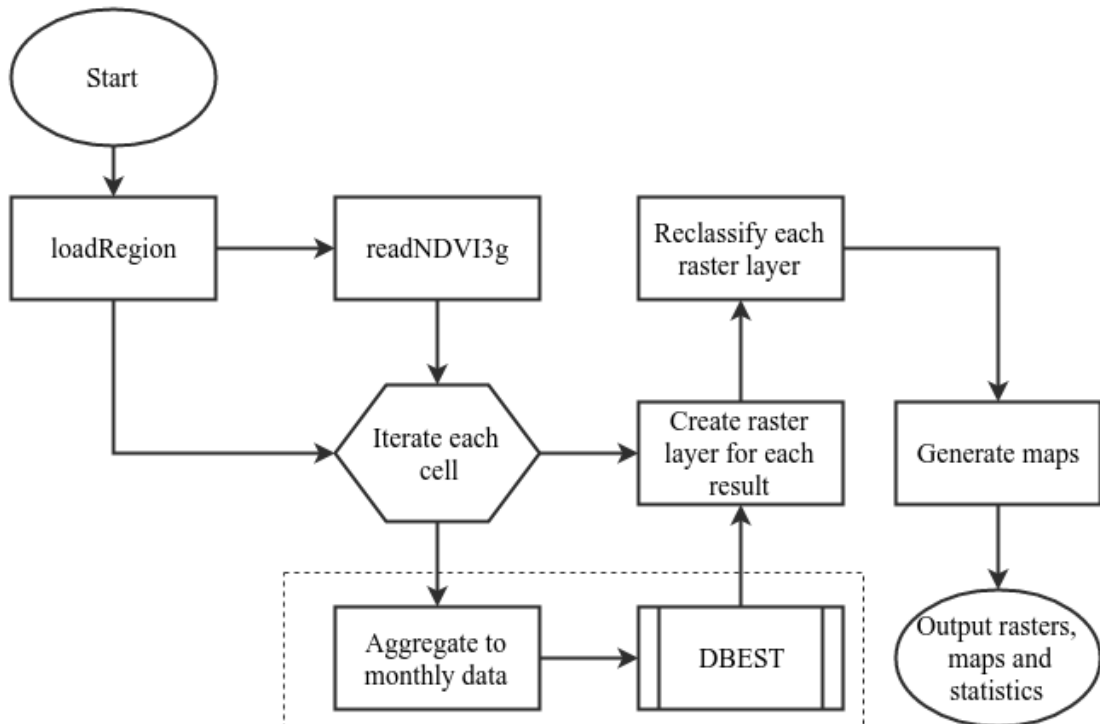


Figure 6.2: Flow chart of META DBEST (implemented in R). The dashed rectangle indicates the wrapper function.



## 6.2 Data preparation

The two functions that are going to be reused in META PolyTrend and META DBEST are related to the data preparation (Figure 6.1 and Figure 6.2). In the source code they are named *loadRegion* (Appendix B) and *readNDVI3g* (Appendix C). Briefly, the *loadRegion* function downloads and merges the borders of the countries in the defined region, executes the *readNDVI3g* function and returns a stack with layers. This stack with layers represents the NDVI 3g data cropped only for the given region.

The *loadRegion* function is constructed with several input parameters: the names of the countries in the region, the start date, the end date and the path to the NDVI 3g dataset. The first parameter should be a list with the countries' codes according to ISO 3 naming or ISO 3166-1 alpha-3 published by the United Nations (UN). The second and the third variables define the period (e.g. '1982-01', '2010-12') that is going to be analysed. The last parameter defines the path to the NDVI 3g dataset.

When executed, the *loadRegion* function begins downloading from the Internet the GADM datasets for the defined countries. The downloaded data is stored in the R Studio working directory. On any following run of the function with the same countries (from any of the algorithms), it will first check if the countries' data are already downloaded, in order to directly use them. The function continues by binding the countries boundaries into a single Spatial Polygon Data Frame (SPDF). Finally, it rasterizes the region SPDF by writing it to a raster template.

After this process, the function defines the parameters that are related to the NDVI 3g dataset. The ones that are related to the actual reading of the data are: the number of columns, the number of rows, the variable type and the byte order. Before stepping into the reading process, the *loadRegion* function creates an ordered list with the names of the files that are going to be used in several steps.

First, the function generates a list with dates for the defined period, where for each month included in the period there are two measurements on every 1<sup>st</sup> and 16<sup>th</sup> day of the month. Then, from the previous list it creates another list that contains the merged years and the names of the months for the period as strings. The last list is modified in a way so that the string '15a' is assigned to each odd element, and to each even is assigned the string '15b' respectively. This corresponds to the naming of GIMMS NDVI3g dataset specification. Each element from the last list is used as a pattern in a function, which finds and returns the full name of the corresponding file for a particular date by locating it in the dataset path.

As it was mentioned, the *loadRegion* function executes the *readNDVI3g* function. This happens by applying of the last function over the list with the NDVI 3g filenames. The input parameters of *readNDVI3g* are: the full name of a current NDVI 3g file, the data size, the data type, the byte's order, the number of columns, the number of lines, the region raster and the region as SPDF. The result from this procedure is a stack with layers, each of which contains the NDVI 3g values for a given date in the defined region.

The *readNDVI3g* function takes a given file and reads it as a binary file by the use of the already mentioned parameters. The returned matrix is written into a raster template, where the water (-10000) and the no-data (-5000) values are set to 'NA' (not applicable). The raster values are then rescaled by 10000. The last raster is masked with the region raster. The result from this operation is cropped to the extent of the region. The last step is the result that the *readNDVI3g* function returns to the *loadRegion* function.

### **6.3 Data processing**

The raster calculator is a function from the R raster package. It creates a new raster object by applying a function or a formula over another raster object. In META PolyTrend and META DBEST, the raster calculator applies a wrapper function over the raster stack that was created by the data preparation process. In both algorithms, the result of the raster calculator is another stack with layers. The layers in this stack represent the values returned by either PolyTrend or DBEST respectfully.

The wrapper function in META PolyTrend begins by checking if the temporal NDVI 3g data contain any NA (no-data or water) values. If they do, the wrapper function will directly finish and return a vector with four NA values. If no NA values are found in the temporal data, the algorithm will check for the existence of a long period without vegetation or low NDVI values. This is validated by calculation of the mean NDVI value for each year from the defined period compared to a threshold. The variable for this threshold can be set in the META PolyTrend configuration section. In this work the value for the threshold is set to 0.1. If any period matches the condition, the wrapper function exits by returning a vector with four NA values.

If the temporal NDVI 3g data passes the two conditions described above, the algorithm aggregates the bi-monthly data to a monthly data. This is done in order to reduce the noise effect, as well as for easier interpretation of the results. The aggregated monthly data is then divided into vectors for each of the years in the period. From each of the years are calculated the mean NDVI values of the green period. The length of the green period can be set in the META PolyTrend configuration section. In this work it is set to 4 months.

Most of the plant growth occurs during the rainy periods. In large part of the Middle East the climate is arid and almost all of the annual precipitation falls during three months. However, most of the vegetation is located in the northern part of the region, in mediterranean and humid continental climate zones. The precipitation for that part of the Middle East is more equally distributed through the year and that is why, the 4-month length of the growing season should be more suitable for the analyse.

The result of these steps is a vector with values that are the mean values of the green period for each year. The last vector is used as an input value for the PolyTrend algorithm. The wrapper function exits by returning the trend type, slope, direction and significance.

The wrapper function in META DBEST exits directly too, if the temporal NDVI 3g data contain any NA values. In contrast to META PolyTrend, if this condition is not passed, the function returns seven NA values. If it is passed, however, the wrapper function in META DBEST directly aggregates the bi-monthly NDVI measurements to monthly data, and uses the last ones in DBEST algorithm as input data. On a successful finish, the function returns a vector with seven values: the number of trend segments, the start year of the change, the duration, the end year of the change, the change value, the change type and significance.

## ***6.4 PolyTrend in R language***

The PolyTrend algorithm (Figure 6.3 and Appendix G) was coded in R language with the same logic described in the article “Automated mapping of vegetation trends with polynomials using NDVI imagery over the Sahel” by Jamali et al. (2014). The R version of the algorithm uses the *stats* package in order to fit the linear models, calculate the p-values and slopes. The PolyTrend algorithm in R language also extends the R plot and print functions. This allows graphics and outputs to be displayed when it is used in R environment.

As an input the PolyTrend algorithm takes a vector data and the significance level (alpha). In this work, the significance level value was set to 0.05. The return from the algorithm is a class that contains the input data plus the trend type, slope, significance, direction and polynomial degree. The trend types can be: a concealed trend, no trend, linear trend, quadratic trend or cubic trend. However, the returned trend type variable is a number, so this corresponds to: -1,0,1,2 and 3 in the same order as the one enumerated above. Likewise, the significance variable is a numeric value too. It can take values of -1 or 1, which correspond to statistically insignificant or statistically significant respectively.

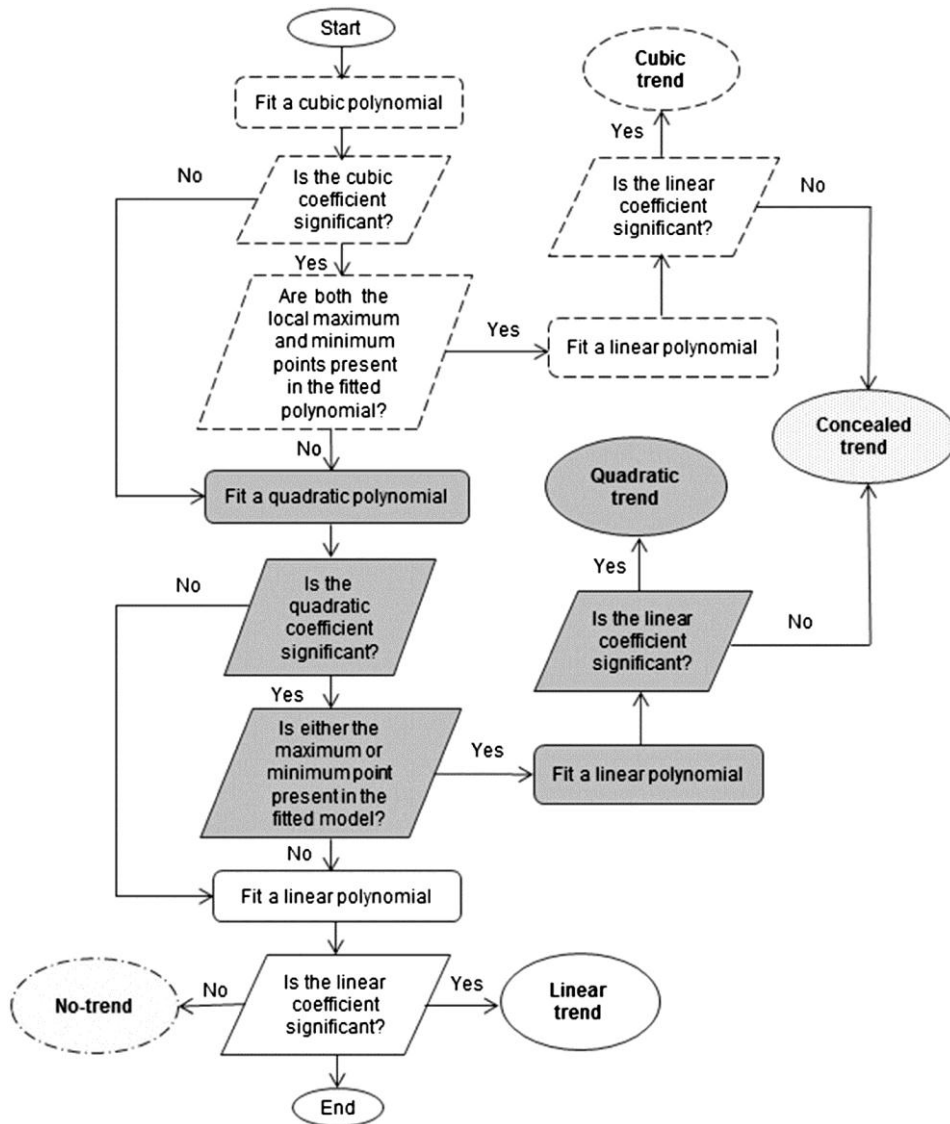
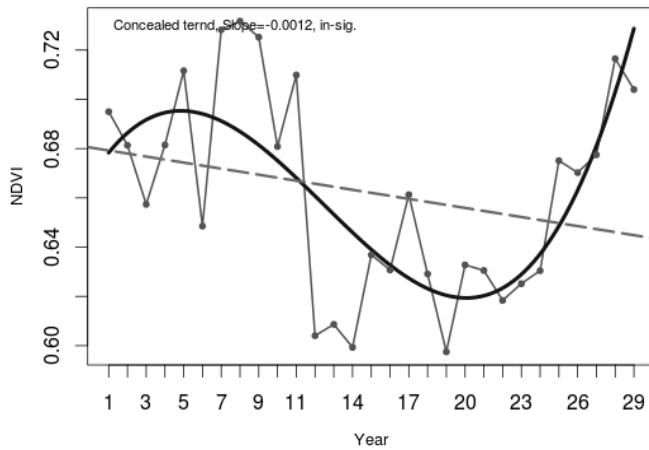


Figure 6.3: Flow chart of PolyTrend. Source: Jamali et al., 2014.

The polynomial degree variable is another numerical variable that is returned by the algorithm. It can be either 0, 1, 2 or 3, which corresponds to no trend, linear, quadratic or cubic trend. However, when a user wants to examine the result from PolyTrend, besides the numerical values, the extended print method outputs their textual interpretations too.



**Figure 6.4:** Example of a graphical output by PolyTrend algorithm (insignificant cubic trend).

### ***6.5 DBEST in R language***

The development of the DBEST algorithm (Appendix H) in R language is based on the article “Detecting changes in vegetation trends using time series segmentation” by Jamali et al. (2015). It is a complex algorithm that can be used either for change detection or trend generalization. In this work, DBEST was used for change detection. Moreover, the algorithm can be configured with various options that are explained further in the text.

The first input to DBEST is the data or in this case, the temporal monthly NDVI data calculated at the beginning of the wrapper function. Other variables related to the input data are the seasonality and the data type periodicity. The latter one can be setup either as cyclical or non-cyclical data. The NDVI data in this research will be analysed as cyclical with one year seasonality (12 months).

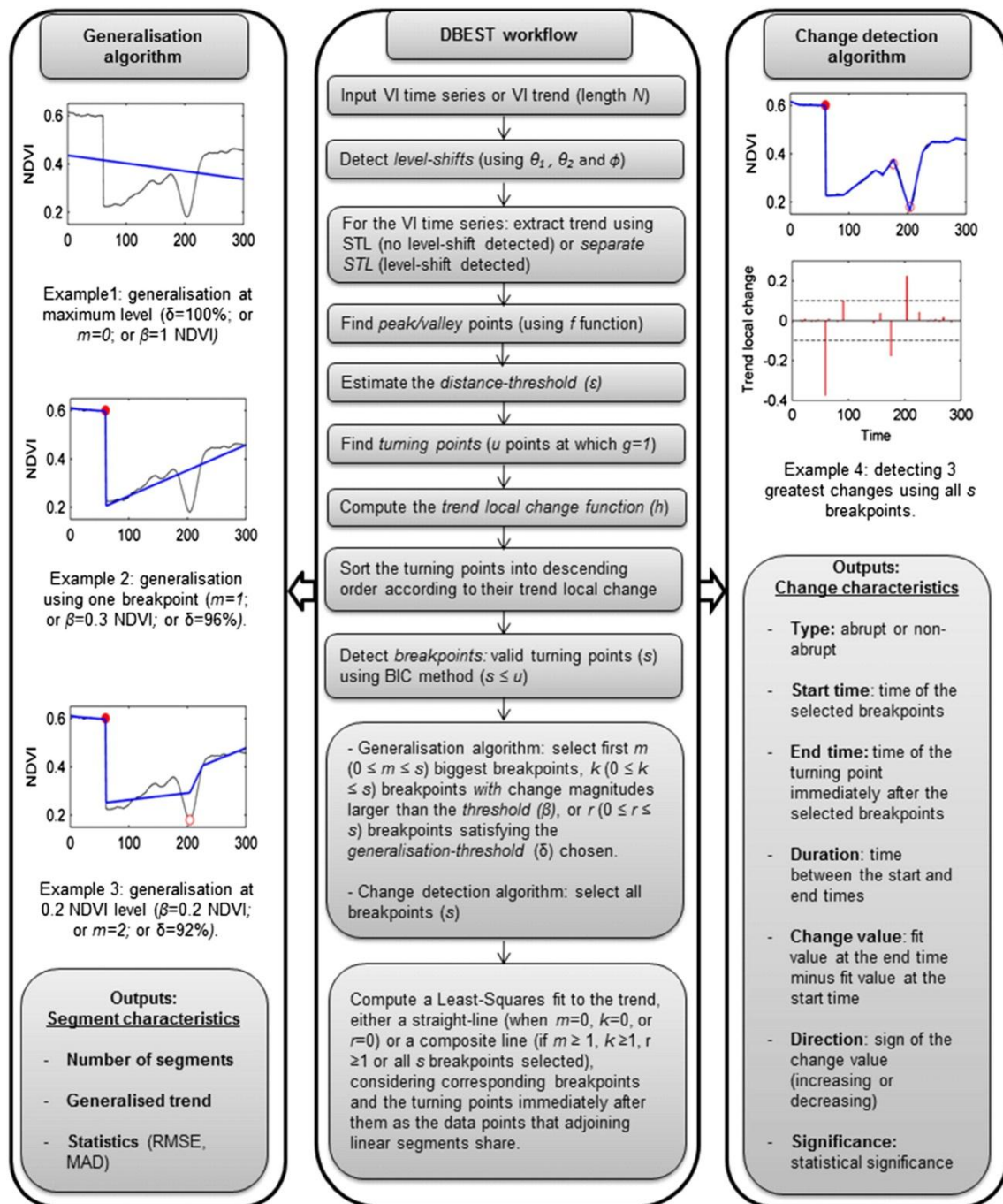


Figure 6.5: Flow chart of DBEST. Source: Jamali et al., 2015.

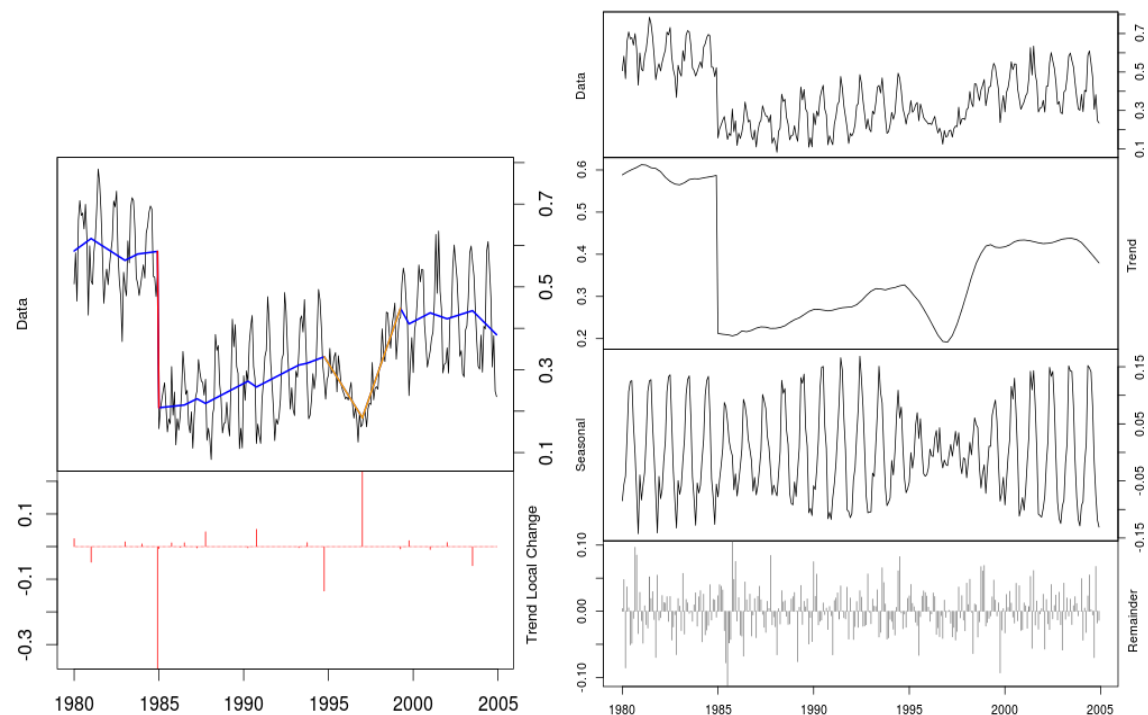
Similar to PolyTrend, in the R version of DBEST, the print and plot methods were extended to match the characteristics of the R environment. DBEST in R language returns two different classes, one type of class when used for generalisation and another when used for change detection. The first class contains: the number of segments, the Root Mean Squared Error (RMSE), the Mean Absolute Deviation (MAD) and the fit values. The change detection class contains: the number of break points, the number of segments,

the start of the change, the duration, the end of the change, the change value, the change type, the change significance and the values of the fit.

Likewise, DBEST can be configured to analyse a specific number of breakpoints, magnitude of change or percentage of generalization respectively. Since in this work the aim is the analysis of the greatest change in the Middle East, the number of breakpoints are set to 1.

The first and the second level-shift thresholds are another two variables that DBEST uses in the computations. The first allows the user to set the lowest absolute difference in the time-series between the level-shift point and next data point. While the second one is the value of the lowest absolute difference in the means of the data calculated over the period before and after the level-shift point. In this research, the first level shift was set to 0.1, the second level shift to 0.2 and the duration to two years or the value 24.

DBEST can calculate the lowest perpendicular distance from farthest data point to the straight line passing through every pair of successive peak and valley points. However, this distance threshold can also be setup by the user. In this work, the distance is left as default to be calculated automatically by DBEST. Like in PolyTrend, the statistical significance level value was set to 0.05. Finally, the optional plotting variable (on/off) allows the user to plot graphics of the output. In the case of META DBEST this option was omitted.



**Figure 6.6: An example of two graphical outputs by DBEST algorithm in a change detection mode. On the left side, the red line represents an abrupt change, while the yellow ones show non-abrupt changes. On the right side, the cyclical NDVI data is decomposed to trend, seasonal and remainder components.**

## **6.6 Outputs**

The raster calculator function in META PolyTrend and META DBEST applies the wrapper function over the processed NDVI 3g data stack. The return of this operation is another stack with layers. These layers are exported to the output path (specified in the configuration section) as files in GeoTIFF format and are named according to the name of the value they represent. Next, the algorithm generates cell statistic in text files which are related to each of the layers. Both algorithms continue with the final process that is different in each one of them. During this process, each of the algorithms will generate a number of maps and write them in files in a PDF format.

The maps in META PolyTrend are produced by a new defined function, which is applied over each of the layers from the last stack. This function sets the map title, colours, legend, type of legend, labels and etc. Moreover, this function can probe a number of random points with the PolyTrend algorithm, draw the location of the samples on an additional map and export the generated from PolyTrend graphic to a PDF file. During the process of random sampling, the algorithm looks for a location for each of the polynomial degree found in the data.

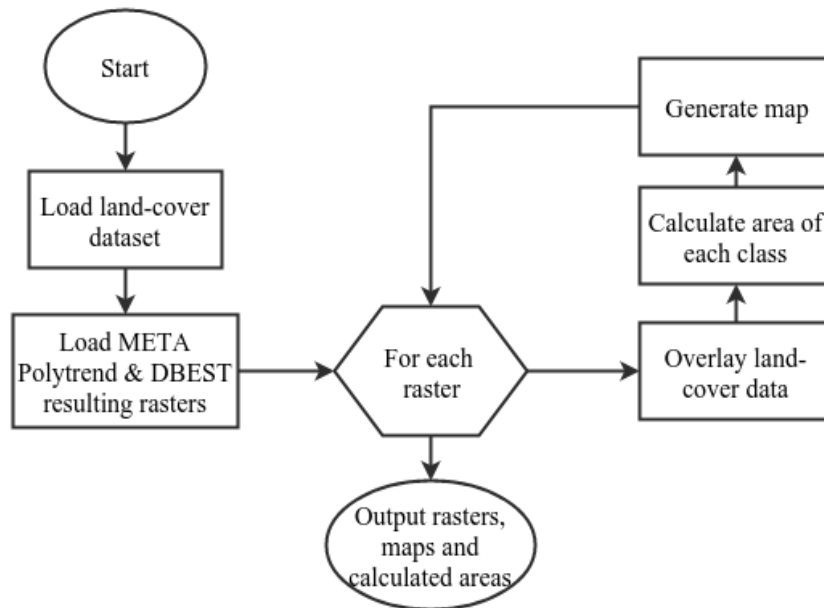
The function that generates the PDF maps in META DBEST takes as input parameters the map title, the map colours, the legend type and labels. Although the last look similar to the function used in META PolyTrend, the function here has some major differences. For example, the function in META DBEST does not sample the data, the legend can display only classes, the displayed values must have ascending order and etc. Moreover, the process of applying the function over the resulting stack of layers is different. In META DBEST the raster layers are reclassified and then used in the function described above.

## **6.7 Land-cover/use overlay analysis**

The MELCA algorithm (Appendix F) starts by defining the META PolyTrend and META DBEST working directories, an output directory and the countries of interest. Then, it loads into a raster object the AVHRR land-cover dataset with 8km resolution (developed by UMD), as well as the raster files produced by META PolyTrend and META DBEST.

Like in the other two algorithms, MELCA downloads the region data from GADM and merges the countries boundaries into a SPDF. The last object is used to crop the land-cover raster to the extent of the defined region. The result of this step is a raster that contains the land-cover of the Middle East. Then, it is reclassified and used as an input parameter to the function that produces a PDF map. The other input parameters of this function are the map title, the colours and the labels according to UMD product specification.





**Figure 6.7: Flow chart of MELCA algorithm (implemented in R).**

The reclassified land-cover raster and land-cover labels are passed as input parameters to a new function. This function calculates the approximate surface area of each raster classes and saves the calculation in a text file. This generated text file contains a table with the names of the raster classes, their integer values and their area in square kilometres. In addition to the above input parameters, this function also needs an output directory and an output file name.

This area calculating function begins by calculating the area of each cell from the input raster and storing the area values to a new raster object. Then, it computes a zonal statistic with the last raster and the input raster which represents the zones. The cell values are summed by zones and stored into a table. Next, the table rows are named with the input label. The function ends by saving the zonal statistic table in a text file to the output directory.

After the area of each land-cover class is computed, the MELCA algorithm continues with a process during which it overlays the land-cover with each class from the rasters produced by META PolyTrend and META DBEST. To do that, the algorithm defines one more function which takes as input parameters a raster result, the last land-cover raster, the land-cover labels, the land-cover colour, the region SPDF, an output file name, the name of the class, the class value and the output directory. As a result this function produces a map in a PDF file format and a text file with calculated areas.

The function begins with re-sampling the land-cover raster to the cell size of the input raster by using the nearest neighbour method. This method works by calculating the distance between both raster cells, and then to each cell of the second raster is assigned the value of the nearest cell from the first raster. The algorithm continues by reclassifying the input raster by setting all raster cell values to 'NA' with exception of the input class value, which is set to 1. Next, the cell values from the re-sampled land-cover raster are multiplied by the cell values from the reclassified input raster. After that the result is

stored in a new raster object, whose unique cell values are used as indexes for the input land-cover labels and land-cover colours. As a result only the current existent land-cover classes will be labelled and coloured.

The function continues by calculating the area of each cell of the last raster object and stores the area values to a new raster object. Next, it computes a zonal statistic from the last two rasters by summing the surface of the cells. The rows of the zonal statistic table are named according to the re-indexed labels and the table is saved in a text file. The function ends by passing the result from the multiplication operation, the re-indexed colours and re-indexed labels to a function that produces a PDF map.

The last described function is iterated for each raster class, for each raster produced by both algorithms. Prior to these loops defined in lists are the class values, the class names and the output filenames. Likewise, each raster produced by both algorithms is passed to the area calculating function. In addition, the produced by META DBEST raster files are reclassified before being processed.

As a result, the MELCA algorithm produced 34 maps in PDF file format and 42 text files with calculated areas. They include a map of the land-cover in the Middle East and the surface area of each land-cover class, the area of each class computed by META PolyTrend and META DBEST, maps and areas of the affected land-cover classes from each class computed by META PolyTrend and META DBEST.

## 7. Results

### 7.1 Results computed by PolyTrend

#### 7.1.1 Trend types of the green season in the Middle East

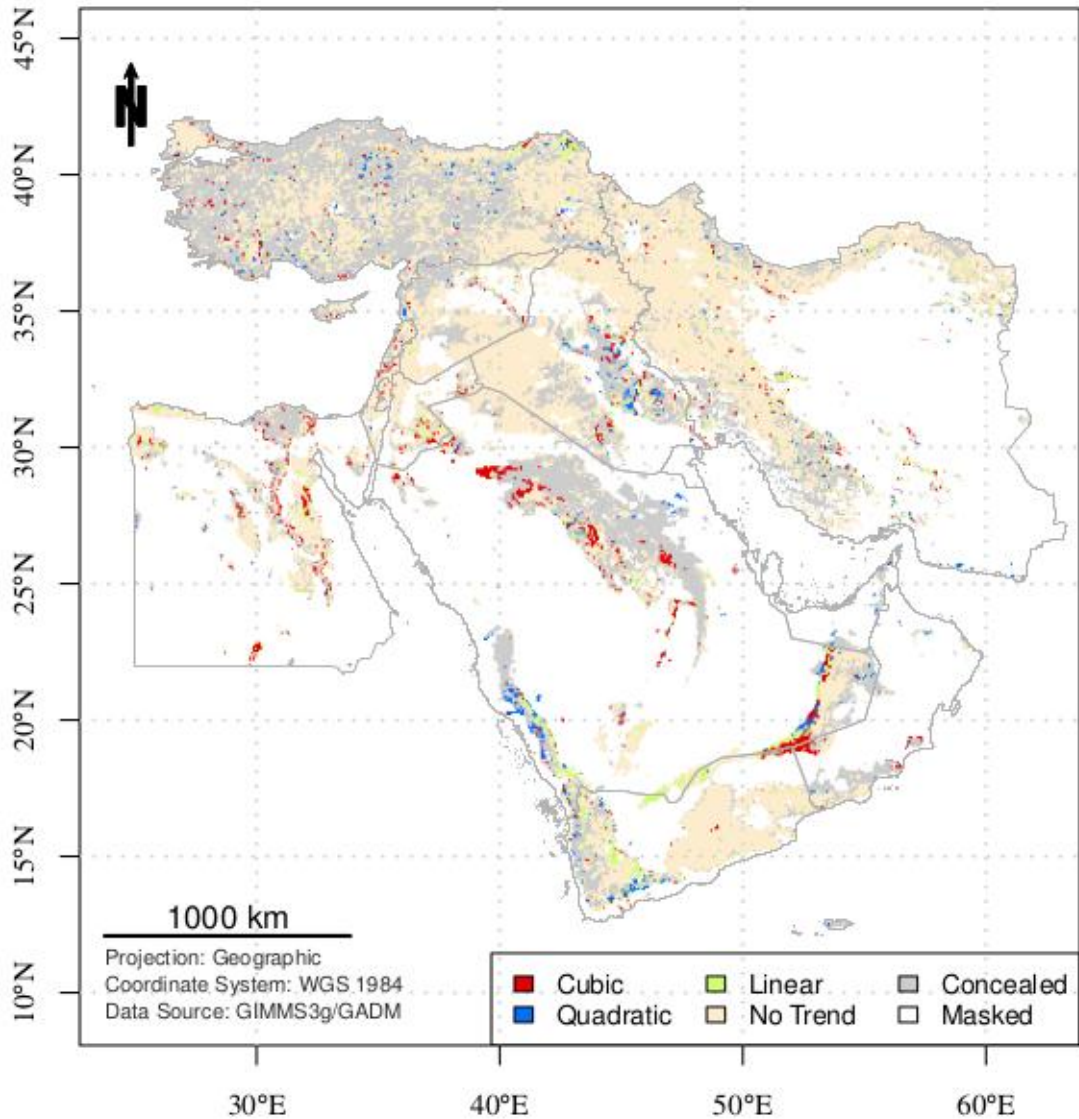


Figure 7.1: NDVI trend types in the Middle East (1982 – 2010).

Regarding spatial coverage of the obtained trends, 3.8% consists of linear trends, 1.8% of quadratic trends and 3.2% of cubic trends (Figure 7.1). The concealed trends and the NDVI data without trends involve 38.3% and 52.8 % of the entire study area, respectively.

### 7.1.1.1 Linear trends

Clusters of linear trends can be seen along the northern border of Yemen (Figure 7.1). If they are connected with a curved line, this line will cover almost half of Yemen's northern border and will continue in the north of Oman into Saudi Arabia. Similar groups of linear trends begin in the south of the mountain Jabal an Nabi Shu'ayb. They continue by surrounding the mountain from east and are stretching alongside the Tihamah coastal plain in Saudi Arabia.

Another, but not so clustered groups of linear trends can be seen in central Iran, near Zagros Mountains, in the northern region of Turkey and in Turkey in general, where they are almost equally distributed. Likewise, linear trends occur in the northern and in the central regions of Egypt, as well as in the southern regions of Jordan, near Damascus in Syria and in the central part of Saudi Arabia, near Riyadh.

The MELCA algorithm calculated that 41.7% of the linear trends are open shrubland, 11.4% grass land and 8.1% cropland (Table A.1). Most of the open shrubland is located near Jabal an Nabi Shu'ayb mountain and alongside the Tihamah coastal plain (Figure A.1).

Almost all of the clusters along the northern border of Yemen are classified as a bare ground. Most of the linear trends that are classified as grassland can be seen equally spread in Turkey, and in the northern and western regions of Iran. Moreover, two of the largest groups of cropland are visible in Turkey and Iran as well. One of the groups is located in the north-eastern corner of Turkey, near the border with Georgia. The other significant group is in central Iran, somewhere near Isfahan.

**Table 7.1: Affected land-cover by linear NDVI trends in the Middle East (1982 – 2010).**

<b>Land Cover</b>	<b>%</b>	<b>Land Cover</b>	<b>%</b>
Evergreen Needleleaf Forest	1.4	Closed Shrubland	3.5
Deciduous Broadleaf Forest	2.4	Open Shrubland	2.3
Mixed Forest	4.1	Grassland	2.1
Woodland	1.3	Cropland	2.9
Wooded Grassland	2.1	Bare Ground	1.1

Most of the mixed forest class estimated in Table 7.1 is located in the northern region of Turkey, close to the Black Sea coastline. The closed shrublands land-cover class from the above table is not grouped and is scattered in the southern parts of Turkey, the western and northern regions of Iran and near the southern corner of the Arabian Peninsula.

### 7.1.1.2 Quadratic trends

The quadratic trends surround the mountain from south. Likewise, compared to the clusters with linear trends, the quadratic trends alongside the Tihamah are found

closer to the Red Sea. Most of these groups are open shrubland (Figure A.2). As with the linear trends, the open shrubland is the biggest land cover class and it represents 44.8% of all quadratic trends (Table A.2).

The quadratic trends are like natural extension of the linear trends along the northern border of Yemen. This extension continues inside Saudi Arabia and it is the other most visible group of quadratic trends. Similarly to the linear trends, most of these groups are bare ground.

The other well visible cluster with groups of quadratic trends is located in Iraq in the area that is north-east of Baghdad. Three land-cover classes can be distinguished in that area: bare ground, open shrubland and grass land. The grass land is the third major land-cover class and it is 10.3% of all quadratic trends.

As with the linear trends, the quadratic trends are equally spread all over Turkey. They are located over various land-cover classes, including forests in the western part of the country. Likewise, croplands are visible in the north-eastern corner of Turkey. Moreover, 7.1% of all quadratic trends are cropland in the Middle East.

**Table 7.2: Affected land-cover by quadratic NDVI trends in the Middle East (1982 – 2010).**

<b>Land Cover</b>	<b>%</b>	<b>Land Cover</b>	<b>%</b>
Evergreen Needleleaf Forest	1.0	Closed Shrubland	1.7
Deciduous Broadleaf Forest	0.2	Open Shrubland	1.2
Mixed Forest	1.0	Grassland	0.9
Woodland	2.7	Cropland	1.2
Wooded Grassland	1.1	Bare Ground	0.5

As seen in Table 7.2, 2.7% of all woodlands have quadratic trends. Small patches from this land-cover class can be seen in the northern and western regions of Turkey, as well as on a few places in northern Iran, close to the Caspian Sea coastline.

### 7.1.1.3 Cubic trends

Significant area from the cubic trends consists of a bare ground (Table A.3). The largest group of them is located over the area where Yemen, Oman and Saudi Arabia join borders. A significant cluster of cubic trends and bare ground is located in the northern and central regions of Saudi Arabia (Figure A.3).

The cubic trends that belong to the open-shrubland land-cover class (20.5%) are almost equally distributed over the Middle East. The third major land-cover group in the cubic trends is the grassland - 11.5%. The most visible group of this class is located in the central region of Saudi Arabia. Significant groups of croplands are hard to be located. This type of class represents 8.9% of the cubic trends in the Middle East.

**Table 7.3: Affected land-cover by cubic NDVI trends in the Middle East (1982 – 2010).**

<b>Land Cover</b>	<b>%</b>	<b>Land Cover</b>	<b>%</b>
Evergreen Needleleaf Forest	1.5	Closed Shrubland	3.1
Deciduous Broadleaf Forest	1.6	Open Shrubland	0.9
Mixed Forest	5.2	Grassland	1.6
Woodland	1.7	Cropland	2.5
Wooded Grassland	4.3	Bare Ground	1.5

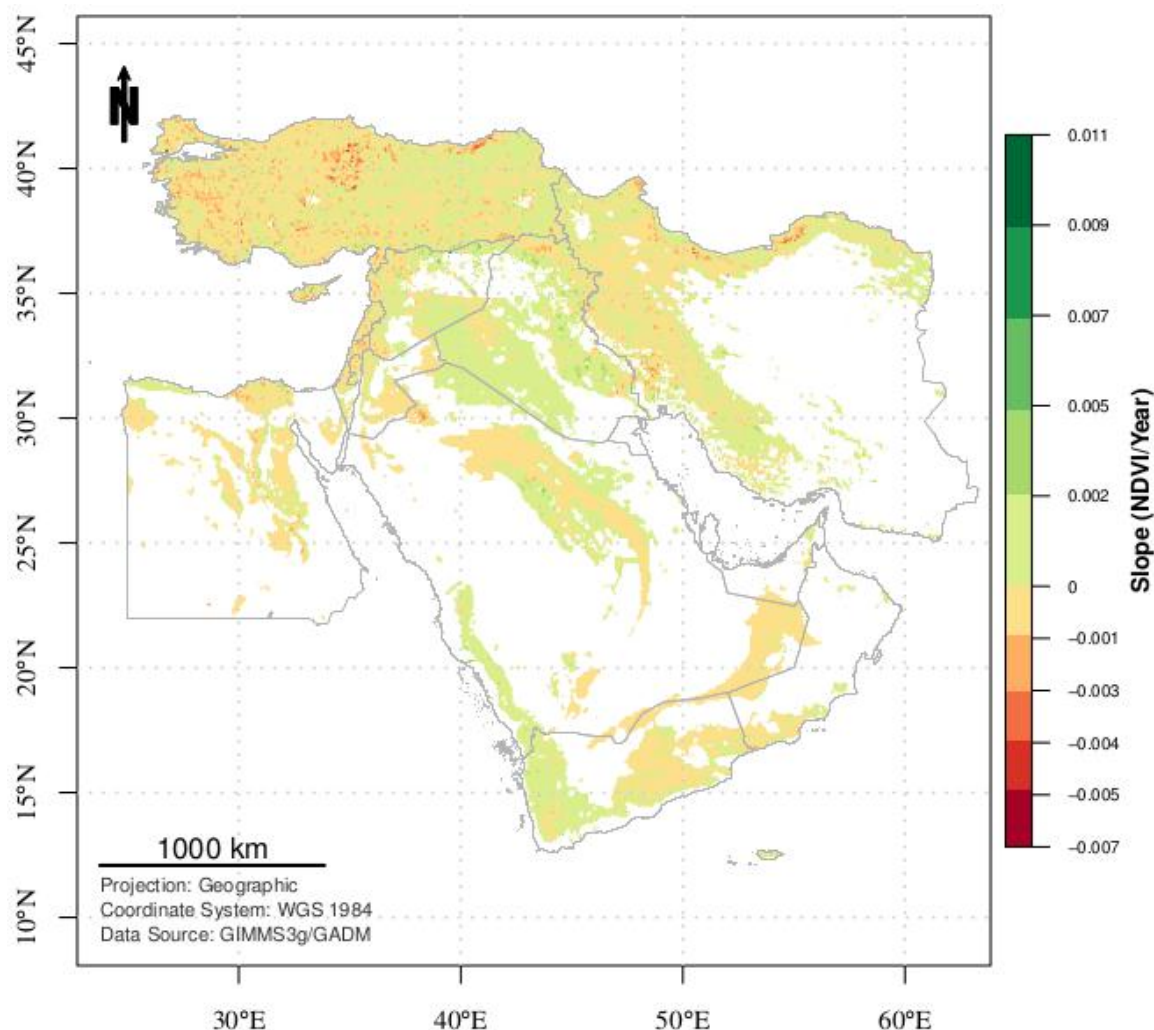
Relatively significant percentages from the total area of mixed forest and wooded grassland have cubic trends (Table 7.3). Most of this mixed forest is located on the European part of Turkey, as well as its northern regions. Likewise, mixed forest can be seen near the western corner of Iran. Patches of wooded grassland are visible in south-western Turkey and near the Mediterranean Sea.

### **7.1.2 Direction of the growing season in the Middle East**

The META PolyTrend algorithm calculated that the positive and negative trends are almost equally distributed - 51% negative trends versus 49% positive trends. Moreover, big parts of both types are well grouped with exception to the trends located in several places in Turkey and the western region of Iran – mainly near Zagros Mountains (Figure 7.2).

Well visible clusters of positive trends are located near the Jabal an Nabi Shu'ayb mountain and alongside the Tihamah coastal plain. Besides, positive trends can be found in the central part of Saudi Arabia, where they surround a large chunk of negative trends. One of the largest clusters of positive trends is located in the southern and south-western region of Iraq. From there, this cluster spreads over the territory of Syria and Saudi Arabia. Likewise, another well outlined group of positive trends is located over the coastline of Egypt.

The most well visible groups of negative trends are located along the northern border of Yemen. These groups spread in the north of Oman into Saudi Arabia. The other significant cluster that spreads like a curved line is located in the central part of Saudi Arabia. Significant groups of negative trends can be seen in Egypt, Jordan, Lebanon and Syria. Moreover, the negative trends significantly dominate the western and the central regions of Turkey, as well as most of the western region of Iran.



**Figure 7.2: NDVI trend slopes in Middle East (1982 – 2010).**

The open shrublands cover 34.3% of the positive trends (Table A.4). Most of these trends are located along the Arabian Peninsula, the eastern border of Iraq, the northern and western borders of Syria and the south-western region of Iran (Figure A.4). The third large land-cover class is the grassland. Almost the entire class is located in the western region of Turkey, near the border with Iran and in the central part of Saudi Arabia.

Out of all positive temporal NDVI trends in the Middle East 2.1% is evergreen needleleaf forest. Two clusters with this class are visible near the Caspian Sea coastline in Iran and close to the border with Turkmenistan. Groups from this class can be seen in some places in the northern region of Turkey, where it is sometimes together with or near to deciduous broadleaf forest (0.6%).

The other visually well distributed land-cover class is the cropland. Most of the cropland with positive trends is located in Turkey, where it is almost equally distributed. However, the most visible groups are located in the western part of the country. Moreover, significant groups of cropland are visible near the Caspian Sea in Iran, where the biggest one is located near the cities Rasht and Lahijan. In Syria another well visible

cluster of cropland can be spotted and its location is near to, and in the North, of Homs. Furthermore, fragmented groups of croplands can be seen along the river Nile.

**Table 7.4: Affected land-cover by positive NDVI trends in the Middle East (1982 – 2010).**

Land Cover	%	Land Cover	%
Evergreen Needleleaf Forest	38.3	Closed Shrubland	47.2
Deciduous Broadleaf Forest	27.3	Open Shrubland	24.0
Mixed Forest	25.0	Grassland	40.3
Woodland	37.3	Cropland	39.5
Wooded Grassland	29.4	Bare Ground	13.9

As visible in Table 7.4, the greening was mostly positive for the closed shrubland, the grassland and the cropland.

Despite the presence of positive trends near the river Nile, most of the cropland located in the region and in the delta of Nile is with negative trends (Figure A.5). This land-cover class represents 12.1% of all negative trends in the Middle East (Table A.5). Patches of cropland are visible along the Israel's coastline, the southern region of Lebanon, west of Homs in Syria, in many places around Turkey and the European part of the country.

Groups with negative trends of evergreen needleleaf forest (3.1%), deciduous broadleaf forest (1.5%) and mixed forest (0.5%) outline the Black Sea's coastline in Turkey and are well presented in the northern parts of this country. Such line from the same land-cover classes is visible close to the Caspian Sea in Iran as well. The wooded grassland (4%) and the woodlands (0.5%) are another well visible class in Turkey. Most of them are located in the southern region of the country and near the Mediterranean Sea coastline.

19.9% of the negative trends in the Middle East are grassland. Almost all of the grassland is located in Turkey and the western region of Iran. The other major negative trends group is the open shrubland (24.9%). Most of this group is located in the western region of Iran, in the south-eastern region of Jordan and in Yemen.

**Table 7.5: Affected land-cover by negative NDVI trends in the Middle East (1982 – 2010).**

Land Cover	%	Land Cover	%
Evergreen Needleleaf Forest	59.2	Closed Shrubland	45.1
Deciduous Broadleaf Forest	69.8	Open Shrubland	18.1
Mixed Forest	62.3	Grassland	48.3
Woodland	51.4	Cropland	58.2
Wooded Grassland	66.3	Bare Ground	14.0

Moreover, in the Middle East the green period was in negative direction for most of the deciduous broadleaf forest, the wooded grassland, the mixed forest and the evergreen needleleaf forest (Table 7.5).



Finally, large part of the NDVI trends with steep positive slope is located at several places in northern and eastern Iraq, and in Syria close to the border with Turkey (Figure 7.2). Meanwhile, the most visible groups with steep negative slope are located in Turkey, in Çorum and Rize provinces, and in Iran, in Golestan Province.

## 7.2 Results computed by DBEST

### 7.2.1 Magnitudes and types of the greatest temporal NDVI changes

7.6% of the NDVI changes in the Middle East, between years 1982 and 2010, were found to be insignificant. Big groups of the land where these trends are found are located in Dasht-e Lut (desert in Iran), in Egypt, the Western Desert and the south corner of the Arab Peninsula, east of Jabal an Nabi Shu'ayb and the Asir mountains (Figure 7.3).

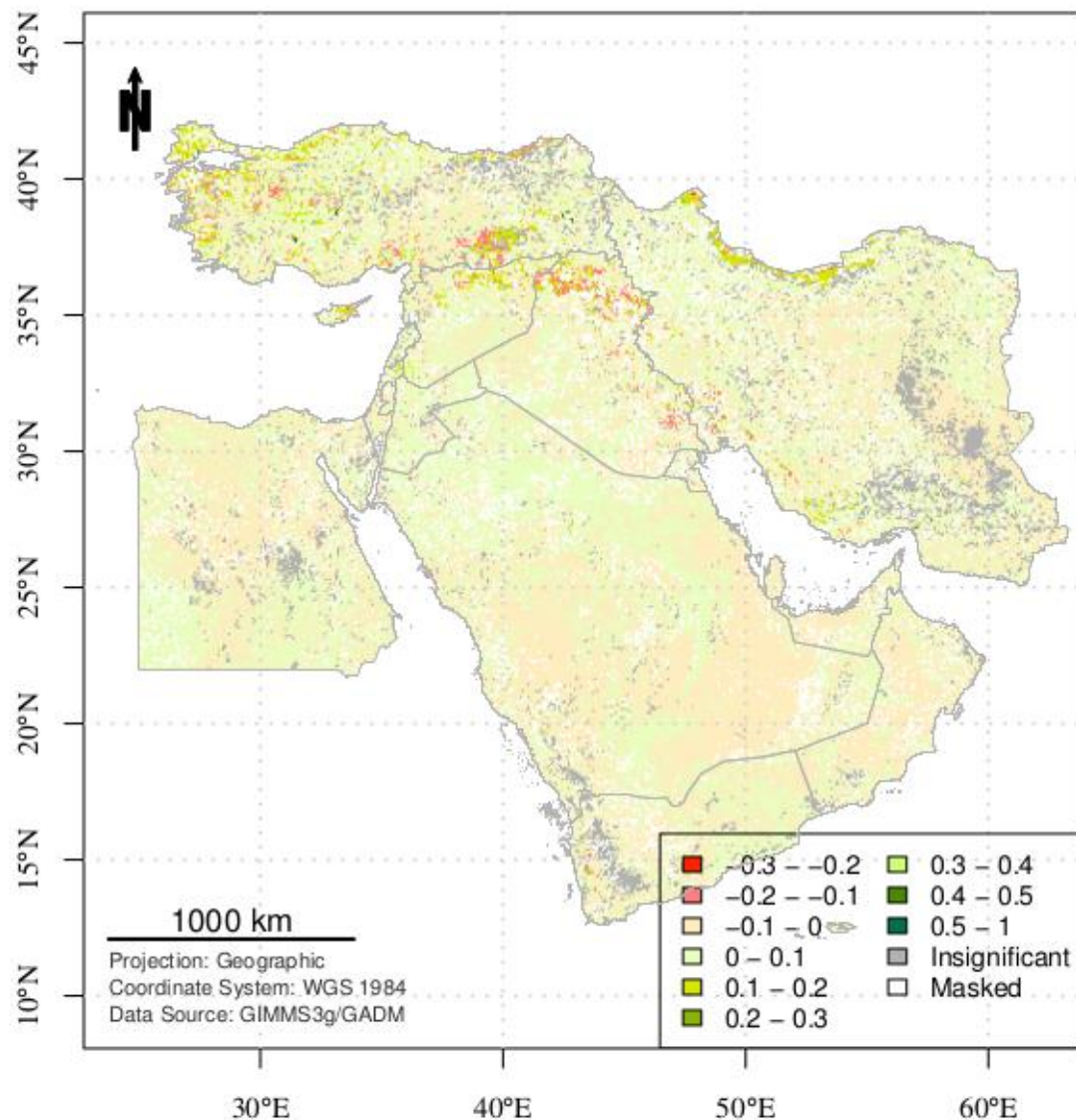


Figure 7.3: Magnitude of the greatest NDVI change in the Middle East (1982 – 2010).

1% of the trends experienced just a slight change between -0.2 and -0.1 NDVI units. As seen in Table A.6, most of these negative changes consist of cropland (24.5%), open shrubland (21.8%) and grassland (20.7%). Likewise, Figure A.6 shows that mainly they are distributed in Turkey, in North and East Iraq. However, compared to the total area of each land cover type, the greatest change with this magnitude affected mostly the forests and the woods (Table 7.6).

**Table 7.6: Affected land-cover by NDVI change between -0.2 and -0.1 units in the Middle East (1982 – 2010).**

<b>Land Cover</b>	<b>%</b>	<b>Land Cover</b>	<b>%</b>
Evergreen Needleleaf Forest	4.0	Closed Shrubland	2.9
Deciduous Broadleaf Forest	5.1	Open Shrubland	0.7
Mixed Forest	7.2	Grassland	2.1
Woodland	6.2	Cropland	4.9
Wooded Grassland	4.1	Bare Ground	0.2

The NDVI trends with greatest changes, between 0.1 – 0.2 units, can be seen in many places in Turkey, in North Syria, in North Iraq, along the Caspian Sea coastline in Iran, in the centre of Cyprus (Figure A.7). The trends affected by these values of change represent 1.9% of the Middle East. 27% of these changes belong to cropland, most of which is located in south-eastern Turkey, in the European and western part of the country, in few places near the Caspian Sea in Iran and in the Northern District of Israel.

The second biggest land-cover class that experienced that change is the grassland – 18.9% (Table A.7). This class is well visible over the central parts of Turkey and in the centre of Cyprus. However, probably the most outlined land-cover classes with this change are the evergreen needleleaf forest (13.1%) and the deciduous broadleaf forest (6%). The last two can be seen in the northern parts of Turkey, but are mostly well grouped near the Caspian Sea coastline in Iran, where they form a long strip. As seen in Table 7.7, the magnitude of these changes affected significant percentage of the area of all forest land-cover classes and almost 10% of all cropland in the Middle East.

**Table 7.7: Affected land-cover by NDVI change between 0.1 and 0.2 units in the Middle East (1982 – 2010).**

<b>Land Cover</b>	<b>%</b>	<b>Land Cover</b>	<b>%</b>
Evergreen Needleleaf Forest	19.0	Closed Shrubland	0.8
Deciduous Broadleaf Forest	21.5	Open Shrubland	0.8
Mixed Forest	19.3	Grassland	3.5
Woodland	5.8	Cropland	9.8
Wooded Grassland	7.5	Bare Ground	0.3

The largest as a magnitude and most significant negative NDVI changes in the Middle East are between -0.2 and -0.3 units. They affected just 0.01% of the NDVI trends in the region. Moreover, this land includes four land-cover classes – cropland, open shrubland, grassland and a small area of bare ground. Out of these, the cropland changed the most as a proportion from its total area. The same percentage in the remaining classes is less than 0.1%.

The more significant positive changes were with greater scale and more dramatic than the negative ones. 0.05% of the temporal NDVI trends in the region experienced change in a positive direction between 0.2 and 1 units. The land-cover affected by that magnitude of change include: grassland, cropland, open shrubland, as well as small areas of bare ground, wooded grassland and closed shrubland. However, as seen in Table 7.8, as a proportion from the total area of a class, this magnitude of change influenced mostly the wooded grassland.

**Table 7.8: Affected land-cover by NDVI change between 0.2 and 1 unit in the Middle East (1982 – 2010).**

<b>Land Cover</b>	<b>%</b>
Wooded Grassland	0.4
Closed Shrubland	0.2
Open Shrubland	< 0.1
Grassland	0.1
Cropland	0.2
Bare Ground	< 0.1

The change in 92.4% of the temporal NDVI trends is gradual, while in 0.05% it is abrupt. Most of the abrupt changes are in Turkey and Iran, but few are also visible on the southern coastline of Cyprus, and in Egypt (Figure 7.4). The land-cover classes associated with the abrupt changes are mainly: grassland (19%), cropland (14.2%), open shrubland (13.4%) and wooded grassland (12%).

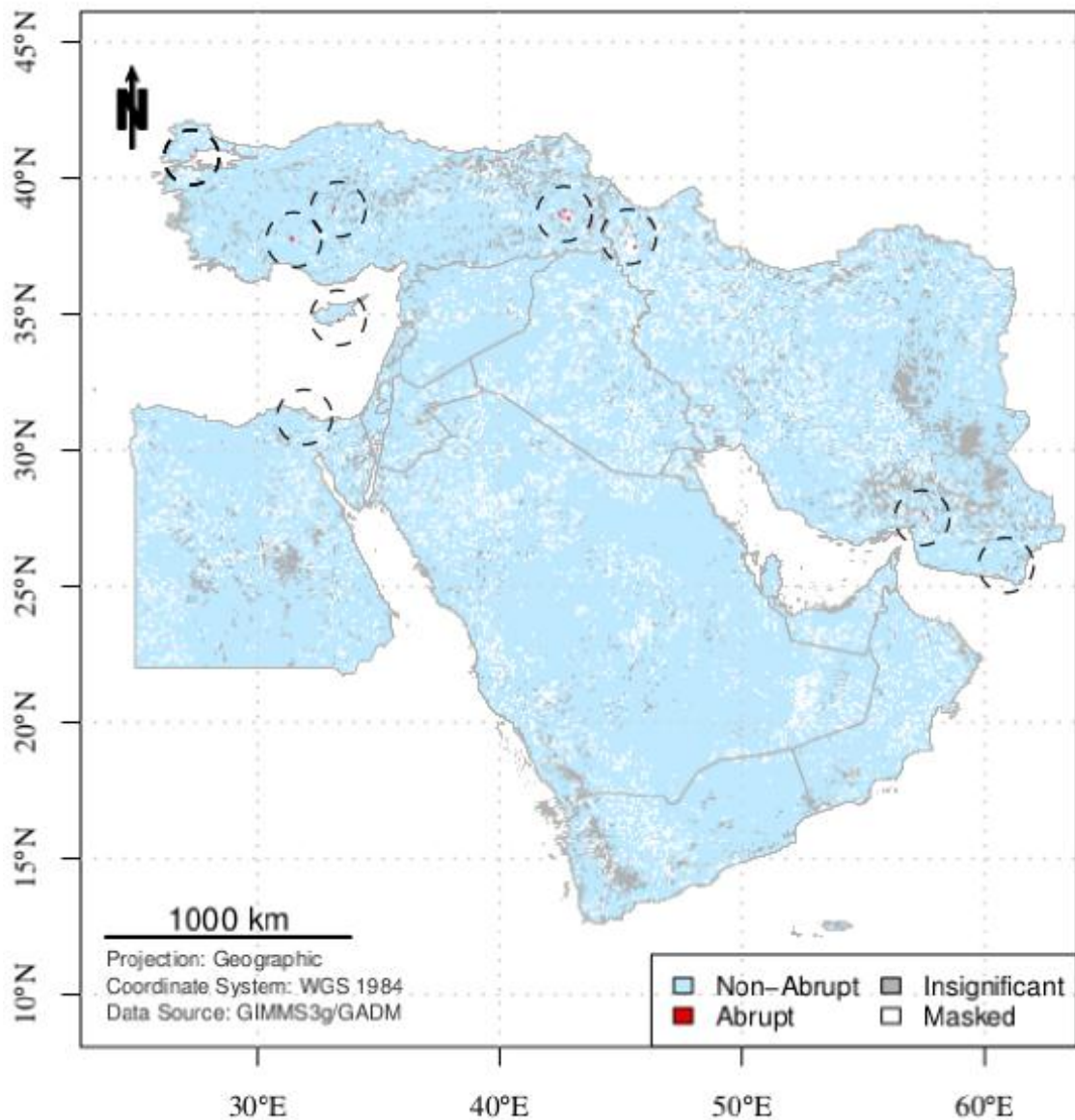
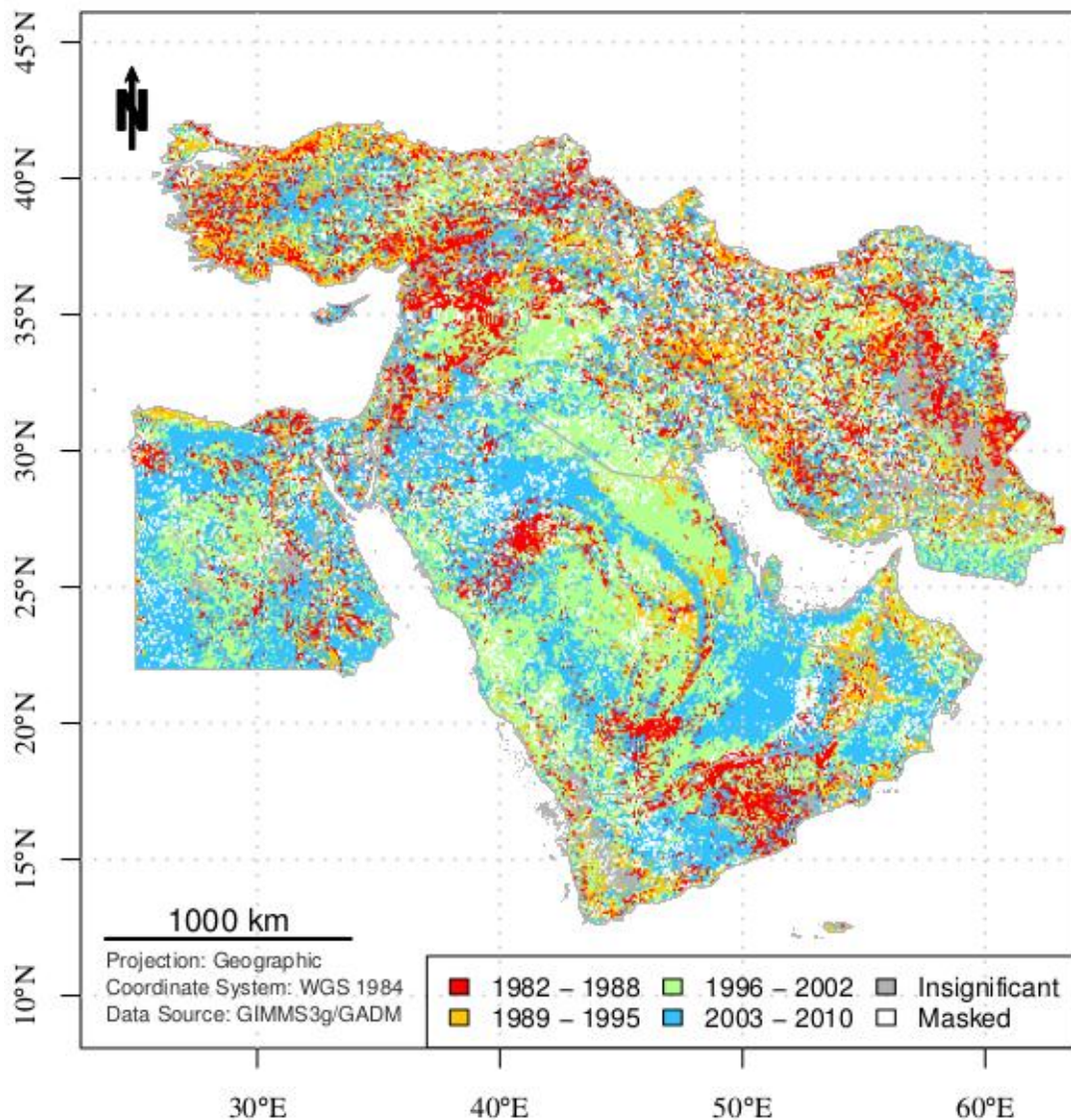


Figure 7.4: Type of the greatest NDVI change in the Middle East (1982 – 2010).

### 7.2.2 Beginning of the greatest temporal NDVI changes

The greatest change for 33.7% of the temporal NDVI trends in the Middle East started between years 2003 and 2010 (Figure 7.5). The data produced by MELCA indicated that 60% of this land is a bare ground, 26.2% open shrubland and 8% grassland. Similar proportions can be found in the second significant period during which the land-cover in the Middle East started to change, namely between 1996 and 2002, 29.7% of the land began to change. However, 64.4% of that land is classified as a bare ground, 25.2% as open shrubland and 5.6% as grassland.



**Figure 7.5: Start of the greatest NDVI change in the Middle East (1982 – 2010).**

Another look over the data showed that as a proportion out of the total area of a class, the most significant change for most of the vegetation land-cover began between 1982 – 1988 (16%) and 1989 - 1995 (13.1%). The additional calculations showed that the change started in the same period for more than half of all forests and woods. Most of these land-cover classes are located in the northern region of Turkey, but mainly in the Black Sea and Marmara regions of the country. The remaining forests are located in the northern region of Iran close to the Caspian Sea.

Moreover, the outputs from MELCA pointed that the affected area of closed shrubland is larger in these periods. Patches of closed shrublands that experienced the change in these periods can be noticed in several places in the Middle East. However, the most significant cluster of this land-cover is located near the Jabal an-Nabi Shu'aib in Yemen.



### 7.2.3 Duration of the greatest temporal NDVI changes

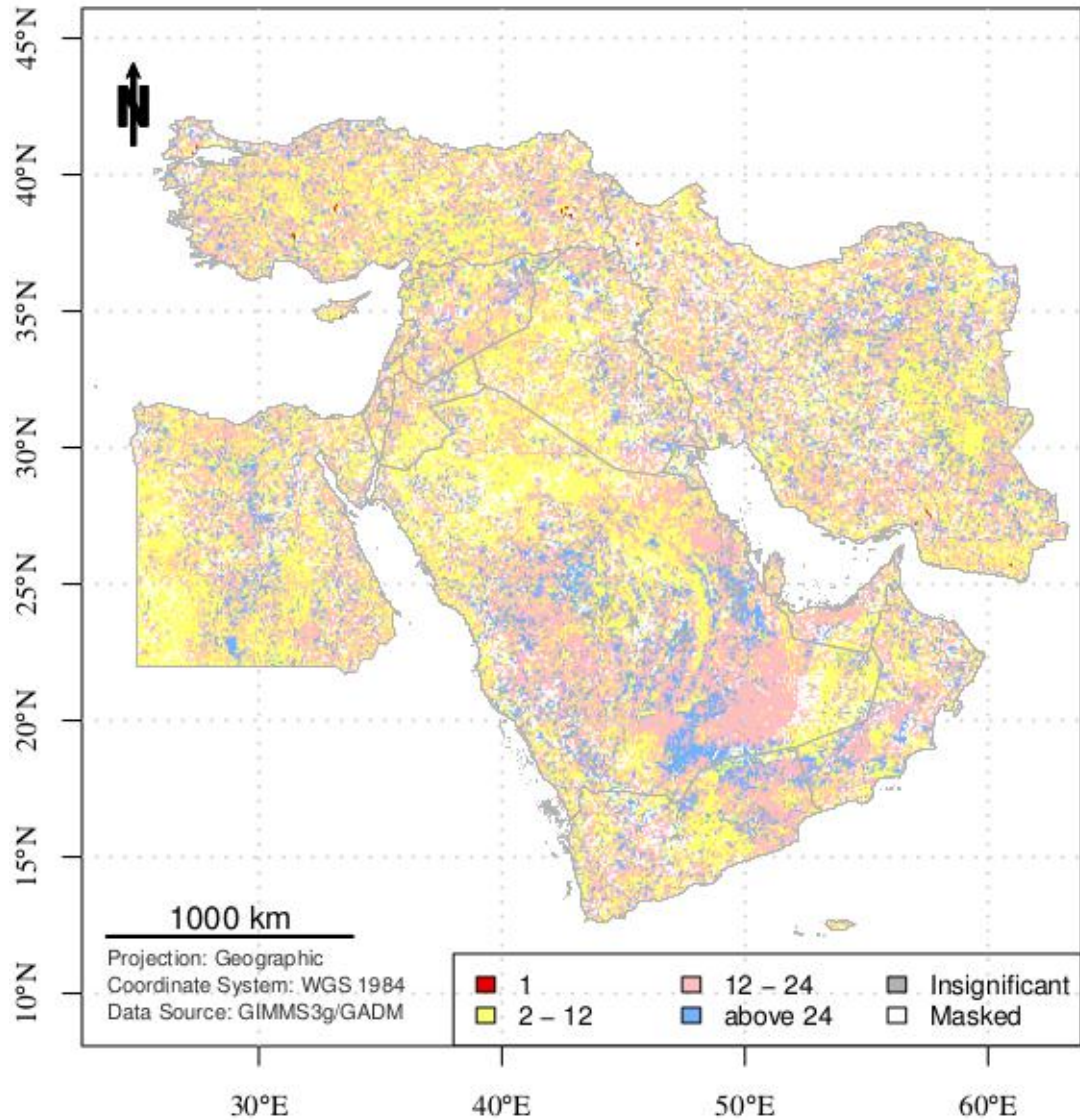


Figure 7.6: Duration of the greatest NDVI change in the Middle East (1982 – 2010).

The duration of the abrupt change was up to 1 month in all affected trends (Figure 7.6). The change duration in most of the Middle East (41.1%) is between 12 and 24 months. However, 52% of this land is classified as a bare ground. The other significant land-cover classes include: open shrubland (29.6%), grassland (8.8%), cropland (4.4%) and wooded grassland (1.5%).

In 37.6% of the trends, the greatest change lasted between 2 and 12 months. Moreover, in 13.7 % of the trends, the change was above 24 months. The ratio and percentages of the land-cover classes from the last two periods is very similar to the period that lasted between 12 and 24 months.

The duration of the greatest changes from the total area of a class is almost equally distributed. However, the change duration of 12 – 24 months was more significant for the closed shrubland, the wooded grassland and the deciduous broadleaf forest. The change with duration of 2 – 12 months affected most of the cropland and the evergreen needleleaf forest.





## **8. Discussion**

### ***8.1 Temporal NDVI trends without vegetation***

A large proportion of the Middle East region is bare ground, which is without any vegetation. However, the META DBEST algorithm had analyzed this territory and it is included in the results. Moreover, the temporal NDVI changes in just a part of the Middle East are classified as insignificant, while most of this land is shown to have changed with values between -0.1 to 0.1 units. Many of these small values of change may be due to noise in the measurements, illuminations or other factors related to the data that are not in the scope of this project.

However, a filtration of the data can be introduced in META DBEST algorithm, the MELCA algorithm or in a separate procedure. On one side, the introduction of a filter prior to the DBEST execution may prevent one of its major advantages – the detection of abrupt changes. Such scenario might be the planting of new crops fields or the development of irrigation systems in a dry region. Likewise, other variables of interest outputted by the DBEST might be distorted by such filter.

On the other side, the processing of the output from the algorithm can also lead to losses of important information. The MELCA algorithm uses the UMD land-cover data set that is only up to year 1994, while the temporal NDVI data goes up to year 2010. In other words, a straight universal approach for resolving this issue may not exist and any introduction of data filters or an additional data processing of the outputs from DBEST should be made according to the given scenario.

### ***8.2 Major patterns and relationships***

Linear, quadratic and cubic trends of the green season in the Middle East were found in each land-cover class. The major three types of vegetation that constitutes each of the trend types are the same – open shrubland, grassland and cropland. The last three are the most common types of vegetation in the Middle East. Moreover, the proportions which they occupy in each trend type are consistent with the proportions of the land-cover in the region. The above points out, that the majority of the analysed trends are not related to a specific land-cover class.

However, the analysed trends affected different proportions of each land-cover class. The calculations show, that the mixed forest are the most influenced by linear, quadratic and cubic trends of the green season. This type of land-cover is mainly located close to the Black Sea in Turkey. Likewise, most of the mixed forest trends were found to be in a negative direction. In addition, the evergreen needleleaf forest and the deciduous broadleaf forest are also mainly located in Turkey, and are also significantly affected by green season trends in a negative direction. However, whether climatic, regional economic or other factors contributed to this relation remains unclear in this research.

The analysis of the greatest temporal NDVI changes doesn't show any forest to be affected by an abrupt change and by high values of change. The extreme negative values of change (between -0.2 and -0.3 units) are mostly related to the croplands. However, approximately two times more cropland was affected by high values of positive change (between 0.2 and 1 unit). The last lead to the suggestion, that while in some regions the agriculture expanded, in other regions it declined.

The grassland is the other significantly affected land-cover class by extreme values of change. It is the most affected land-cover class by high values of change as a territory. Moreover, the grassland and the cropland are often bordering land-cover classes in the Middle East. That is why one suggestion could be that, those high values of change in the grassland are related to a process of cultivation or expansion of the cropland in the region.

The spatial distribution of the high values of the greatest NDVI change is mostly in the northern part of the Middle East. Clusters of high positive values of change are well outlined along the Caspian Sea in Iran. They are also well visible in Turkey, in the northern region of Syria and Iraq. However, in Turkey and Iraq, some of the lands affected by high values of positive change are near lands affected by extreme values of negative change. The assumption for Turkey is, that this is may be a process of clearance due to an agricultural expansion. However, the same assumption can't be made for Iraq, since most of its territory is desert. Most probably the negative temporal NDVI changes in the northern part of Iraq are a process of agricultural decline.

One of the most interesting results in this work is the relatively small areas with high values of change magnitude. These areas are to be found mainly in the northern region of the Middle East, probably due to the fact that the most of the vegetation is located there. However, though the algorithms computed their locations and land-cover, what remained unclear is what have been the events that triggered these changes. The last also represents a barrier for the relation of this research with some of the other reviewed studies in the Middle East.

Al-Bakri et al. (2001), Bitter & Sofer (2013) and Guneroglu (2015) reported expansion of urban areas in various locations. Some of the found spots that were subject to an abrupt change or to changes with high values seem to be located near cities. However, the scale of this study doesn't allow confirmation of such findings. Moreover, it is not clear if this process is a result from the expansion of the cities or from other activities that are taking place near the cities, such as agricultural development.

Al-Bakri et al. (2001), Bitter & Sofer (2013) and Wafi Al-Fares (2013) reported expansion of the agricultural fields in various places in the Middle East. However, the investigation of transition of one land-cover to another was not a subject of this work. What was found independently by both algorithms was that a significant part of the cropland in the Middle East has been subject of changes between 1982 – 2010. Likewise, at the end, most of the cropland NDVI trends were in negative direction as a whole.

Significant changes in the cropland in Iraq can be seen for the same periods as the one reported by Gibson et al. (2015). Similar to the research findings, the output from the META DBEST algorithm shows negative changes in the cropland in the Kurdish Autonomous Region. Likewise, the green period trends computed by the META PolyTrend algorithm are in negative direction for this region. Few relatively small patches of cubic trends in negative directions are also visible in the same location over the computed maps.

### ***8.3 Automated temporal NDVI analysis***

META PolyTrend, META DBEST and MELCA were developed in such way, that the algorithms can perform a temporal NDVI analysis for different region or time-period. The reconfiguration of the algorithms was made as easy as possible and it can be done in minutes. All this allows rapid automated temporal NDVI analysis, which will generate data sets from the outputs of DBEST or PolyTrend, but as well as ready maps and area calculations.

Because of the open nature of the R language, the algorithms can be modified and improved for specific tasks. An interesting concept would be the development of additional functions that can handle other types of temporal data, since both DBEST and PolyTrend are not limited to the NDVI trends.

### ***8.4 Sources of errors***

The UMD land-cover data set is developed for the period between the years 1981 and 1994, while the NDVI3g data analyzed by PolyTrend and DBEST are for the period between the years 1982 and 2010. Besides, the UMD land-cover classification entails some degree of uncertainty, which varies in the different land-cover classes (Table 5.3). That is why some of the calculated results might be inaccurately associated with a given land-cover class.

The procedures prior the overlay analysis should also be taken into consideration. The grids produced by META PolyTrend and META DBEST are with a resolution of 0.0833 degrees, while the UMD land-cover data set has a resolution of 0.08 degrees. In order to estimate the affected land-cover, the MELCA algorithm resamples the UMD land-cover data set to the resolution of the resulting data sets by the use of the nearest neighbor technique. This, however, introduces a false accuracy of the computations, since the UMD land-cover data are resampled to a higher resolution.

Last, but not least, the interpretations of the spatial distribution of the results produced by the algorithms are made through a visual inspection. That raises some concerns, especially when it comes to the locations of smaller administrative units and geographical places.

## ***8.5 Strengths and weaknesses of the study***

Due to the date of this work, the studies that analyze the vegetation trends in the Middle East region are focus only over specific places. However, the climatic and the socio-political factors may often have an impact over the ecosystems or the agriculture on a regional scale. That is why the computed data by the algorithms in this study have a significant value. Moreover, the developed algorithms produced more than 40 data sets with information related to the trends in the vegetation in the Middle East region. These data sets form a significant base for various types of investigations. However, the interpretations of the produced information by the algorithms in the current research are subjective. Likewise, the research does not provide details of what are the potential causes that led to the appearance of an observed pattern or a type of trend. Besides, with the exception of the abrupt changes, the geographical locations of the greatest temporal NDVI changes with extreme values are not presented in the analysis, since their spatial coverage is too small in order to be visually observed.

## ***8.6 Future work***

PolyTrend and DBEST can be published as an R package, which will make them available to a wider audience. This will speed up the further development of the algorithms, because more people will be able to test both algorithms in various scenarios. A possible step in this direction will be the development of a manual for both algorithms. Moreover, PolyTrend and DBEST should be presented for a discussion to some of the R language groups, which have special interest in the development of packages for spatial analysis and geographic data (e.g. the R-sig-Geo group).

In the presented R version of DBEST in this work, the source code of the algorithm follows strictly a mathematical model without taking the full advantage of the specific of the R language. The future development of DBEST for R language can improve the speed of the algorithm, since some of the loops could be avoided through a vectorization of the functions (i.e. by the use of `apply`, `sapply` and `lapply`).

The future development of META PolyTrend and META DBEST includes the unification of both algorithms and the extension of their flexibility. This includes steps, such as the development of additional functions for handling other NDVI data set products and improvement of the flexibility related to the spatial extent (e.g. options for selection of lower administrative units).

## 9. Conclusions

This study demonstrated the great potential of R language for the remote sensing by the development of several procedures that automated the process of temporal NDVI analysis. These procedures applied the two algorithms – DBEST and PolyTrend over GIMMS3g data set in the Middle East for the period 1982 – 2010.

In the process of development, the last two algorithms were translated in R language and successfully tested. In addition, their interfaces and output were modified according to the R environment characteristics in order to benefit other researchers too. A third procedure demonstrated how the resulted data sets could be expanded with more information computed by an overlay analysis with land-cover data set.

As a result, the study found that the spatial coverage of the obtained green season trends consists of 3.8% linear trends, 1.8% quadratic trends and 3.2% cubic trends. Regarding the associated land-cover, 41.7% of the linear trends are open shrubland. The quadratic and cubic trends associated with this land-cover class are 44.8% and 20.5%, respectively. In most of the cases each of the trend types was found to be spatially grouped. Moreover, the visual inspection showed that one type of group often borders with another. Significant clusters of linear trends were located along the northern border of Yemen, where they are adjacent clusters of quadratic and cubic trends near the area where Yemen, Oman, and Saudi Arabia join borders. Most of the quadratic trends were found to be grouped along the Tihamah coastal plain and closer to the Red Sea. Large clusters of cubic trends were located in the northern and central regions of Saudi Arabia.

The study concludes that the trends are almost equally distributed in each type of vegetation, but mostly affected the mixed forest (4.1% by linear trends and 5.2% by cubic trends, respectively). Besides, 51% of the trends are in a negative direction and 49% in a positive one. The research also estimated that the trends with negative directions are significantly dominant in all forest types in the Middle East.

The investigated greatest temporal NDVI changes pointed that most of the abrupt changes (0.05% of the trends) are located in Turkey and Iran, but few were also found on the southern coastline of Cyprus, and in Egypt. The largest as a magnitude negative NDVI changes (0.01% of the trends) were found to be between -0.2 and -0.3 units, while the largest positive changes (0.05% of the trends) were estimated between 0.2 and 1 units. Most of the land-cover associated with the extreme NDVI changes was related to grassland and cropland. The results also show that most of the changes in the vegetation land-cover began and ended in the first half of the studied period (1982 – 1995), as well as that the duration of the changes for most of the vegetation lasted between 12-24 months.

Likewise, the computations indicated that in general the NDVI trends associated with cropland and grassland experienced significant shifts in both directions. This led to the suggestion that in some regions the agriculture expanded, while in other regions it declined. The maps and the analysis in the study associated such process of agricultural decline with the northern region of Iraq.



## 10. References

- Al-Bakri, J, Taylor, J, & Brewer, T 2001, 'Monitoring Land Use Change in the Badia Transition Zone in Jordan Using Aerial Photography and Satellite Imagery', *The Geographical Journal*, 3, p. 248, JSTOR Journals, EBSCOhost, viewed 15 October 2015.
- Al-Fares, W 2013, *Historical Land Use/Land Cover Classification Using Remote Sensing. [Elektronisk Resurs] : A Case Study Of The Euphrates River Basin In Syria*, n.p.: Heidelberg : Springer International Publishing : Imprint: Springer, 2013., Library catalogue (Lovisa), EBSCOhost, viewed 15 October 2015.
- Alcantara, C, Kuemmerle, T, Prishchepov, A, & Radeloff, V 2012, 'Mapping abandoned agriculture with multi-temporal MODIS satellite data', *Remote Sensing Of Environment*, 124, pp. 334-347, Inspec, EBSCOhost, viewed 16 April 2015.
- AQUASTAT-FAO, 2008. 'Irrigation in the Middle East region in figures: AQUASTAT Survey - 2008.' Available online at: <http://noaasis.noaa.gov/NOAASIS/ml/avhrr.html> (accessed 1 April 2015).
- Bittner, C, & Sofer, M 2013, 'Land use changes in the rural–urban fringe: An Israeli case study', *Land Use Policy*, 33, pp. 11-19, ScienceDirect, EBSCOhost, viewed 15 October 2015.
- Bivand, R, Pebesma, E, & Gómez-Rubio, V 2013, *Applied Spatial Data Analysis With R. [Elektronisk Resurs]*, n.p.: New York, NY : Springer New York : Imprint: Springer, 2013., Library catalogue (Lovisa), EBSCOhost, viewed 29 March 2015.
- Chen, L, Michishita, R, & Xu, B 2014, 'Abrupt spatiotemporal land and water changes and their potential drivers in Poyang Lake, 2000–2012', *ISPRS Journal Of Photogrammetry And Remote Sensing*, 98, pp. 85-93, ScienceDirect, EBSCOhost, viewed 16 April 2015.
- Culcasi, K 2012, 'Mapping the Middle East from Within: (Counter-)Cartographies of an Imperialist Construction', *Antipode*, 44, 4, pp. 1099-1118, Academic Search Complete, EBSCOhost, viewed 31 March 2015.
- Davison, Roderic H. 1960. 'Where is the Middle East?'. *Foreign Affairs* 38 (4): 665–75.
- De Fries, R., Hansen, M., Townshend, J., & Sohlberg, R. (1998), 'Global land cover classifications at 8 km spatial resolution: the use of training data derived from Landsat imagery in decision tree classifiers', *International Journal Of Remote Sensing*, 19, 16, pp. 3141-3168, EBSCOhost, viewed 16 March 2016.
- ESA, 2009. 50 years of Earth Observation. Available online at: [http://www.esa.int/About\\_Us/Welcome\\_to\\_ESA/ESA\\_history/50\\_years\\_of\\_Earth\\_Observation](http://www.esa.int/About_Us/Welcome_to_ESA/ESA_history/50_years_of_Earth_Observation) (accessed 29 March 2015).
- EUMETSAT , 2015. An Introduction to EUMETSAT Polar System. Available online at: [http://www.eumetsat.int/eps\\_webcast/eps/print.htm](http://www.eumetsat.int/eps_webcast/eps/print.htm) (accessed 29 March 2015).
- FAO, 2000. 'Global Forest Resources Assessment 2000. Chapter 21. West Asia.' Available at: <http://www.fao.org/docrep/004/y1997e/y1997e0q.htm>. (Accessed 1 April 2015).
- Foody, G 2003, 'Geographical weighting as a further refinement to regression modelling: An example focused on the NDVI-rainfall relationship', *Remote Sensing Of Environment*, 88, 3, p. 283-293, Scopus®, EBSCOhost, viewed 27 March 2015.

- GADM, 2015. "Global Administrative Areas". Available online at: <http://www.gadm.org/> (Accessed 27 October 2015).
- Gibson, G, Campbell, J, & Zipper, C 2015, 'Sociopolitical influences on cropland area change in Iraq, 2001–2012', *Applied Geography*, 62, pp. 339-346, Academic Search Complete, EBSCOhost, viewed 15 October 2015.
- GLCF, 2014. "Global Inventory Modeling and Mapping Studies (GIMMS)". Available online at: <http://glcf.umd.edu/data/gimms/index.shtml> (Accessed 10 April 2015).
- Goslee, S 2011, 'Analyzing remote sensing data in R: The landsat package', *Journal Of Statistical Software*, 43, 4, p. 1-25, Scopus®, EBSCOhost, viewed 28 March 2015.
- Griffiths, P, Kuemmerle, T, Kennedy, R, Abrudan, I, Knorn, J, & Hostert, P 2012, 'Using annual time-series of Landsat images to assess the effects of forest restitution in post-socialist Romania', *Remote Sensing Of Environment*, 118, pp. 199-214, Inspec, EBSCOhost, viewed 16 April 2015.
- Grogan, K, Pflugmacher, D, Hostert, P, Kennedy, R, & Fensholt, R 2015, 'Cross-border forest disturbance and the role of natural rubber in mainland Southeast Asia using annual Landsat time series', *Remote Sensing Of Environment*, ScienceDirect, EBSCOhost, viewed 16 April 2015.
- Guneroglu, A 2015, 'Coastal changes and land use alteration on Northeastern part of Turkey', *Ocean And Coastal Management*, ScienceDirect, EBSCOhost, viewed 15 October 2015.
- Hansen, M., R. DeFries, J.R.G. Townshend, and R. Sohlberg (1998), *UMD Global Land Cover Classification, 1 Kilometer, 1.0*, Department of Geography, University of Maryland, College Park, Maryland, 1981-1994.
- Hastings, David A., and William J. Emery, 1992. *The Advanced Very High Resolution Radiometer (AVHRR): A Brief Reference Guide*. *Photogrammetric Engineering and Remote Sensing*, vol. 58, No. 8, August 1992, pp. 1183-1188.
- Huete, A, Didan, K, Miura, T, Rodriguez, E, Gao, X, & Ferreira, L 2002, 'Overview of the radiometric and biophysical performance of the MODIS vegetation indices', *Remote Sensing of Environment*, vol. 83, no. 1, pp. 195-213.
- Ihaka, Ross 1998. *R : Past and Future History (Technical report)*. Statistics Department, The University of Auckland, Auckland, New Zealand.
- IPCC, 2001. 'Chapter 7: Middle East and Arid Asia. IPCC Special Report on The Regional Impacts of Climate Change: An Assessment of Vulnerability.' Available online at: [http://www.grida.no/publications/other/ipcc\\_sr/?src=/climate/ipcc/regional/index.htm](http://www.grida.no/publications/other/ipcc_sr/?src=/climate/ipcc/regional/index.htm) (accessed 1 April 2015).
- Jamali, S, Eklundh, L, Ardö, J, Seaquist, J, & Jönsson, P 2015, 'Detecting changes in vegetation trends using time series segmentation', *Remote Sensing Of Environment*, 156, p. 182-195, Scopus®, EBSCOhost, viewed 27 March 2015.
- Jamali, S, Seaquist, J, Eklundh, L, & Ardö, J 2014, 'Automated mapping of vegetation trends with polynomials using NDVI imagery over the Sahel', *Remote Sensing Of Environment*, 141, pp. 79-89, ScienceDirect, EBSCOhost, viewed 13 October 2015.
- Jönsson, P, & Eklundh, L 2002, 'Seasonality extraction by function fitting to time-series of satellite sensor data', *IEEE Transactions On Geoscience And Remote Sensing*, 40, 8, p. 1824-1832, Scopus®, EBSCOhost, viewed 29 March 2015.



- Jonsson, P, & Eklundh, L 2004, 'TIMESAT; a program for analyzing time-series of satellite sensor data', *Computers & Geosciences*, 30, 8, pp. 833-845, GeoRef, EBSCOhost, viewed 15 April 2015.
- Kennedy, R, Zhiqiang, Y, & Cohen, W 2010, 'Detecting trends in forest disturbance and recovery using yearly Landsat time series: 1. LandTrendr - Temporal segmentation algorithms', *Remote Sensing Of Environment*, 114, 12, pp. 2897-2910, Inspec, EBSCOhost, viewed 15 April 2015.
- Khorrarn, S, Nelson, S, Koch, F, & van der Wiele, C 2012, *Remote Sensing*. [Elektronisk Resurs], n.p.: Boston, MA : Springer US, 2012., Library catalogue (Lovisa), EBSCOhost, viewed 23 March 2015.
- Koppes, CR 1976, 'Captain Mahan, General Gordon, and the Origins of the Term 'Middle East'', *Middle Eastern Studies*, 1, p. 95, JSTOR Journals, EBSCOhost, viewed 1 April 2015.
- Kriegler, F.J., Malila, W.A., Nalepka, R.F., and Richardson, W. 1969. 'Preprocessing transformations and their effects on multispectral recognition.' *Proceedings of the Sixth International Symposium on Remote Sensing of Environment*, p. 97-131.
- Kun, J, Shunlin, L, Xiangqin, W, Yunjun, Y, Yingru, S, Bo, J, & Xiaoxia, W 2014, 'Land Cover Classification of Landsat Data with Phenological Features Extracted from Time Series MODIS NDVI Data', *Remote Sensing*, 6, 11, pp. 11518-11532, Inspec, EBSCOhost, viewed 16 April 2015.
- Lambin, E, & Ehrlich, D 1997, 'Land-cover changes in Sub-Saharan Africa (1982-1991): Application of a change index based on remotely sensed surface temperature and vegetation indices at a continental scale', *Remote Sensing Of Environment*, 61, 2, p. 181-200, Scopus®, EBSCOhost, viewed 29 March 2015.
- Lein, JK 2012, *Environmental Sensing*. [Elektronisk Resurs] : Analytical Techniques For Earth Observation, n.p.: New York, NY : Springer New York, 2012., Library catalogue (Lovisa), EBSCOhost, viewed 26 March 2015.
- Liang, L, Hawbaker, T, Chen, Y, Zhu, Z, & Gong, P 2014, 'Characterizing recent and projecting future potential patterns of mountain pine beetle outbreaks in the Southern Rocky Mountains', *Applied Geography*, 55, pp. 165-175, ScienceDirect, EBSCOhost, viewed 16 April 2015.
- NOAA, 2007. NOAA KLM User's Guide (April 2007 revision). Available online at: <http://www.ncdc.noaa.gov/oa/pod-guide/ncdc/docs/klm/html/c3/sec3-1.htm> (accessed 29 March 2015).
- NOAA, 2015. NOAA POES Status. Available online at: <http://www.ospo.noaa.gov/Operations/POES/status.html> (accessed 29 March 2015).
- NOAASIS, 2015. AVHRR NOAA Satellite Information System. Available online at: <http://noaasis.noaa.gov/NOAASIS/ml/avhrr.html> (accessed 29 March 2015).
- Novikova, G. 2000. "Armenia and the Middle East". *Middle East Review of International Affairs*. Available online at: <http://www.gloria-center.org/meria/2000/12/novikova.pdf> (accessed 1 April 2015).
- Özalp, ON 2011, 'Where is the Middle East? The Definition and Classification Problem of the Middle East as a Regional Subsystem in International Relations', *Turkish Journal Of Politics*, 2, 2, pp. 5-21, *Political Science Complete*, EBSCOhost, viewed 31 March 2015.
- Pinzon JE, Tucker CJ (2010) GIMMS 3g NDVI set and global NDVI trends. In:Second Yamal Land-Cover Land-Use Change Workshop. 8-10. March 2010

- Rovaniemi, Finland. Available online at:  
[http://www.geobotany.org/library/talks/PinzonJE2010\\_yamal\\_tal100308.pdf](http://www.geobotany.org/library/talks/PinzonJE2010_yamal_tal100308.pdf) (Accessed 10 April 2015).
- Pebesma, E.J., R.S. Bivand, 2005. Classes and methods for spatial data in R. R News 5 (2), <http://cran.r-project.org/doc/Rnews/>. (Accessed 10 April 2015).
- Pinzon, J, & Tucker, C 2014, 'A non-stationary 1981-2012 AVHRR NDVI3g time series', Remote Sensing, 6, 8, p. 6929-6960, Scopus®, EBSCOhost, viewed 9 April 2015.
- Pinzon, J., Brown, M.E. and Tucker, C.J., 2005. Satellite time series correction of orbital drift artifacts using empirical mode decomposition. In: N. Huang (Editor), Hilbert-Huang Transform: Introduction and Applications, pp. 167-186.
- R Foundation 2015. What is R?. Available online at: <http://www.r-project.org/about.html> (accessed 29 March 2015).
- Rouse, J.W., Jr., R.H. Haas, J.A. Schell, and D.W. Deering. 1973. Monitoring the vernal advancement and retrogradation (green wave effect) of natural vegetation. Prog. Rep. RSC 1978-1, Remote Sensing Center, Texas A&M Univ., College Station, 93p.
- Science Education through Earth Observation for High Schools (SEOS) Project, 2015. Available online at: <http://www.seos-project.eu> (accessed 13 October 2015).
- Sivakumar, M, Lal, R, Selvaraju, R, & Hamdan, I 2013, Climate Change And Food Security In West Asia And North Africa. [Elektronisk Resurs], n.p.: Dordrecht : Springer, 2013, Library catalogue (Lovisa), EBSCOhost, viewed 1 April 2015.
- Tucker, C.J., J. E. Pinzon, M. E. Brown, D. Slayback, E. W. Pak, R. Mahoney, E. Vermote and N. El Saleous (2005), An Extended AVHRR 8-km NDVI Data Set Compatible with MODIS and SPOT Vegetation NDVI Data. International Journal of Remote Sensing, Vol 26:20, pp 4485-5598.
- UMD, 2014. "GLCF: AVHRR Global Land Cover Classification". Available online at: <http://glcf.umd.edu/data/landcover/data.shtml> (Accessed 23 October 2015).
- USGS, 2015. "Remote Sensing Phenology" , Available online at: [http://phenology.cr.usgs.gov/ndvi\\_avhrr.php](http://phenology.cr.usgs.gov/ndvi_avhrr.php) (Accessed 10 April 2015).
- Verbesselt, J, Hyndman, R, Newnham, G, & Culvenor, D 2010, 'Detecting trend and seasonal changes in satellite image time series', Remote Sensing Of Environment, 114, pp. 106-115, ScienceDirect, EBSCOhost, viewed 15 April 2015.
- Wang, Q, Tenhunen, J, Dinh, N, Reichstein, M, Vesala, T, & Keronen, P 2004, 'Similarities in ground- and satellite-based NDVI time series and their relationship to physiological activity of a Scots pine forest in Finland', Remote Sensing Of Environment, 93, pp. 225-237, ScienceDirect, EBSCOhost, viewed 27 March 2015.
- Watts, L, & Laffan, S 2014, 'Effectiveness of the BFAST algorithm for detecting vegetation response patterns in a semi-arid region', Remote Sensing Of Environment, 154, pp. 234-245, Inspec, EBSCOhost, viewed 16 April 2015.
- Yufang, Y, Xiao, C, Xinwu, L, Lei, L, & Heygster, G 2012, 'Monitoring antarctic ice sheet melting periods with SSM/119H Ghz data and time series analysis', Inspec, EBSCOhost, viewed 16 April 2015.
- Zhangyan, J, Huete, A, Youngwook, K, & Didan, K 2007, '2-band enhanced vegetation index without a blue band and its application to AVHRR data', Proceedings Of The SPIE - The International

Society For Optical Engineering, 6679, 1, pp. 667905-1-9, Inspec, EBSCOhost, viewed 27 March 2015.

Zhaohui, X, Peijun, D, & Li, F 2014, 'Phenology-Driven Land Cover Classification and Trend Analysis Based on Long-term Remote Sensing Image Series', IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing, 7, 4, pp. 1142-1156, Inspec, EBSCOhost, viewed 16 April 2015.



## Appendix A. Selected outputs from the MELCA algorithm.

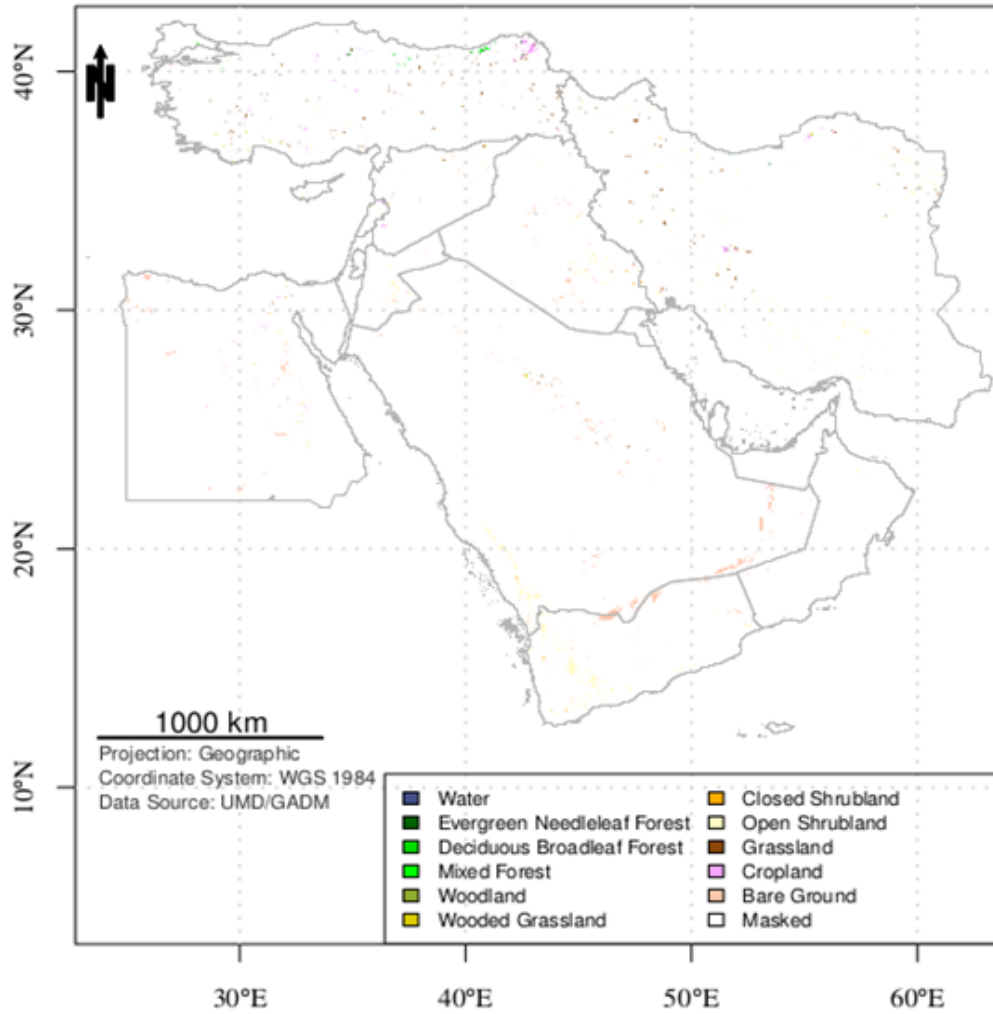


Figure A.1: Linear NDVI trends and land cover in the Middle East (1982 – 2010).

Table A.1: Linear NDVI trends and land cover in the Middle East (1982 – 2010).

Land Cover	km <sup>2</sup>	Land Cover	km <sup>2</sup>
Evergreen Needleleaf Forest	1126	Closed Shrubland	1379
Deciduous Broadleaf Forest	788	Open Shrubland	48552
Mixed Forest	519	Grassland	13271
Woodland	279	Cropland	9413
Wooded Grassland	1997	Bare Ground	38918

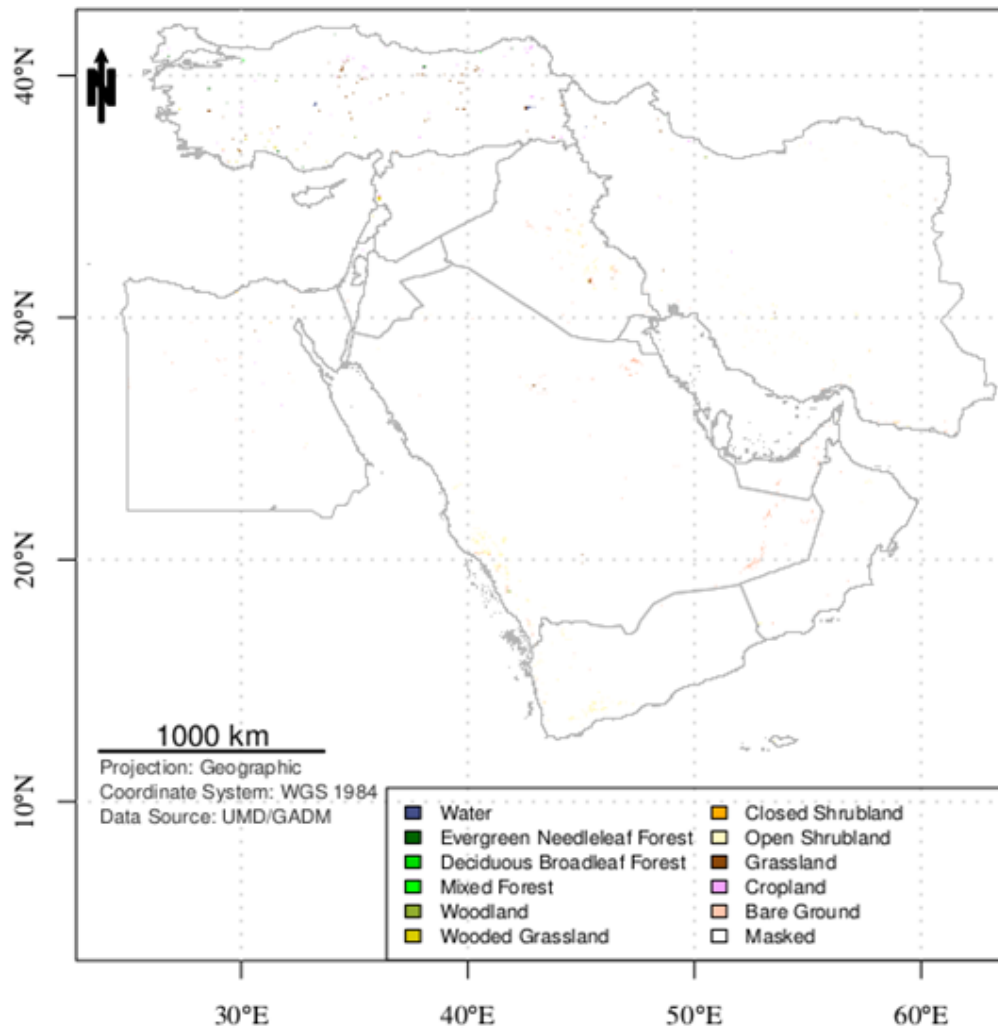


Figure A.2: Quadratic NDVI trends and land cover in the Middle East (1982 – 2010).

Table A.2: Quadratic NDVI trends and land cover in the Middle East (1982 – 2010).

Land Cover	km <sup>2</sup>	Land Cover	km <sup>2</sup>
Evergreen Needleleaf Forest	797	Closed Shrubland	672
Deciduous Broadleaf Forest	65	Open Shrubland	24847
Mixed Forest	130	Grassland	5740
Woodland	573	Cropland	3921
Wooded Grassland	1039	Bare Ground	17014

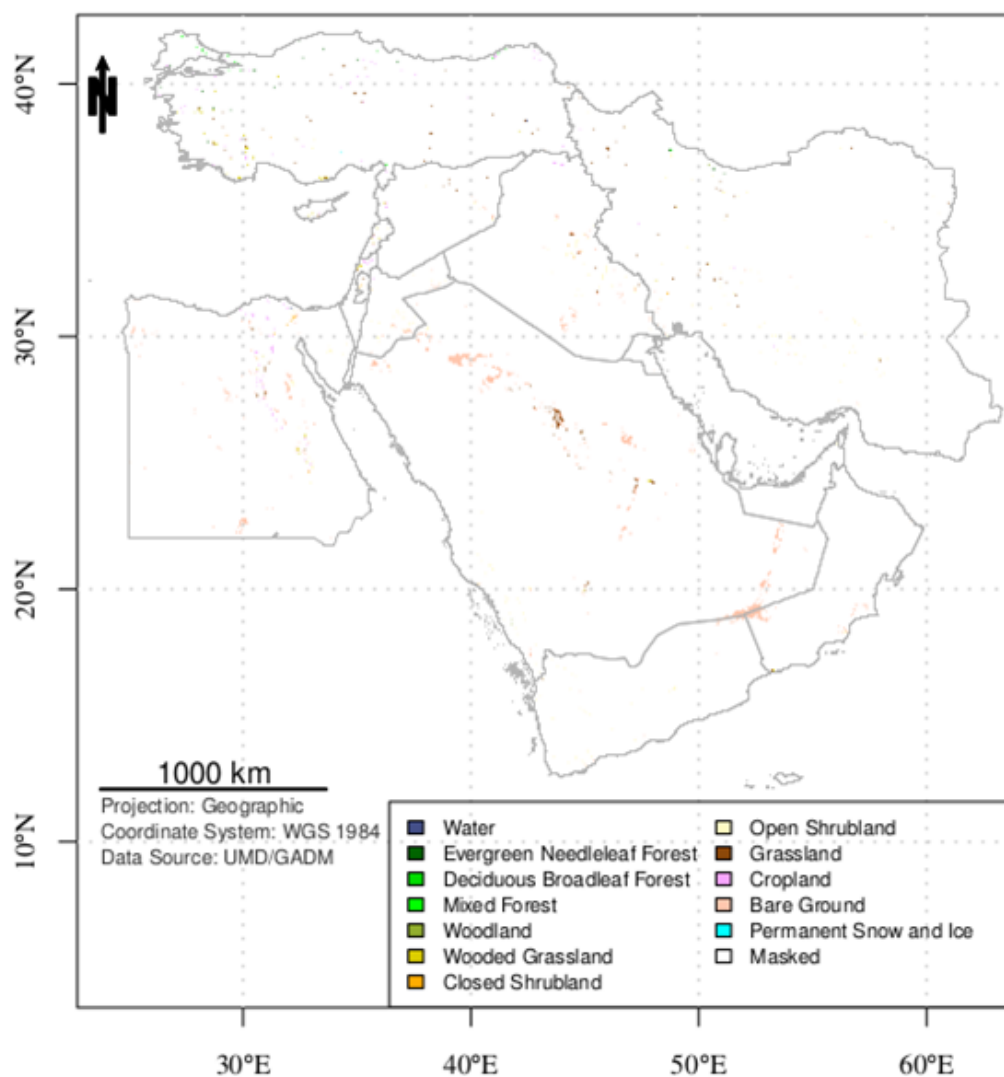


Figure A.3: Cubic NDVI trends and land cover in the Middle East (1982 – 2010).

Table A.3: Cubic NDVI trends and land cover in the Middle East (1982 – 2010).

Land Cover	km <sup>2</sup>	Land Cover	km <sup>2</sup>
Evergreen Needleleaf Forest	1261	Closed Shrubland	1244
Deciduous Broadleaf Forest	522	Open Shrubland	18831
Mixed Forest	655	Grassland	10539
Woodland	372	Cropland	8139
Wooded Grassland	3978	Bare Ground	52841

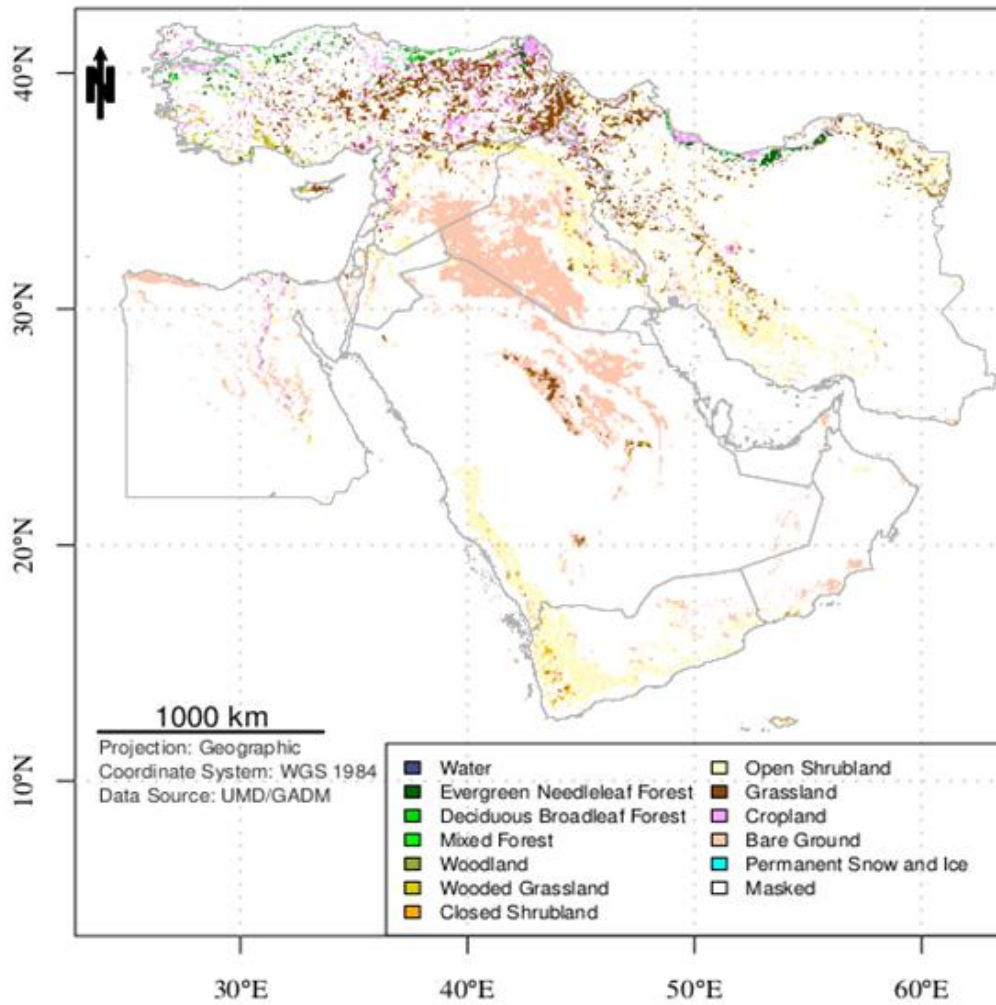


Figure A.4: Positive NDVI trends and land cover in the Middle East (1982 - 2010).

Table A.4: Positive NDVI trends and land cover in the Middle East (1982 - 2010).

Land Cover	km <sup>2</sup>	Land Cover	km <sup>2</sup>
Evergreen Needleleaf Forest	31252	Closed Shrubland	18691
Deciduous Broadleaf Forest	8973	Open Shrubland	513613
Mixed Forest	3153	Grassland	258202
Woodland	7995	Cropland	127694
Wooded Grassland	27439	Bare Ground	496815



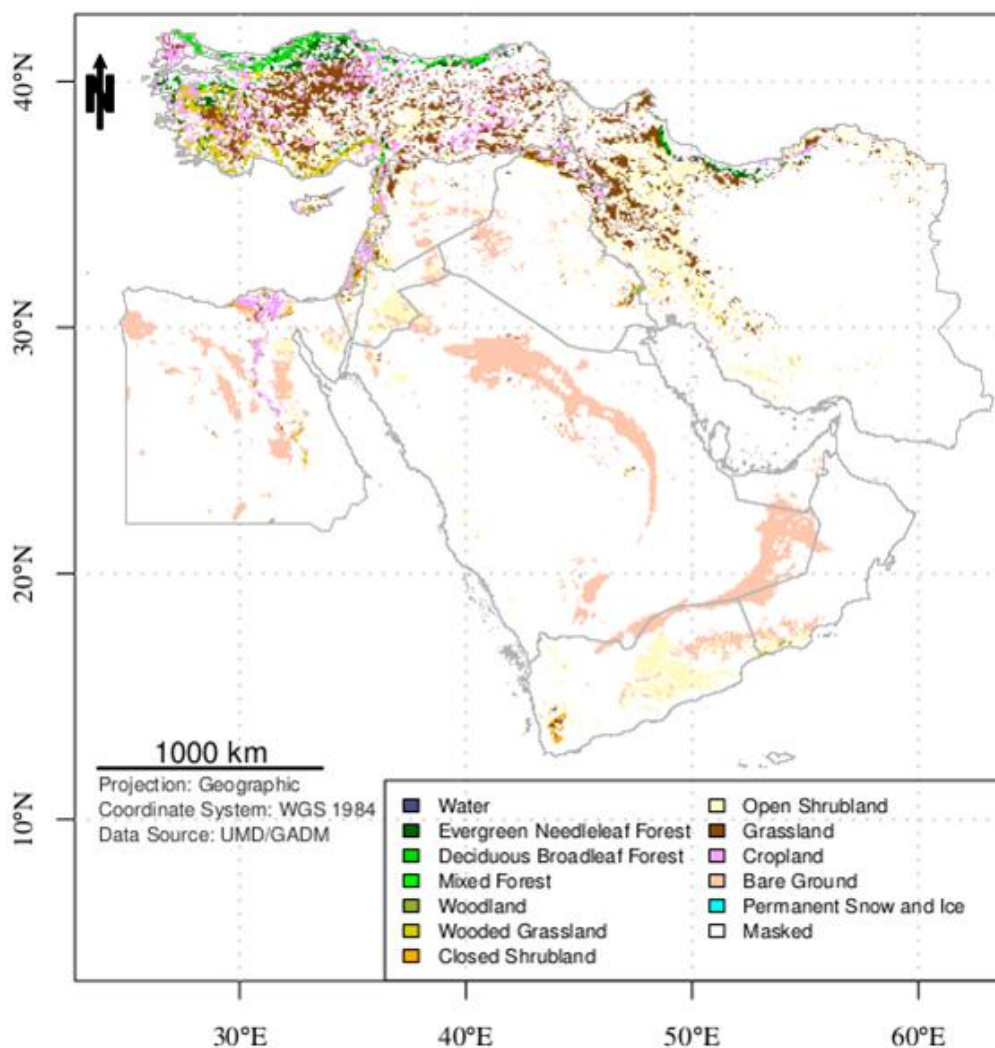
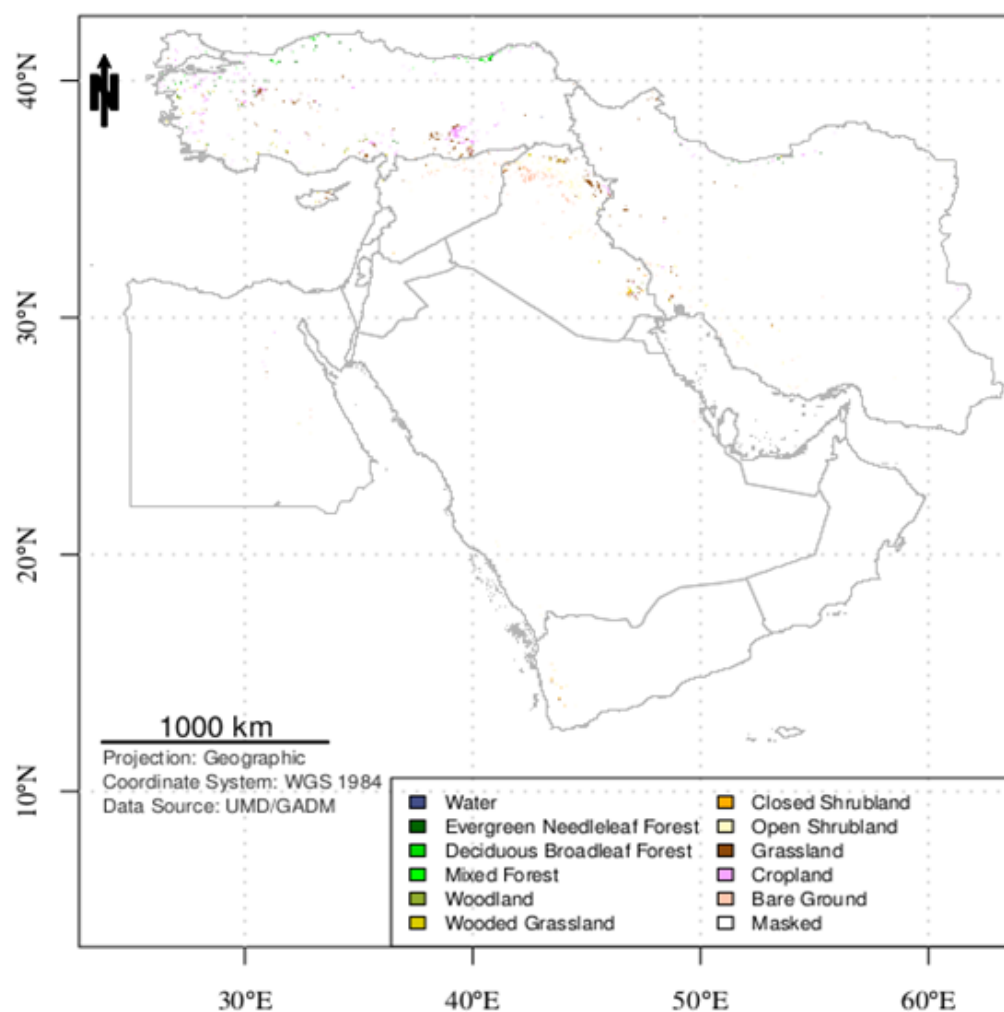


Figure A.5: Negative NDVI trends and land cover in the Middle East (1982 - 2010).

Table A.5: Negative NDVI trends and land cover in the Middle East (1982 - 2010).

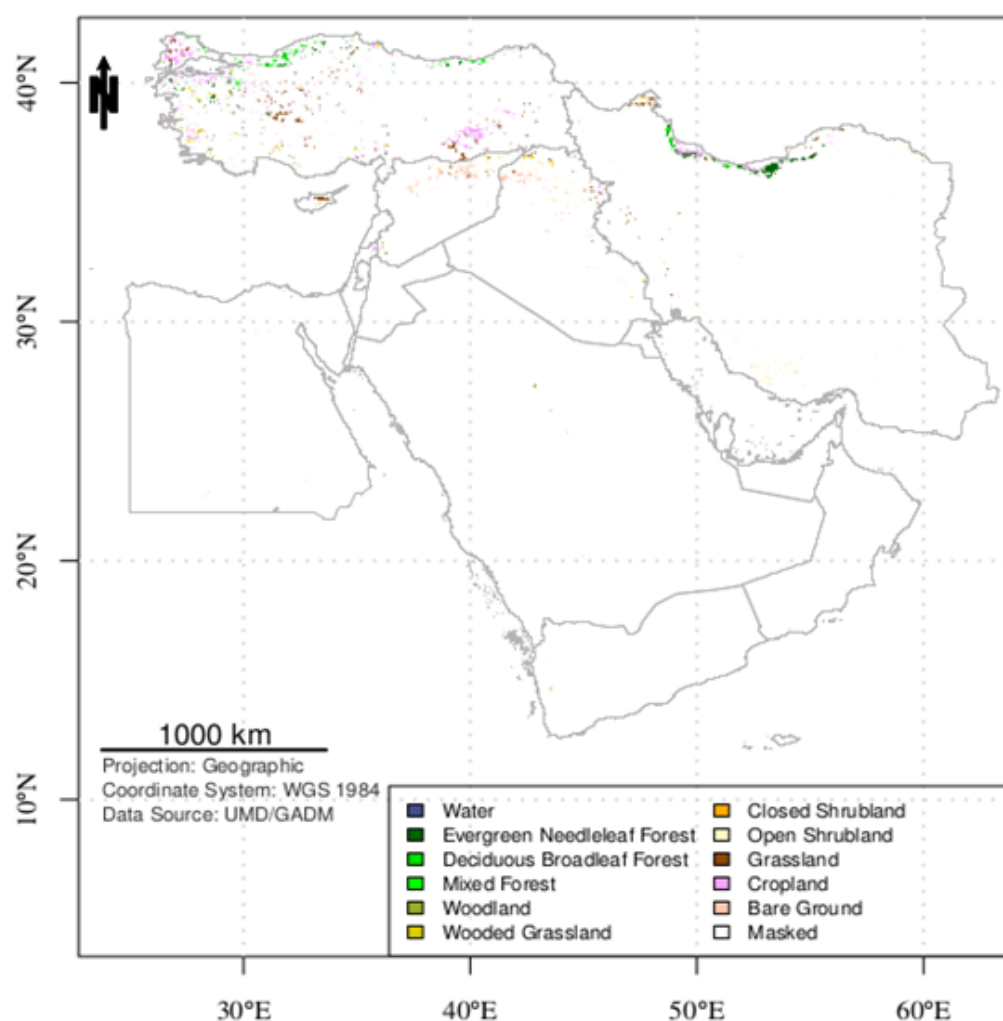
Land Cover	km <sup>2</sup>	Land Cover	km <sup>2</sup>
Evergreen Needleleaf Forest	48258	Closed Shrubland	17830
Deciduous Broadleaf Forest	22969	Open Shrubland	387180
Mixed Forest	7849	Grassland	309254
Woodland	11019	Cropland	188044
Wooded Grassland	61960	Bare Ground	499909



**Figure A.6: Magnitude of NDVI change value -0.2 - -0.1 and land cover in the Middle East (1982 – 2010).**

**Table A.6: Magnitude of NDVI change value -0.2 - -0.1 and land cover in the Middle East (1982 – 2010).**

Land Cover	km <sup>2</sup>	Land Cover	km <sup>2</sup>
Evergreen Needleleaf Forest	3272	Closed Shrubland	1155
Deciduous Broadleaf Forest	1691	Open Shrubland	14065
Mixed Forest	913	Grassland	13363
Woodland	1324	Cropland	15826
Wooded Grassland	3793	Bare Ground	8684



**Figure A.7: Magnitude of NDVI change value 0.1 – 0.2 and land cover in the Middle East (1982 – 2010).**

**Table A.7: Magnitude of NDVI change value 0.1 – 0.2 and land cover in the Middle East (1982 – 2010).**

Land Cover	km <sup>2</sup>	Land Cover	km <sup>2</sup>
Evergreen Needleleaf Forest	15453	Closed Shrubland	311
Deciduous Broadleaf Forest	7072	Open Shrubland	17722
Mixed Forest	2428	Grassland	22311
Woodland	1243	Cropland	31811
Wooded Grassland	6992	Bare Ground	11961



## Appendix B. Load region function.

Filename: loadRegion.R

---

```
loadRegion <- function(region.countries,data.date.start,
                        data.date.end,data.path) {

  message("Downloading countries borders...","\n",appendLF=FALSE)
  flush.console()

  # Load region countries data from GADM.org
  region.GADM <- lapply(region.countries ,function(x)
    raster::getData('GADM', country=x, level=0))

  message("Merging the selected countries...","\n",appendLF=FALSE)
  flush.console()

  # Change polygons IDs to unique values
  region.set.IDs <- function(sp,i) spChFIDs(sp,
    paste(i,row.names(sp@data),sep="."))

  # Combine SpatialPolygonDataFrame objects
  region.SPDF <- do.call(rbind,mapply(region.set.IDs,
    region.GADM,seq_along(region.GADM)))

  # Create an empty template
  region.r.ext <- extent(c(-180, 180, -90 , 90))
  region.r.res <- 1/12 ## resolution from GIMMS 3g specification
  region.r.tpl <- raster(ext = region.r.ext,
    resolution = region.r.res)

  # Rasterize SPDF polygons to raster with 0 value
  region.r <- rasterize(region.SPDF, region.r.tpl, field=0)

  #
  # Load GIMMS NDVI 3g data
  #

  # Define a raster stack for the NDVI time-series
  region.ts <- stack()

  # GIMMS NDVI 3g grid parameters
  data.samples <- 2160 ## or cols
  data.lines <- 4320 ## or rows
  data.size <- data.samples*data.lines ## number of items
  data.type <- 2 ## 2-byte short
  data.byte.order <- 1 ## 0 for little endian / 1 for big endian

  # Check what type of endian the reader should use
  ieee <- if(.Platform$endian=="big") 1 else 0
  ## check if this machine use big endian
  data.endian <- if(ieee==data.byte.order | data.byte.order<0)
    .Platform$endian else "swap"

  # Generate dates for the period
  data.dates <- as.yearmon(seq(as.numeric(data.date.start),
    as.numeric(data.date.end), 1/24))
  data.dates <- c(data.dates, data.date.end)
  ## Append one more measurement (two per month)

  # Format according to GIMMS file naming
```

```

data.names <- format(data.dates, format="%y%b")
data.names <- tolower(data.names)
## months to lower case

# Create two periods (a/b) for each month
data.names[1:length(data.names) %% 2 == 1] <- paste(data.names[1:
  length(data.names) %% 2 == 1], "15a", sep = "")
data.names[1:length(data.names) %% 2 == 0] <- paste(data.names[1:
  length(data.names) %% 2 == 0], "15b", sep = "")

# Create a list of files
file.patern <- paste("geo", data.names, ".*-VI3g", sep = "")
data.files <- lapply(file.patern, function(x) {list.files(
  data.path, pattern = x, full.names = TRUE)[1]})

message("Loading NDVI3g data...", "\n", appendLF=FALSE)
flush.console()

pb <- txtProgressBar(min = 0, max = length(data.files), style = 3)

region.ts <- stack(sapply(seq(data.files), function(i) {
  setTxtProgressBar(pb, i)
  flush.console()
  readNDVI3g(data.file=data.files[[i]], data.size, data.type,
    data.endian, data.samples, data.lines,
    region.r, region.SPDF)
}))

close(pb)

# Name the layers
names(region.ts) <- data.names

# Set layers' dates
data.dates <- as.Date(as.yearmon(data.dates))
data.dates[1:length(data.dates) %% 2 == 0] <- data.dates[1:
  length(data.dates) %% 2 == 0] + 15
region.ts <- setZ(region.ts, data.dates)
}

```

## Appendix C. Read NDVI3g function.

Filename: readNDVI3g.R

---

```
readNDVI3g <- function(data.file,data.size, data.type, data.endian,
                      data.samples, data.lines, region.r, region.SPDF) {

  # Open and read the first found file
  f.in <- file(data.file,'rb')
  X <- readBin(f.in, integer(), n=data.size, size=data.type,
              endian=data.endian)
  close(f.in)

  # Put the grid values in matrix
  X <- matrix(X, nrow=data.samples, ncol=data.lines)

  # Define raster template
  r.ext <- extent(c(-180, 180, -90 , 90))
  r.template <- raster(nrows = data.samples, ncols = data.lines,
                      ext = r.ext)

  # Insert the matrix values in template
  r <- raster(X, template = r.template)

  # Set NA values and scale according to GIMMS 3g specification
  r[r[] == -10000] <- NA ## Water
  r[r[] == -5000] <- NA ## No data
  r <- r / 10000 ## Scale

  # Mask with ME region and crop to ME
  r <- mask(r,region.r)
  r <- crop(r,region.SPDF)

}
```





## Appendix D. META PolyTrend algorithm.

Filename: META-POLYTREND.R

---

```
library(zoo)
library(sp)
library(raster)

rasterOptions(progress="text")

# Config section

region.countries <- list ( "BHR", ## Bahrain
                           "CYP", ## Cyprus
                           "EGY", ## Egypt
                           "IRN", ## Iran
                           "IRQ", ## Iraq
                           "ISR", ## Israel
                           "JOR", ## Jordan
                           "KWT", ## Kuwait
                           "LBN", ## Lebanon
                           "OMN", ## Oman
                           "PSE", ## Palestine
                           "QAT", ## Qatar
                           "SAU", ## Saudi Arabia
                           "SYR", ## Syria
                           "TUR", ## Turkey
                           "ARE", ## United Arab Emirates
                           "YEM") ## Yemen

# Path to GIMMS NDVI3g dataset
data.path <- "/PATH/T0/NDVI3g/"

# Path to output the results
output.path <- "/PATH/T0/ResultsPT/"

data.date.start <- as.yearmon('1982-01')
data.date.end <- as.yearmon('2010-12')

# NDVI threshold for the period
ndvi3g.threshold <- 0.1

# Number of growing months
nMonths <- 4

PolyTrend.DIR <- "/PATH/T0/PolyTrend/"
MidEastAnlz.DIR <- "/PATH/T0/META-POLYTREND/"

# End Config

# Load PolyTrend
source(paste(PolyTrend.DIR,"PolyTrend.R",sep=""))

# Load MidEastAnlz files
source(paste(MidEastAnlz.DIR,"readNDVI3g.R",sep=""))
source(paste(MidEastAnlz.DIR,"loadRegion.R",sep=""))
source(paste(MidEastAnlz.DIR,"PTwrapper.R",sep=""))
source(paste(MidEastAnlz.DIR,"GenerateMaps.R",sep=""))

region.ts <- loadRegion(region.countries,data.date.start,data.date.end,
                        data.DIR)
```

```

NDVI3g.dates <- getZ(region.ts)
message("Processing dataset...", "\n", appendLF=FALSE)
flush.console()

rc.trends <- calc(region.ts, fun=function(x) {PTwrapper(x,s
              tackDates=data.dates,ndvi3g.threshold=ndvi3g.threshold,
              nMonths=nMonths)}, forceapply=TRUE)

names(rc.trends) <- c("TrendType", "Slope", "Direction", "Significance")

# Report
message("Writing the results...", "\n", appendLF=FALSE)
flush.console()

output.files <- paste(output.path, format(data.date.start, format="%Y%b"),
                    format(data.date.end, format="%Y%b"), ".",
                    names(rc.trends), ".", "PolyTrends", sep="")

writeRaster(rc.trends, filename=output.files, bylayer=TRUE,
            format=output.format, overwrite=TRUE)

output.statfile <- paste(output.path,
                        format(data.date.start, format="%Y%b"),
                        format(data.date.end, format="%Y%b"), ".",
                        "PolyTrends.txt", sep="")

out<-capture.output(freq(rc.trends))
cat(out, file=output.statfile, sep="\n")

GenerateMaps(output.path, output.name = "TrendType", region.SPDF,
            map.raster=rc.trends$TrendType,
            map.title=paste("Trend Type",
                            format(data.date.start, format="%Y%b"),
                            format(data.date.end, format="%Y%b", sep=" ")),
            colors=c( "#e60001", "#0170fd", "#d0ff72", "#fdedce", "#cdcdcd"),
            legend.classes = TRUE, legend.cols = 3,
            legend.labels=c("Cubic", "Quadratic", "Linear", "No Trend",
                             "Concealed"),
            revOrder = TRUE)

GenerateMaps(output.path, output.name = "TrendSignificance", region.SPDF,
            map.raster=rc.trends$Significance,
            map.title=paste("Trend Significance",
                            format(data.date.start, format="%Y%b"),
                            format(data.date.end, format="%Y%b", sep=" ")),
            colors=c("#08ae00", "#f5b338"),
            legend.classes = TRUE, legend.cols = 2,
            legend.labels=c("Significant", "Not Significant"),
            revOrder = TRUE)

GenerateMaps(output.path, output.name = "TrendSlope", region.SPDF,
            map.raster=rc.trends$Slope,
            map.title=paste("Trend Slope",
                            format(data.date.start, format="%Y%b"),
                            format(data.date.end, format="%Y%b", sep=" ")),
            colors=brewer.pal(10, "RdYlGn"))

GenerateMaps(output.path, output.name = "TrendDirection", region.SPDF,
            map.raster=rc.trends$Direction,
            map.title=paste("Trend Direction",
                            format(data.date.start, format="%Y%b"),
                            format(data.date.end, format="%Y%b", sep=" ")),
            colors=c("#a6d226", "#d07b1d"),

```

```

legend.classes = TRUE, legend.cols = 2,
legend.labels=c("Positive","Negative"),
revOrder = TRUE)

GenerateMaps(output.path, output.name = "TrendSamples",region.SPDF,
map.raster=rc.trends$TrendType,
map.title=paste("Trend Samples",
                format(data.date.start, format="%Y%b"),
                format(data.date.end, format="%Y%b", sep=" ")),
colors=c( "#e60001", "#0170fd", "#d0ff72", "#fdedce", "#cdcddc"),
legend.classes = TRUE, legend.cols = 3,
legend.labels=c("Cubic","Quadratic","Linear","No Trend",
                "Concealed"),

revOrder = TRUE,
prob.samples = TRUE,
nSamplesPerClass = 2,
pdfGrid = c(5,2),
region.ts = region.ts)

message("Done.", "\n", appendLF=FALSE)
flush.console()

```

### Filename: PTwrapper.R

```

PTwrapper <- function(x, stackDates, ndvi3g.threshold, nMonths) {

  if(any(is.na(x))) { ## filter NA

    return(c(NA, NA, NA, NA))
  } else {

    ndvi3g.ts <- zoo(as.vector(x), stackDates)

    if (any(as.vector(rollapply(ndvi3g.ts, 24, mean,
                               by = 24)) < ndvi3g.threshold)){
      return(c(NA, NA, NA, NA))
    } else {

      ndvi3g.ts <- aggregate(ndvi3g.ts, as.yearmon, mean)
      by.years <- split( as.vector(ndvi3g.ts), as.vector(
        format(time(ndvi3g.ts), "%y") ) )
      ndvi3g.ts <- vapply(by.years, function(x) mean(
        sort(x, decreasing = TRUE)[1:nMonths]),
        FUN.VALUE=double(1))

      ndvi3g.ts <- unname(ndvi3g.ts)
      PT <- PolyTrend(Y = ndvi3g.ts, alpha = 0.05)
      return(c(PT$TrendType, PT$Slope, PT$Direction,
              PT$Significance))
    }
  } ## end na check
} ## end

```

### Filename: GenerateMaps.R

```

GenerateMaps <- function(output.path, output.name, countries.SPDF,
                        countries.bg = "white",
                        countries.border = "gray70",
                        map.raster, map.title, scalebar.dist=1000,
                        northarrow.size=3,
                        northarrow.topoffset=1, colors,
                        legend.classes = FALSE, legend.cols = 2,

```

```

        legend.lables, prob.samples=FALSE,
        nSamplesPerClass = 2, revOrder=FALSE,
        pdfGrid = c(5,2), region.ts) {

pdf(paste(output.path, paste(output.name, "pdf", sep="."),sep="/"))
par.old <- par(no.readonly = TRUE)
on.exit(par(par.old))
par(mar=c(4, 4, 4, 5))

plot(countries.SPDF, lwd=0.2, border=countries.bg,col=countries.bg,
     xlim=c(xmin(map.raster) + 1, xmax(map.raster) - 1),
     ylim=c(ymin(map.raster) + 1, ymax(map.raster) - 1),axes = F)

degAxis(1, col = 'white', col.ticks = 'black', cex.axis = 0.95,
        font = 3, family = 'serif')
degAxis(2, col = 'white', col.ticks = 'black', cex.axis = 0.95,
        font = 3, family = 'serif')
box()
if(legend.classes) {
  if(revOrder) {
    plot(map.raster, legend = FALSE, col=rev(colors),
         axes=F, add=TRUE)
  }else{
    plot(map.raster, legend = FALSE, col=colors,
         axes=F, add=TRUE)
  }
} else {
  redvals <- seq(minValue(map.raster),0,length.out = 6)[-6]
  greenvals <- seq(0,maxValue(map.raster),length.out = 6)[-1]
  breakingpoints <-c(redvals,0,greenvals)

  plot(map.raster, legend = FALSE, col=colors,
       breaks = breakingpoints,
       axes=F, add=TRUE)

  plot(map.raster, legend.only=TRUE, col=colors,
       breaks = breakingpoints,
       legend.width=1, legend.shrink=0.75,
       axis.args=list(at=breakingpoints,
                     labels=round(breakingpoints,3),
                     cex.axis=0.6),
       legend.args=list(text='Slope (NDVI/Year)', side=4,
                       font=2, line=2.5, cex=0.8))
}

invisible()

grid(lwd = 1.8)
text(xmin(map.raster) + 0.5, ymin(map.raster) - 3,
     "Projection: Geographic\nCoordinate System: WGS 1984\n
     Data Source: GIMMS3g/GADM",
     adj=c(0,0), cex=0.7, col="gray18")
scalebar(scalebar.dist, xy = c(xmin(map.raster) +
                              0.5,ymin(map.raster)),
         lonlat = TRUE, label = paste(scalebar.dist, "km"), lwd=2)

if(legend.classes) {
  colors <- c(colors,"#ffffff")
  legend.lables <- c(legend.lables,"Masked")
}

```

```

        legend("bottomright", legend = legend.labels,
              fill = colors, ncol=legend.cols, cex = 0.8)
    }

    north.coord <- c(xmin(map.raster) + northarrow.topoffset,
                   ymax(map.raster) - northarrow.topoffset)
    SpatialPolygonsRescale(layout.north.arrow(1), offset = north.coord,
                          scale = northarrow.size, plot.grid=F)
    plot(countries.SPDF, lwd=0.8, border="gray70", add=TRUE, axes = F)

    mtext(side=3, line=1, map.title, cex=1.2)
    mtext(side=1, "Longitude", line=2.5, cex=0.8)
    mtext(side=2, "Latitude", line=2.5, cex=0.8)

    if(prob.samples) {

        prob <- probRaster(nSamplesPerClass, map.raster)
        points(prob$xy.samples[,1], prob$xy.samples[,2])
        text(prob$xy.samples[,1], prob$xy.samples[,2],
             paste(letters[1:nrow(prob$xy.samples)]), cex = 0.85,
             pos=4)

        dev.off()

        pdf(paste(output.path, paste(output.name, "Fig", "pdf",
                                     sep=".")),
            title="PolyTrend Samples", width=8.27, height=11.7)
        par(mfrow = pdfGrid, oma=c(2,5,5,0), mar=c(3,3,2,2))

        figDates <- unique(as.vector(format(getZ(region.ts), "%y")))
        ndvi3g.vals <- sapply(prob$cell.samples,
                             function(x) aggregateProb(
                                 region.ts[x], getZ(region.ts)))
        sapply(1:ncol(ndvi3g.vals), function(x) plot(
            PolyTrend(Y = ndvi3g.vals[,x], alpha = 0.05),
            fig.text=paste(letters[x], ""), sep=""),
            fig.dates=figDates))

        mtext(side=3, line=1, map.title, cex=1.4,
              adj = 0.06, outer = TRUE)
        dev.off()

        output.statfile <- paste(output.path,
                                 format(data.date.start, format="%Y%b"),
                                 format(data.date.end, format="%Y%b"), ".",
                                 "SamplesNDVI.txt", sep="")
        out<-capture.output(list("ndvi3g.vals"=ndvi3g.vals,
                                "Years" =figDates))
        cat(out, file=output.statfile, sep="\n")

    } else {

        dev.off()
    }
}

probRaster <- function(nSamples, rst) {
  xy.samples <- vector(mode="numeric", length=0)
  cell.samples <- vector(mode="numeric", length=0)
  unq <- unique(rst)
  for(i in 1:length(unq))
  {
    unq.values <-which(values(rst)==rev(unq[i]))
  }
}

```

```

    rnd.samples <- sample(unq.values, nSamples)
    coord.sample <- xyFromCell(rst, rnd.samples)

    xy.samples <- rbind(xy.samples, coord.sample)
    cell.samples <- cbind(cell.samples, rnd.samples)

  }
  return(list("xy.samples" = xy.samples,
            "cell.samples" = cell.samples))
}

aggregateProb <- function(x, stackDates) {

  ndvi3g.ts <- zoo(as.vector(x), stackDates)
  ndvi3g.ts <- aggregate(ndvi3g.ts, as.yearmon, mean)
  by.years <- split(as.vector(ndvi3g.ts),
                    as.vector(format(time(ndvi3g.ts), "%y"))) )
  ndvi3g.ts <- as.vector(sapply(by.years,
                               function(x) mean(sort(x, decreasing = TRUE)[1:4])))
  return(ndvi3g.ts)
}

```

## Appendix E. META DBEST algorithm.

Filename: META-DBEST.R

```
library(zoo)
library(sp)
library(raster)

rasterOptions(progress="text")

# Config section

region.countries <- list ( "BHR", ## Bahrain
                           "CYP", ## Cyprus
                           "EGY", ## Egypt
                           "IRN", ## Iran
                           "IRQ", ## Iraq
                           "ISR", ## Israel
                           "JOR", ## Jordan
                           "KWT", ## Kuwait
                           "LBN", ## Lebanon
                           "OMN", ## Oman
                           "PSE", ## Palestine
                           "QAT", ## Qatar
                           "SAU", ## Saudi Arabia
                           "SYR", ## Syria
                           "TUR", ## Turkey
                           "ARE", ## United Arab Emirates
                           "YEM") ## Yemen

Output.DIR <- "/PATH/TO/Outputs/"
DBEST.DIR <- "/PATH/TO/DBEST/"
MidEastAnlz.DIR <- "/PATH/TO/METADBEST/"
data.DIR <- "/PATH/TO/ndvi3g_81_11/"

data.date.start <- as.yearmon('1982-01')
data.date.end <- as.yearmon('2010-12')

# End Config

# Load DBEST
source(paste(DBEST.DIR, "inv.R", sep=""))
source(paste(DBEST.DIR, "pinv.R", sep=""))
source(paste(DBEST.DIR, "m1divide.R", sep=""))
source(paste(DBEST.DIR, "DETREND_SJ.R", sep=""))
source(paste(DBEST.DIR, "Pvalue_ith_coeff.R", sep=""))
source(paste(DBEST.DIR, "Point_to_line.R", sep=""))
source(paste(DBEST.DIR, "BIC_BP.R", sep=""))
source(paste(DBEST.DIR, "DBEST_timeseries_BP.R", sep=""))
source(paste(DBEST.DIR, "DBEST_timeseries_Threshold.R", sep=""))
source(paste(DBEST.DIR, "Compute_epsilon.R", sep=""))
source(paste(DBEST.DIR, "f_level_shift.R", sep=""))
source(paste(DBEST.DIR, "Trend_local_change.R", sep=""))
source(paste(DBEST.DIR, "DBEST.R", sep=""))

# Load MidEastAnlz files
source(paste(MidEastAnlz.DIR, "readNDVI3g.R", sep=""))
source(paste(MidEastAnlz.DIR, "loadRegion.R", sep=""))
source(paste(MidEastAnlz.DIR, "wrapperDBEST.R", sep=""))
source(paste(MidEastAnlz.DIR, "plotMaps.R", sep=""))
```

```

region.ts <- loadRegion(region.countries,data.date.start,data.date.end,
                        data.DIR)

NDVI3g.dates <- getZ(region.ts)
message("Processing dataset...", "\n", appendLF=FALSE)
flush.console()
trends.characteristics <- calc(region.ts, fun=function(x) {
  wrapperDBEST(x,stackDates=NDVI3g.dates)}, forceapply=TRUE)
names(trends.characteristics) <- c("TrendSegments", "StartYear", "Duration",
                                   "EndYear", "Change", "ChangeType",
                                   "Significance")

# Output GTiff
output.files <- paste(Output.DIR, format(data.date.start, format="%Y%b"),
                     format(data.date.end, format="%Y%b"), ".",
                     names(trends.characteristics), ".", "DBEST", sep="")
writeRaster(trends.characteristics, filename=output.files, bylayer=TRUE,
            format="GTiff", overwrite=TRUE)

# Output Statistic
output.statfile <- paste(Output.DIR, format(data.date.start,
                                           format="%Y%b"),
                        format(data.date.end, format="%Y%b"),
                        ".", "DBEST.txt", sep="")
out<-capture.output(freq(trends.characteristics))
cat(out, file=output.statfile, sep="\n")

# Output Maps
region.GADM <- lapply(region.countries ,function(x) raster::getData('GADM',
                                                                    country=x, level=0))
region.set.IDs <- function(sp,i) spChFIDs(sp,paste(i,row.names(sp@data),
                                                  sep="."))
region.SPDF <- do.call(rbind,mapply(region.set.IDs,region.GADM,
                                   seq_along(region.GADM)))

# Classify insignificant pixels
rcl.insignificant <- trends.characteristics$Significance - 1
rcl.insignificant <- rcl.insignificant * -9999

# TrendSegments
rcl.TrendSegments <- trends.characteristics$TrendSegments +
  rcl.insignificant
classes.TrendSegments <- c(0, 20, 1, 20, 40, 2, 40, 60, 3, 60, 100, 4,
                          100, Inf, 5)
classes.TrendSegments <- matrix(classes.TrendSegments, ncol=3, byrow=TRUE)
rcl.TrendSegments <- reclassify(rcl.TrendSegments, classes.TrendSegments)

plotMaps(Output.DIR, output.name = "TrendSegments",region.SPDF,
         map.raster=rcl.TrendSegments,
         map.title=paste("Number of Trend Segments",
                        format(data.date.start, format="%Y%b"),
                        format(data.date.end, format="%Y%b", sep=" ")),
         colors=c("#e60000", "#feff72", "#ffbfbf", "#73b2ff", "#b2b2b2"),
         legend.cols = 3,
         legend.lables=c("bellow 20", "20 - 40", "40 - 60", "above 60",
                        "Insignificant")
)

rcl.StartYear <- trends.characteristics$StartYear + rcl.insignificant
classes.StartYear <- c(1981, 1988, 1, 1988, 1995, 2, 1995, 2002, 3, 2002,
                     2010, 4, 2010, Inf, 5)
classes.StartYear <- matrix(classes.StartYear, ncol=3, byrow=TRUE)

```



```

rcl.StartYear <- reclassify(rcl.StartYear, classes.StartYear)

plotMaps(Output.DIR, output.name = "StartYear",region.SPDF,
  map.raster=rcl.StartYear,
  map.title=paste("Start Year of Change",
    format(data.date.start, format="%Y%b"),
    format(data.date.end, format="%Y%b", sep=" ")),
  colors=c( "#fe0000", "#ffc701", "#b7ff8f", "#33c2fe", "#b2b2b2"),
  legend.cols = 3,
  legend.lables=c("1982 - 1988","1989 - 1995","1996 - 2002",
    "2003 - 2010", "Insignificant")
)

# Duration
rcl.Duration <- trends.characteristics$Duration + rcl.insignificant
classes.Duration <- c(0, 1, 1, 1, 12, 2, 12, 24, 3, 24, 36,4, 36, 100, 5,
  100, Inf, 6)
classes.Duration <- matrix(classes.Duration, ncol=3, byrow=TRUE)
rcl.Duration <- reclassify(trends.characteristics$Duration,
  classes.Duration)

plotMaps(Output.DIR, output.name = "Duration",region.SPDF,
  map.raster=rcl.Duration,
  map.title=paste("Duration of Change (months)",
    format(data.date.start, format="%Y%b"),
    format(data.date.end, format="%Y%b", sep=" ")),
  colors=c( "#e60000", "#feff72", "#ffbfbf", "#73b2ff", "#b2b2b2"),
  legend.cols = 3,
  legend.lables=c("1","2 - 12","12 - 24","above 24","Insignificant")
)

# EndYear
rcl.EndYear <- trends.characteristics$EndYear + rcl.insignificant
classes.EndYear <- c(1981, 1988, 1, 1988, 1995, 2, 1995, 2002, 3, 2002,
  2010, 4, 2010, Inf, 5)
classes.EndYear <- matrix(classes.EndYear, ncol=3, byrow=TRUE)
rcl.EndYear <- reclassify(rcl.EndYear, classes.EndYear)

plotMaps(Output.DIR, output.name = "EndYear",region.SPDF,
  map.raster=rcl.EndYear,
  map.title=paste("End Year of Change",
    format(data.date.start, format="%Y%b"),
    format(data.date.end, format="%Y%b", sep=" ")),
  colors=c( "#fe0000", "#ffc701", "#b7ff8f", "#33c2fe", "#b2b2b2"),
  legend.cols = 3,
  legend.lables=c("1982 - 1988","1989 - 1995","1996 - 2002",
    "2003 - 2010", "Insignificant")
)

# Change
rcl.Change <- trends.characteristics$Change + rcl.insignificant
classes.Change <- c(-0.3, -0.2, 1, -0.2, -0.1, 2, -0.1, 0, 3, 0, 0.1, 4,
  0.1, 0.2, 5, 0.2, 0.3, 6, 0.3, 0.4, 7, 0.4, 0.5, 8,
  0.5, 1, 9, 1, Inf, 10)
classes.Change <- matrix(classes.Change, ncol=3, byrow=TRUE)
rcl.Change <- reclassify(rcl.Change, classes.Change)

plotMaps(Output.DIR, output.name = "Change",region.SPDF,
  map.raster=rcl.Change,
  map.title=paste("Magnititude of Change",
    format(data.date.start, format="%Y%b"),
    format(data.date.end, format="%Y%b", sep=" ")),

```

```

colors=c( "#ff2100", "#ff7f7e","#ffedbf","#e9febf","#d6e700",
          "#8bb402","#d0ff72","#4a8a00","#00734c", "#b2b2b2"),
legend.cols = 2,
legend.labels=c("-0.3 - -0.2","-0.2 - -0.1", "-0.1 - 0",
               "0 - 0.1", "0.1 - 0.2", "0.2 - 0.3", "0.3 - 0.4",
               "0.4 - 0.5", "0.5 - 1", "Insignificant")
)

# ChangeType
rcl.ChangeType <- trends.characteristics$ChangeType + rcl.insignificant
rcl.ChangeType <- reclassify(rcl.ChangeType, c(0,1,1,1,2,2,2,Inf,3))

plotMaps(Output.DIR, output.name = "ChangeType",region.SPDF,
         map.raster=rcl.ChangeType,
         map.title=paste("Change Type",
                        format(data.date.start, format="%Y%b"),
                        format(data.date.end, format="%Y%b", sep=" ")),
         colors=c( "#bfe9ff", "#e60000", "#b2b2b2"),
         legend.cols = 2,
         legend.labels=c("Non-Abrupt", "Abrupt", "Insignificant")
)

```

## Filename: wrapperDBEST.R

```

wrapperDBEST <- function(x, stackDates) {

  if(any(is.na(x))) { ## filter NA
    return(c(NA, NA, NA, NA, NA, NA, NA))
  } else {

    DBEST.results <- tryCatch({

      ndvi.pixel <- zoo(x,stackDates, frequency=24)
      ndvi.pixel.monthly <- aggregate(ndvi.pixel,
                                     as.yearmon(index(ndvi.pixel)), "mean")

      OutputDBEST=DBEST(ndvi.pixel.monthly,
                       seasonality=12,
                       data.type="cyclical",
                       algorithm="change detection",
                       breakpoints.no=1,
                       first.level.shift=0.1,
                       second.level.shift=0.2,
                       duration=24,
                       distance.threshold="default",
                       alpha=0.05)

      TrendSegments <- OutputDBEST$SegmentNo
      StartYear <- as.numeric(format(
        as.Date(time(ndvi.pixel.monthly)[
          OutputDBEST$Start]), "%Y"))
      Duration <- OutputDBEST$Duration
      EndYear <- as.numeric(format(
        as.Date(time(ndvi.pixel.monthly)[
          OutputDBEST$End]), "%Y"))
      Change <- OutputDBEST$Change
      ChangeType <- OutputDBEST$ChangeType
      Significance <- OutputDBEST$Significance
      return(c(TrendSegments, StartYear, Duration, EndYear,
              Change, ChangeType, Significance))

    }, error=function(e){

```

```

        return(c(NA, NA, NA, NA, NA, NA, NA))
    }) ## end try catch
    return(DBEST.results)
}
}

```

## Filename: plotMaps.R

```

plotMaps <- function(output.path, output.name, countries.SPDF,
  countries.bg = "white", countries.border = "gray70",
  map.raster, map.title, scalebar.dist=1000,
  northarrow.size=3,northarrow.topoffset=1, colors,
  legend.cols = 2, legend.labels, pdfGrid = c(5,2),
  region.ts) {

  pdf(paste(output.path, paste(output.name, "pdf", sep="."),
    sep="/"))

  par.old <- par(no.readonly = TRUE)
  on.exit(par(par.old))
  par(mar=c(4, 4, 4, 5))
  plot(countries.SPDF, lwd=0.2, border=countries.bg,
    col=countries.bg,
    xlim=c(xmin(map.raster) + 1, xmax(map.raster) - 1),
    ylim=c(ymin(map.raster) + 1, ymax(map.raster) - 1),axes = F)

  degAxis(1, col = 'white', col.ticks = 'black', cex.axis = 0.95,
    font = 3, family = 'serif')
  degAxis(2, col = 'white', col.ticks = 'black', cex.axis = 0.95,
    font = 3, family = 'serif')

  box()
  plot(map.raster, legend = FALSE, col=colors, axes=F, add=TRUE)
  invisible()

  grid(lwd = 1.8)
  text(xmin(map.raster) + 0.5, ymin(map.raster) - 3,
    "Projection: Geographic\nCoordinate System:
    WGS 1984\nData Source: GIMMS3g/GADM",
    adj=c(0,0), cex=0.7, col="gray18")

  scalebar(scalebar.dist, xy = c(xmin(map.raster) + 0.5,
    ymin(map.raster)), lonlat = TRUE,
    label = paste(scalebar.dist, "km"), lwd=2)
  colors <- c(colors,"#ffffff")
  legend.labels <- c(legend.labels,"Masked")
  legend("bottomright", legend = legend.labels,
    fill = colors,ncol=legend.cols,cex = 0.8)
  north.coord <- c(xmin(map.raster) + northarrow.topoffset,
    ymax(map.raster) - northarrow.topoffset)
  SpatialPolygonsRescale(layout.north.arrow(1), offset = north.coord,
    scale = northarrow.size, plot.grid=F)
  plot(countries.SPDF, lwd=0.8, border="gray70", add=TRUE, axes = F)
  mtext(side=3, line=1, map.title, cex=1.2)
  mtext(side=1, "Longitude", line=2.5, cex=0.8)
  mtext(side=2, "Latitude", line=2.5, cex=0.8)
  dev.off() }

```



## Appendix F. MELCA algorithm.

Filename: MELCA.R

---

```
library(sp)
library(raster)

# Directory of the script
LandCoverMapAnlz.DIR <- "/PATH/TO/MELCA/"

# Load
source(paste(LandCoverMapAnlz.DIR,"lcPlotMaps.R",sep=""))
source(paste(LandCoverMapAnlz.DIR,"landcoverAnlz.R",sep=""))
source(paste(LandCoverMapAnlz.DIR,"areaStat.R",sep=""))

#UMD Land Cover Classification
Output.DIR <- "/PATH/TO/Outputs/"
lc.data <- raster("/PATH/TO/AVHRR_8km_LANDCOVER_1981_1994.GLOBAL.tif")
Results.PolyTrend.DIR <-"/PATH/TO/Results/"
Results.DBEST.DIR <-"/PATH/TO/Outputs/"

# files to map PolyTrend
Results.PolyTrend.Direction <- paste(Results.PolyTrend.DIR,
                                     "1982Jan2010Dec.Direction.PolyTrends.tif",sep="")
Results.PolyTrend.Significance <- paste(Results.PolyTrend.DIR,
                                       "1982Jan2010Dec.Significance.PolyTrends.tif",sep="")
Results.PolyTrend.TrendType <- paste(Results.PolyTrend.DIR,
                                     "1982Jan2010Dec.TrendType.PolyTrends.tif",sep="")

# files to map DBEST
Results.DBEST.Significance <- paste(Results.DBEST.DIR,
                                   "1982Jan2010Dec.Significance.tif",sep="")
Results.DBEST.ChangeType <- paste(Results.DBEST.DIR,
                                  "1982Jan2010Dec.ChangeType.tif",sep="")
Results.DBEST.StartYear <- paste(Results.DBEST.DIR,
                                 "1982Jan2010Dec.StartYear.tif",sep="")
Results.DBEST.Duration <- paste(Results.DBEST.DIR,
                                "1982Jan2010Dec.Duration.tif",sep="")
Results.DBEST.EndYear <- paste(Results.DBEST.DIR,
                               "1982Jan2010Dec.EndYear.tif",sep="")
Results.DBEST.Change <- paste(Results.DBEST.DIR,
                              "1982Jan2010Dec.Change.tif",sep="")

region.countries <- list ( "BHR", ## Bahrain
                           "CYP", ## Cyprus
                           "EGY", ## Egypt
                           "IRN", ## Iran
                           "IRQ", ## Iraq
                           "ISR", ## Israel
                           "JOR", ## Jordan
                           "KWT", ## Kuwait
                           "LBN", ## Lebanon
                           "OMN", ## Oman
                           "PSE", ## Palestine
                           "QAT", ## Qatar
                           "SAU", ## Saudi Arabia
                           "SYR", ## Syria
                           "TUR", ## Turkey
                           "ARE", ## United Arab Emirates
                           "YEM") ## Yemen
```

```

# land-cover according to http://glcf.umd.edu/data/landcover/
lc.labels <- c(
  "Water",
  "Evergreen Needleleaf Forest",
  "Deciduous Broadleaf Forest",
  "Mixed Forest",
  "Woodland",
  "Wooded Grassland",
  "Closed Shrubland",
  "Open Shrubland",
  "Grassland",
  "Cropland",
  "Bare Ground",
  "Permanent Snow and Ice"
)

# land-cover according to http://glcf.umd.edu/data/landcover/
lc.colors <- c(
  rgb(068,079,137, max = 255),
  rgb(001,100,000, max = 255),
  rgb(002,220,000, max = 255),
  rgb(000,255,000, max = 255),
  rgb(146,174,047, max = 255),
  rgb(220,206,000, max = 255),
  rgb(255,173,000, max = 255),
  rgb(255,251,195, max = 255),
  rgb(140,072,009, max = 255),
  rgb(247,165,255, max = 255),
  rgb(255,199,174, max = 255),
  rgb(000,255,255, max = 255)
)

# Load region countries data from GADM.org
region.GADM <- lapply(region.countries,
  function(x) raster::getData('GADM', country=x, level=0))

# Change polygons IDs to unique values
region.set.IDs <- function(sp,i) spChFIDs(sp,paste(i,row.names(sp@data),
  sep="."))

# Combine SpatialPolygonDataFrame objects
region.SPDF <- do.call(rbind,mapply(region.set.IDs,region.GADM,
  seq_along(region.GADM)))

# Mask with ME region and crop to ME
lc.local <- mask(lc.data,region.SPDF)
lc.local <- crop(lc.local,region.SPDF)

#reclass
m.rcl <- c(-1,0,1,
  0,1,2,
  1,2,NA,
  2,3,NA,
  3,4,3,
  4,5,4,
  5,6,5,
  6,7,6,
  7,8,7,
  8,9,8,
  9,10,9,
  10,11,10,
  11,12,11,

```

```

12,13,12)

m.rcl <- matrix(m.rcl, ncol=3, byrow=TRUE)
lc.rcls <- reclassify(lc.local, m.rcl)

# land cover map
lcPlotMaps(Output.DIR, output.name = "LandCoverMap", region.SPDF, map.raster=lc.rcls,
  map.title=paste("Land Cover in the Middle East", "1981", "-", "1994"),
  colors=lc.colors,
  legend.cols = 2,
  legend.labels=lc.labels,
  northarrow.topoffset=4
)

# land cover statistic
areaStat(lc.rcls,lc.labels,Output.DIR, "LandCover8194")

# Generate statistics and maps from land-cover & results

# Trend Direction
Results.PolyTrend.Direction <- raster(Results.PolyTrend.Direction)

cl.values <- c(-1,1)
cl.names <- c("Negative Trends (1982-2010)","Positive Trends (1982-2010)")
cl.fileNames <- c("LC_NegativeTrends","LC_PositiveTrends")

areaStat(Results.PolyTrend.Direction,cl.names,Output.DIR,
  "TrendDirections8210")
for(i in 1:length(cl.values)) {
  landcoverAnlz(r.result=Results.PolyTrend.Direction,
    lc.rcls = lc.rcls,
    lc.labels=lc.labels,
    lc.colors=lc.colors,
    region.SPDF=region.SPDF,
    class.filename=cl.fileNames[i],
    class.name=cl.names[i],
    class.value=cl.values[i],
    Output.DIR=Output.DIR)
}

# Trend Significance
Results.PolyTrend.Significance <- raster(Results.PolyTrend.Significance)

cl.values <- c(-1,1)
cl.names <- c("Insignificant Trends (1982-2010)",
  "Significant Trends (1982-2010)")
cl.fileNames <- c("LC_InsignificantTrends","LC_SignificantTrends")

areaStat(Results.PolyTrend.Significance,cl.names,Output.DIR,
  "TrendSignificance8210")
for(i in 1:length(cl.values)) {
  landcoverAnlz(r.result=Results.PolyTrend.Significance,
    lc.rcls = lc.rcls,
    lc.labels=lc.labels,
    lc.colors=lc.colors,
    region.SPDF=region.SPDF,
    class.filename=cl.fileNames[i],
    class.name=cl.names[i],
    class.value=cl.values[i],

```

```

Output.DIR=Output.DIR)
}
# Trend Type
Results.PolyTrend.TrendType <- raster(Results.PolyTrend.TrendType)

cl.values <- c(-1,0,1,2,3)
cl.names <- c("Concealed Trends (1982-2010)",
             "No Trends (1982-2010)",
             "Linear Trends (1982-2010)",
             "Quadratic Trends (1982-2010)",
             "Cubic Trends (1982-2010)")
cl.filesnames <- c("LC_ConcealedTrends",
                  "LC_NoTrends",
                  "LC_LinearTrends",
                  "LC_QuadraticTrends",
                  "LC_CubicTrends")

areaStat(Results.PolyTrend.TrendType,cl.names,Output.DIR,"TrendType8210")
for(i in 1:length(cl.values)) {

  landcoverAnlz(r.result=Results.PolyTrend.TrendType,
               lc.rcls = lc.rcls,
               lc.labels=lc.labels,
               lc.colors=lc.colors,
               region.SPDF=region.SPDF,
               class.filename=cl.filesnames[i],
               class.name=cl.names[i],
               class.value=cl.values[i],
               Output.DIR=Output.DIR)

}
# DBEST
Results.DBEST.Significance <- raster(Results.DBEST.Significance)

# Classify insignificant pixels
rcl.insignificant <- Results.DBEST.Significance - 1
rcl.insignificant <- rcl.insignificant * -9999

# ChangeType
Results.DBEST.ChangeType <- raster(Results.DBEST.ChangeType)
Results.DBEST.ChangeType <- Results.DBEST.ChangeType + rcl.insignificant
classes.ChangeType <- c(0,1,1,1,2,2,2,Inf,3)
classes.ChangeType <- matrix(classes.ChangeType, ncol=3, byrow=TRUE)
Results.DBEST.ChangeType <- reclassify(Results.DBEST.ChangeType,
                                     classes.ChangeType)

cl.values <- c(0,1,3)
cl.names <- c("Non-Abupt Changes (1982-2010)",
             "Abupt Changes (1982-2010)",
             "Insignificant Changes (1982-2010)")
cl.filesnames <- c("LC_NonAbupt","LC_Abupt", "LC_InsignificantDBEST")

areaStat(Results.DBEST.ChangeType,cl.names,Output.DIR,"ChangeType8210")
for(i in 1:length(cl.values)) {

  landcoverAnlz(r.result=Results.DBEST.ChangeType,
               lc.rcls = lc.rcls,
               lc.labels=lc.labels,
               lc.colors=lc.colors,
               region.SPDF=region.SPDF,
               class.filename=cl.filesnames[i],

```



```

        class.name=cl.names[i],
        class.value=cl.values[i],
        Output.DIR=Output.DIR)
}
# Start Year
Results.DBEST.StartYear <- raster(Results.DBEST.StartYear)
Results.DBEST.StartYear <- Results.DBEST.StartYear + rcl.insignificant
classes.StartYear <- c(1981, 1988, 1, 1988, 1995, 2, 1995, 2002, 3,
                      2002, 2010, 4, 2010, Inf, 5)
classes.StartYear <- matrix(classes.StartYear, ncol=3, byrow=TRUE)
Results.DBEST.StartYear <- reclassify(Results.DBEST.StartYear,
                                     classes.StartYear)

cl.values <- c(1,2,3,4)
cl.names <- c("Change Start 1982 - 1988","Change Start 1989 - 1995",
             "Change Start 1996 - 2002", "Change Start 2003 - 2010")
cl.fileNames <- c("LC_Start82to88","LC_Start89to95", "LC_Start96to02",
                 "LC_Start03to10")

areaStat(Results.DBEST.StartYear,cl.names,Output.DIR, "StartYear8210")
for(i in 1:length(cl.values)) {

    landcoverAnlz(r.result=Results.DBEST.StartYear,
                 lc.rcls = lc.rcls,
                 lc.lables=lc.lables,
                 lc.colors=lc.colors,
                 region.SPDF=region.SPDF,
                 class.filename=cl.fileNames[i],
                 class.name=cl.names[i],
                 class.value=cl.values[i],
                 Output.DIR=Output.DIR)

}
# Duration
Results.DBEST.Duration <- raster(Results.DBEST.Duration)
Results.DBEST.Duration <- Results.DBEST.Duration + rcl.insignificant
classes.Duration <- c(0, 1, 1, 1, 12, 2, 12, 24, 3, 24, 36,4, 36, 100, 4,
                    100, Inf, 5)
classes.Duration <- matrix(classes.Duration, ncol=3, byrow=TRUE)
Results.DBEST.Duration <- reclassify(Results.DBEST.Duration,
                                     classes.Duration)

cl.values <- c(1,2,3,4)
cl.names <- c("Change Duration 1 Month","Change Duration 2 - 12 Months",
             "Change Duration 12 - 24 Months",
             "Change Duration Above 24 Months")
cl.fileNames <- c("LC_Duration1","LC_Duration2to12",
                 "LC_Duration12to24","LC_Above24")

areaStat(Results.DBEST.Duration,cl.names,Output.DIR, "Duration8210")
for(i in 1:length(cl.values)) {

    landcoverAnlz(r.result=Results.DBEST.Duration,
                 lc.rcls = lc.rcls,
                 lc.lables=lc.lables,
                 lc.colors=lc.colors,
                 region.SPDF=region.SPDF,
                 class.filename=cl.fileNames[i],
                 class.name=cl.names[i],
                 class.value=cl.values[i],
                 Output.DIR=Output.DIR)
}

```

```

}
# EndYear
Results.DBEST.EndYear <- raster(Results.DBEST.EndYear)
Results.DBEST.EndYear <- Results.DBEST.EndYear + rcl.insignificant
classes.EndYear <- c(1981, 1988, 1, 1988, 1995, 2, 1995, 2002, 3, 2002,
                    2010, 4, 2010, Inf, 5)
classes.EndYear <- matrix(classes.EndYear, ncol=3, byrow=TRUE)
Results.DBEST.EndYear <- reclassify(Results.DBEST.EndYear, classes.EndYear)

cl.values <- c(1,2,3,4)
cl.names <- c("Change End 1982 - 1988", "Change End 1989 - 1995",
             "Change End 1996 - 2002", "Change End 2003 - 2010")
cl.filesnames <- c("LC_End82to88", "LC_End89to95", "LC_End96to02",
                  "LC_End03to10")

areaStat(Results.DBEST.EndYear, cl.names, Output.DIR, "EndYear8210")
for(i in 1:length(cl.values)) {

  landcoverAnlz(r.result=Results.DBEST.EndYear,
               lc.rcls = lc.rcls,
               lc.lables=lc.lables,
               lc.colors=lc.colors,
               region.SPDF=region.SPDF,
               class.filename=cl.filesnames[i],
               class.name=cl.names[i],
               class.value=cl.values[i],
               Output.DIR=Output.DIR)
}
# Change
Results.DBEST.Change <- raster(Results.DBEST.Change)
Results.DBEST.Change <- Results.DBEST.Change + rcl.insignificant
classes.Change <- c(-0.3, -0.2, 1, -0.2, -0.1, 2, -0.1, 0, 3, 0, 0.1, 4,
                  0.1, 0.2, 5, 0.2, 0.3, 6, 0.3, 0.4, 7, 0.4, 0.5, 8,
                  0.5, 1, 9, 1, Inf, 10)
classes.Change <- matrix(classes.Change, ncol=3, byrow=TRUE)
Results.DBEST.Change <- reclassify(Results.DBEST.Change, classes.Change)

cl.values <- c(1,2,3,4,5,6,7,8,9)
cl.names <- c("Change Value -0.3 - -0.2",
             "Change Value -0.2 - -0.1", "Change Value -0.1 - 0",
             "Change Value 0 - 0.1", "Change Value 0.1 - 0.2",
             "Change Value 0.2 - 0.3", "Change Value 0.3 - 0.4",
             "Change Value 0.4 - 0.5", "Change Value 0.5 - 1")

cl.filesnames <- c("LC_Change-0.3to-0.2",
                  "LC_Change-0.2to-0.1", "LC_Change-0.1to0",
                  "LC_Change0to0.1", "LC_Change0.1to0.2",
                  "LC_Change0.2to0.3", "LC_Change0.3to0.4",
                  "LC_Change0.4to0.5", "LC_Change0.5to1")

areaStat(Results.DBEST.Change, cl.names, Output.DIR, "Change8210")
for(i in 1:length(cl.values)) {

  landcoverAnlz(r.result=Results.DBEST.Change,
               lc.rcls = lc.rcls,
               lc.lables=lc.lables,
               lc.colors=lc.colors,
               region.SPDF=region.SPDF,
               class.filename=cl.filesnames[i],
               class.name=cl.names[i],
               class.value=cl.values[i],
               Output.DIR=Output.DIR)}

```

## Filename: lcPlotMaps.R

```
landcoverPlotMaps <- function(output.path, output.name, countries.SPDF,
                              countries.bg = "white",
                              countries.border = "gray70",
                              map.raster, map.title,
                              scalebar.dist=1000,northarrow.size=3,
                              northarrow.topoffset=1, colors, legend.cols = 2,
                              legend.lables, pdfGrid = c(5,2), region.ts) {

  pdf(paste(output.path, paste(output.name, "pdf", sep="."),sep="/"))

  par.old <- par(no.readonly = TRUE)
  on.exit(par(par.old))
  par(mar=c(4, 4, 4, 5))

  plot(countries.SPDF, lwd=0.2, border=countries.bg,col=countries.bg,
        xlim=c(xmin(map.raster) + 1, xmax(map.raster) - 1),
        ylim=c(ymin(map.raster) - 7, ymax(map.raster) - 1),axes = F)

  degAxis(1, col = 'white', col.ticks = 'black', cex.axis = 0.95,
          font = 3, family = 'serif')
  degAxis(2, col = 'white', col.ticks = 'black', cex.axis = 0.95,
          font = 3, family = 'serif')

  box()
  plot(map.raster, legend = FALSE, col=colors, axes=F, add=TRUE)
  invisible()

  grid(lwd = 1.8)
  text(xmin(map.raster) + 0.5, ymin(map.raster) - 3,
       "Projection: Geographic\nCoordinate System: WGS 1984\n
       Data Source: UMD/GADM",
       adj=c(0,0), cex=0.7, col="gray18")
  scalebar(scalebar.dist, xy = c(xmin(map.raster) + 0.5,
                                ymin(map.raster)), lonlat = TRUE,
           label = paste(scalebar.dist, "km"), lwd=2)

  colors <- c(colors,"#ffffff")
  legend.lables <- c(legend.lables,"Masked")
  legend("bottomright", legend = legend.lables,
        fill = colors,ncol=legend.cols,cex = 0.7)
  north.coord <- c(xmin(map.raster), ymax(map.raster) -
                  northarrow.topoffset)
  SpatialPolygonsRescale(layout.north.arrow(1), offset = north.coord,
                        scale = northarrow.size, plot.grid=F)
  plot(countries.SPDF, lwd=0.8, border="gray70", add=TRUE, axes = F)
  mtext(side=3, line=1, map.title, cex=1.2)
  mtext(side=1, "Longitude", line=2.5, cex=0.8)
  mtext(side=2, "Latitude", line=2.5, cex=0.8)
  dev.off() }
```

## Filename: landcoverAnlz.R

```
landcoverAnlz <- function(r.result,lc.rcls,lc.lables,lc.colors,
                          region.SPDF,class.filename,class.name,
                          class.value,Output.DIR) {

  # crop and resample by using the nearest neighbor
  lc.rcls <- resample(lc.rcls , r.result , method = "ngb" )
  lc.rcls <- crop(lc.rcls,r.result)

  # set other classes to NA
  r.class <- r.result
```

```

r.class[r.class[] != class.value] <- NA

# set current class to 1
r.class[r.class[] == class.value] <- 1

# land-cover x 1
r.lc.class <- lc.rcls * r.class

# get land-cover labels related to this class only
lc.class.labels <- lc.labels[unique(r.lc.class)]

# get land-cover colors related to this class only
lc.class.colors <- lc.colors[unique(r.lc.class)]

# calculate area and zonal
r.lc.class.area <- area(r.lc.class)
zonalarea <- zonal(r.lc.class.area, r.lc.class, 'sum')

# save zonal
rownames(zonalarea) <- lc.class.labels
out<-capture.output(zonalarea)
output.statfile <- paste(Output.DIR, class.filename, ".txt", sep="")
cat(out, file=output.statfile, sep="\n")

# plot the map
landcoverPlotMaps(Output.DIR, output.name = class.filename,
                  region.SPDF, map.raster=r.lc.class,
                  map.title=paste("Middle East Land Cover", " & ",
                                class.name),
                  colors=lc.class.colors,
                  legend.cols = 2,
                  legend.labels=lc.class.labels,
                  northarrow.topoffset=4
                )
}

```

Filename: areaStat.R

```

areastat <- function(r.result, cl.names, Output.DIR, filename) {

# calculate area and zonal
r.result.area <- area(r.result)
zonalarea <- zonal(r.result.area, r.result, 'sum')

# check if Insignificant class
check <- dim(zonalarea)[1] - length(cl.names)
if( check == 1) {
  cl.names <- c(cl.names, "Insignificant")
}

# save zonal
rownames(zonalarea) <- cl.names
out<-capture.output(zonalarea)
output.statfile <- paste(Output.DIR, filename, ".txt", sep="")
cat(out, file=output.statfile, sep="\n")

}

```

## Appendix G. PolyTrend algorithm.

Filename: PolyTrend.R

---

```
PolyTrend <- function(Y, alpha) {  
  
  X <- c(1:length(Y))  
  p3 <- lm(Y ~ poly(X, 3, raw=TRUE))  
  Pcubic <- unname(coef(summary(p3))[4,4])  
  
  Roots3 <- polyroot(c( unname(coef(p3)[2]), 2 * unname(coef(p3)[3]),  
                      3 * unname(coef(p3)[4])))  
  Roots3 <- sort(Roots3)  
  
  if (all.equal(Im(Roots3),c(0,0))==TRUE && Roots3[1]!=Roots3[2] &&  
      X[1]<=Re(Roots3[1]) && Re(Roots3[1])<=X[length(X)] &&  
      X[1]<= Re(Roots3[2]) && Re(Roots3[2])<=X[length(X)] &&  
      Pcubic<alpha) {  
  
    p1 <- lm(Y ~ X)  
    Plin <- unname(coef(summary(p1))[2,4])  
    Slope <- unname(coef(p1)[2])  
    Direction <- sign(Slope)  
  
    if(Plin<alpha) {  
  
      Trend_type <- 3  
      Significance <- 1  
      Poly_degree <- 3  
  
    } else {  
  
      Trend_type <- -1  
      Significance <- -1  
      Poly_degree <- 3  
  
    }  
  
  } else {  
  
    p2 <- lm(Y ~ poly(X, 2, raw=TRUE))  
    Pquadratic <- unname(coef(summary(p2))[3,4])  
    Roots2 <- polyroot( c(unname(coef(p2)[2]), 2 *  
                        unname(coef(p2)[3])) )  
    Roots2 <- Re(Roots2)  
  
    if (X[1]<=Roots2 && Roots2<=X[length(X)] &&  
        Pquadratic<alpha) {  
  
      p1 <- lm(Y ~ X)  
      Plin <- unname(coef(summary(p1))[2,4])  
      Slope <- unname(coef(p1)[2])  
      Direction <- sign(Slope)  
  
      if(Plin<alpha) {  
  
        Trend_type <- 2  
        Significance <- 1  
        Poly_degree <- 2  
  
      } else {  
  

```

```

        Trend_type <- -1
        Significance <- -1
        Poly_degree <- 2
    }

    } else {

        p1 <- lm(Y ~ poly(X, raw=TRUE))
        Plin <- unname(coef(summary(p1))[2,4])
        Slope <- unname(coef(p1)[2])
        Direction <- sign(Slope)

        if(Plin<alpha) {

            Trend_type <- 1
            Significance <- 1
            Poly_degree <- 1

        } else {

            Trend_type <- 0
            Significance <- -1
            Poly_degree <- 0

        }

    }

}

PT.values <- list(
  "Y" = as.vector(Y),
  "alpha" = alpha,
  "TrendType" = Trend_type,
  "Slope" = Slope,
  "Direction" = Direction,
  "Significance" = Significance,
  "PolynomialDegree" = Poly_degree
)
class(PT.values) <- "PolyTrend"

return(PT.values)
}

print.PolyTrend <- function(x, ...) {

  cat("\nPolyTrend input data:\n")
  cat(sprintf("\nY: %s\n", paste(x$Y, collapse=" ")))
  cat(sprintf("alpha: %.2f\n", x$alpha))

  cat("\nPolyTrend classification:\n")

  strTrendType <-c("concealed", "no trend", "linear", "quadratic",
                 "cubic")
  cat(sprintf("\nTrend type: %i (%s)\n", x$TrendType,
             strTrendType[x$TrendType+2] ))

  cat(sprintf("Slope: %.4f\n", x$Slope))

  strDirection <- "positive"
  if(x$Direction < 0) strDirection <- "negative"

```

```

cat(sprintf("Direction: %i (%s)\n", x$Direction, strDirection))

strSignificance <- "statistically significant"
if(x$Significance < 0) {
  strSignificance <- "statistically insignificant"
}
cat(sprintf("Significance: %i (%s)\n", x$Significance,
  strSignificance))

strPolynomialDegree <-c("no trend", "linear", "quadratic", "cubic")
cat(sprintf("Polynomial degree: %i (%s)\n", x$PolynomialDegree,
  strPolynomialDegree[x$PolynomialDegree+1]))

invisible(x)
}

plot.PolyTrend <-function(x, fig.dates = c(1:length(x$Y)), xlab = "Year",
  ylab = "NDVI", fig.text="", ...) {

plot(c(1:length(x$Y)), x$Y, type="l", col="gray32", lwd=2,
  xlab = "", ylab = "", xaxt = "n")
axis(1, at=1:length(x$Y), labels=fig.dates)

mtext(side=3, line=0.5, fig.text, cex=1.1, adj = 0)
mtext(side=1, xlab, line=2.5, cex=0.8)
mtext(side=2, ylab, line=2.5, cex=0.8)

strTrendType <-c("Concealed ternd", "No trend", "Linear Trend",
  "Quadratic Trend", "Cubic Trend")

strSignificance <- "sig."
if(x$Significance < 0) strSignificance <- "in-sig."
strSlope <- paste("Slope=",round(x$Slope, 4), sep = "")
strLabel <- paste(strTrendType[x$TrendType+2],
  strSlope,strSignificance,sep = ", ")

legend("topleft", strLabel, bty = "n", cex=0.7, text.font = 1,
  y.intersp=1)

if(x$PolynomialDegree == 1) {

  fit1 <- lm(x$Y ~ poly( c(1:length(x$Y)), raw=TRUE))
  abline(fit1, col = "gray8", lwd=4)
  points(c(1:length(x$Y)), x$Y, type="p", pch = 20,
    col="gray32")
}

if(x$PolynomialDegree == 2) {

  fit2 <- lm(x$Y ~ poly( c(1:length(x$Y)), 2, raw=TRUE))
  pol2 <- function(x) coef(fit2)[3]*x^2 + coef(fit2)[2]*x +
    coef(fit2)[1]

  curve(pol2, col = "gray8", lwd=4, add=T)
  points(c(1:length(x$Y)), x$Y, type="p", pch = 20,
    col="gray32")

  if(x$Significance == -1) {
    p1 <- lm(x$Y ~ c(1:length(x$Y)))
    abline(coef = coef(p1), col = "gray42", lty = 5,
      lwd = 3)
  }
}
}

```

```

    }
}
if(x$PolynomialDegree == 3) {
  fit3 <- lm(x$Y ~ poly( c(1:length(x$Y)), 3, raw=TRUE))
  pol3 <- function(x) coef(fit3)[4]*x^3 + coef(fit3)[3]*x^2 +
    coef(fit3)[2]*x + coef(fit3)[1]

  curve(pol3, col = "gray8", lwd=4, add=T)
  points(c(1:length(x$Y)), x$Y, type="p", pch = 20,
    col="gray32")

  if(x$Significance == -1) {
    p1 <- lm(x$Y ~ c(1:length(x$Y)))
    abline(coef = coef(p1), col = "gray42", lty = 5,
      lwd = 3)
  }
}
}
}

```



## Appendix H. DBEST algorithm.

Filename: DBEST.R

---

```
DBEST <- function(data, data.type, seasonality=-1, algorithm,
                 breakpoints.no=-1, generalization.percent=-1,
                 change.magnitude=-1, first.level.shift,
                 second.level.shift, duration,
                 distance.threshold, alpha, plot=-1){

  data.type <- tolower(data.type)
  algorithm <- tolower(algorithm)

  if(is.character(distance.threshold)) {
    distance.threshold <- tolower(distance.threshold)
  }

  if(is.character(breakpoints.no)) {
    breakpoints.no <- tolower(breakpoints.no)
    if(breakpoints.no=="all") {
      breakpoints.no <- length(as.vector(data))
    }
  }

  if(data.type=="non-cyclical") {
    data.type <- "noncyclical"
  }

  if(data.type=="non cyclical") {
    data.type <- "noncyclical"
  }

  if(algorithm=="generalization") {
    algorithm <- "generalisation"
  }

  if(algorithm=="change detection") {
    algorithm <- "changedetection"
  }

  if(algorithm=="change-detection") {
    algorithm <- "changedetection"
  }

  if((breakpoints.no)!=-1) {
    option <- "breakpoint"
  }

  if(generalization.percent!=-1) {
    option <- "percent"
  }

  if(change.magnitude!=-1) {
    option <- "magnitude"
  }

  if(seasonality!=-1) {
    if(is.character(seasonality)) {
      seasonality <- tolower(seasonality)
    }
  }
}
```

```

if(is.zoo(data) || is.ts(data)) {
  if(seasonality=="none" || seasonality=="null") {
    data <- as.zoo(ts(as.vector(data),
                     start=time(data)[1]))
  } else {
    data <- as.zoo(ts(as.vector(data),
                     start = time(data)[1],
                     frequency=seasonality))
  }
} else {
  if(seasonality=="none" || seasonality=="null") {
    data <- as.zoo(ts(as.vector(data),
                     start=1980, deltat=1/12))
  } else {
    data <- as.zoo(ts(as.vector(data),
                     start=1980,
                     frequency=seasonality))
  }
}

flvl <- f_level_shift(as.vector(data),first.level.shift,
                     second.level.shift,duration)
vec <- flvl$LS_point
no <- flvl$no
npoints <- length(as.vector(data))

switch(data.type,
       cyclical = {
         if(no > 0) {
           initial <- 1
           trend <- c()
           seasonal <- c()
           remainder <- c()
           temp <- sort(vec)

           for(i in 1:no) {
             stl.tmp <- stl(data[
               initial:temp[i]],
                           s.window = 7)
             trend1 <- stl.tmp$time.series[,
                                           "trend"]
             seasonal1 <- stl.tmp$time.series[,
                                              "seasonal"]
             remainder1 <- stl.tmp$time.series[,
                                                "remainder"]

             trend <- c(trend, trend1)
             seasonal <- c(seasonal, seasonal1)
             remainder <- c(remainder, remainder1)

             initial <- temp[i] + 1
           } ## end for loop

           stl.tmp <- stl(data[initial:npoints],

```

```

                                s.window = 7)
trend2 <- stl.tmp$time.series[,"trend"]
seasonal2 <- stl.tmp$time.series[,
                                "seasonal"]
remainder2 <- stl.tmp$time.series[,
                                "remainder"]

trend <- c(trend, trend2)
seasonal <- c(seasonal, seasonal2)
remainder <- c(remainder, remainder2)
trend <- as.vector(trend)

} else { ## end if no > 0

stl.tmp <- stl(data, s.window = 7)
trend <- stl.tmp$time.series[,"trend"]
seasonal <- stl.tmp$time.series[,"seasonal"]
remainder <- stl.tmp$time.series[,
                                "remainder"]

trend <- as.vector(trend)

} ## end else no > 0

},
noncyclical = {
trend <- as.vector(data)
}

) ## end switch 'type'

if(distance.threshold == "default") {
epsilon_max <- abs(max(as.vector(data)) -
                    min(as.vector(data)))
distance.threshold <- Compute_epsilon(trend,
                                     epsilon_max, vec)
}

TLC <- Trend_local_change(trend,distance.threshold,vec)
turning_no <- TLC$n_turning
f_local <- TLC$f_local
f_sign <- TLC$f_sign
turning <- TLC$Turning
RL_DIST <- TLC$RL_DIST
BIC_BP_no <- BIC_BP(trend,f_local,turning_no,f_sign,turning)
DTSBP <- DBEST_timeseries_BP(trend,BIC_BP_no,f_local,turning_no,
                             f_sign,turning)

Start <- DTSBP$Start
End <- DTSBP$End
Duration <- DTSBP$Duration
Magnitude <- DTSBP$Magnitude
BP_mark <- DTSBP$BP_mark
BP_order <- DTSBP$BP_order
trend_data_series <- DTSBP$trend_data_series
Param_no <- DTSBP$Param_no
RSS <- DTSBP$RSS
RMSE <- DTSBP$RMSE
MAD <- DTSBP$MAD
p <- DTSBP$p
R <- DTSBP$R
dfSJ <- DTSBP$dfSJ
normr <- DTSBP$normr

```

```

shared <- DTSBP$shared

Pvalue <- c()

for(i in 1:(Param_no-1)) {
  Pvalue <- c(Pvalue, Pvalue_ith_coeff(p,R,normr,dfSJ,i))
}

Sig_Slope_no <- length(Pvalue[Pvalue[-length(Pvalue)] < alpha])
Sig_Slope_position <- c()
Sig_BP_position <- c()

if(Sig_Slope_no>0) {
  Sig_Slope_position <- which(Pvalue[-length(Pvalue)]<alpha)
  Sig_turning_position <- c()
  if(shared[1] == 1) {
    for(i in 1:Sig_Slope_no) {
      Sig_turning_position[i] <- shared[
        Sig_Slope_position[i]]
    }
  } else {
    for(i in 1:Sig_Slope_no) {
      if(Sig_Slope_position[i]==1) {
        Sig_turning_position[i] <- 1
      } else {
        Sig_turning_position[i] <- shared[
          Sig_Slope_position[i]-1]
      }
    }
  }

  Sig_BP_position <- intersect(Start,Sig_turning_position)
}

switch(algorithm,
  generalisation={
    switch(option,
      percent={
        BP_no <- ceiling( BIC_BP_no -
          ( BIC_BP_no *
            ( as.integer(
              generalization.percent)/100
            )))
        DTSBP <- DBEST_timeseries_BP(trend,
          BP_no,f_local,
          turning_no,f_sign,
          turning)
        Start <- DTSBP$Start
        End <- DTSBP$End
        Duration <- DTSBP$Duration
        Magnitude <- DTSBP$Magnitude
        BP_mark <- DTSBP$BP_mark
        BP_order <- DTSBP$BP_order
        fit <- DTSBP$trend_data_series
        Param_no <- DTSBP$Param_no
        RSS <- DTSBP$RSS
        RMSE <- DTSBP$RMSE
        MAD <- DTSBP$MAD
        p <- DTSBP$p
        R <- DTSBP$R
      }
    )
  }
)

```

```

dfSJ <- DTSBP$dfSJ
normr <- DTSBP$normr
shared <- DTSBP$shared
},
breakpoint={
  if(breakpoints.no>BIC_BP_no) {
    breakpoints.no <- BIC_BP_no
  }
  BP_no <- breakpoints.no
  DTSBP <- DBEST_timeseries_BP(trend,
    BP_no,f_local,
    turning_no,f_sign,
    turning)

  Start <- DTSBP$Start
  End <- DTSBP$End
  Duration <- DTSBP$Duration
  Magnitude <- DTSBP$Magnitude
  BP_mark <- DTSBP$BP_mark
  BP_order <- DTSBP$BP_order
  fit <- DTSBP$trend_data_series
  Param_no <- DTSBP$Param_no
  RSS <- DTSBP$RSS
  RMSE <- DTSBP$RMSE
  MAD <- DTSBP$MAD
  p <- DTSBP$p
  R <- DTSBP$R
  dfSJ <- DTSBP$dfSJ
  normr <- DTSBP$normr
  shared <- DTSBP$shared

},
magnitude={

  DTSTH <- DBEST_timeseries_Threshold(
    trend,change.magnitude,
    f_local,turning_no,
    f_sign,turning,BIC_BP_no)

  BP_no <- DTSTH$n_BP
  Start <- DTSTH$Start
  End <- DTSTH$End
  Duration <- DTSTH$Duration
  Magnitude <- DTSTH$Magnitude
  BP_mark <- DTSTH$BP_mark
  BP_order <- DTSTH$BP_order
  fit <- DTSTH$trend_data_series
  Param_no <- DTSTH$Param_no
  RSS <- DTSTH$RSS
  RMSE <- DTSTH$RMSE
  MAD <- DTSTH$MAD
  shared <- DTSTH$shared

}

) ## end switch option

segment_no <- Param_no-1
BP_no <- BP_no

Output.SegmentNo <- segment_no
Output.RMSE <- RMSE

```

```

Output.MAD <- MAD
Output.Fit <- fit

DBEST.values <- list(
  "SegmentNo" = Output.SegmentNo,
  "RMSE" = Output.RMSE,
  "MAD" = Output.MAD,
  "Fit" = Output.Fit
)

}, ## end case generalisation algorithm
changedetection={

  DTSBP <- DBEST_timeseries_BP(trend,BIC_BP_no,
                                f_local,turning_no,
                                f_sign,turning)

  Start <- DTSBP$Start
  End <- DTSBP$End
  Duration <- DTSBP$Duration
  Magnitude <- DTSBP$Magnitude
  BP_mark <- DTSBP$BP_mark
  BP_order <- DTSBP$BP_order
  fit <- DTSBP$trend_data_series
  Param_no <- DTSBP$Param_no
  RSS <- DTSBP$RSS
  RMSE <- DTSBP$RMSE
  MAD <- DTSBP$MAD
  shared <- DTSBP$shared

  switch(option,
    breakpoint={
      if(breakpoints.no>BIC_BP_no) {
        breakpoints.no<-BIC_BP_no
      }
    },
    magnitude={
      breakpoints.no <- length(Magnitude[
        abs(Magnitude)>
        change.magnitude])
    }
  ) ## end switch option

  segment_no <- Param_no-1
  BP_no <- breakpoints.no

  Output.BreakpointNo <- breakpoints.no
  Output.SegmentNo <- segment_no
  Output.Fit <- fit

  if(breakpoints.no == 0) {

    Output.Start <- c()
    Output.Duration <- c()
    Output.End <- c()
    Output.Change <- c()
    Output.ChangeType <- c()
    Output.Significance <- c()

  } else {

    Output.Start <- Start[1:breakpoints.no]

```

```

        Output.Duration <- Duration[1:
                                breakpoints.no]
        Output.End <- End[1:breakpoints.no]
        Output.Change <- Magnitude[1:breakpoints.no]
        Output.ChangeType <- match(Start[1:
                                breakpoints.no],vec)
        Output.ChangeType[
            !is.na(Output.ChangeType)] <- 1
        Output.ChangeType[
            is.na(Output.ChangeType)] <- 0
        Output.Significance= match(Start[
            1:breakpoints.no],Sig_BP_position);
        Output.Significance[
            !is.na(Output.Significance)] <- 1
        Output.Significance[
            is.na(Output.Significance)] <- 0
    }

    DBEST.values <- list(
        "BreakpointNo" = Output.BreakpointNo,
        "SegmentNo" = Output.SegmentNo,
        "Start" = Output.Start,
        "Duration" = Output.Duration,
        "End" = Output.End,
        "Change" = Output.Change,
        "ChangeType" = Output.ChangeType,
        "Significance" = Output.Significance,
        "Fit" = Output.Fit
    )
}

) ## end switch algorithm

DBEST.values <- append(DBEST.values, list(Data=data))

if(exists("trend")) {
    DBEST.values <- append(DBEST.values, list(Trend=trend))
}

if(exists("seasonal")) {
    DBEST.values <- append(DBEST.values,
        list(Seasonal=seasonal))
}

if(exists("remainder")) {
    DBEST.values <- append(DBEST.values,
        list(Remainder=remainder))
}

if(exists("f_local")) {
    DBEST.values <- append(DBEST.values, list(f_local=f_local))
}

class(DBEST.values) <- "DBEST"

if(plot!=-1) {
    if(is.character(plot)) {
        plot <- tolower(plot)
    }
}

```

```

        if(plot=="on") {
            plot.DBEST(DBEST.values, figure=1)
            plot.DBEST(DBEST.values, figure=2)
        }

        if(plot=="fig1") {
            plot.DBEST(DBEST.values, figure=1)
        }

        if(plot=="fig2") {
            plot.DBEST(DBEST.values, figure=2)
        }

    }

    return(DBEST.values)
}

print.DBEST <- function(x, ...) {

    cat("\nDBEST Results:\n")

    if(exists("RMSE", where=x)){

        cat("Number of segments: ", x$SegmentNo,"\n")
        cat("RMSE: ", x$RMSE, "\n")
        cat("MAD: ", x$MAD, "\n")
        cat("Fit: ", x$Fit, "\n")

    } else {

        cat("Number of break points: ", x$BreakpointNo, "\n")
        cat("Number of segments: ", x$SegmentNo, "\n")
        cat("Starting at: ", x$Start, "\n")
        cat("Duration: ", x$Duration, "\n")
        cat("Ending at: ", x$End, "\n")
        cat("Change: ", x$Change, "\n")
        cat("Change type: ", x$ChangeType, "\n")
        cat("Significance: ", x$Significance, "\n")
        cat("Fit: ", x$Fit, "\n")

    }

    invisible(x)

}

plot.DBEST <-function(x, figure=1) {

    if(figure==1) {

        par(mfrow=c(2,1))
        par(mar=c(0,5,3,3))
        plot(x$Data, main="DBEST fig.1", ylab="Data", xaxt='n',
            yaxt='n',cex.lab=0.8)
        axis(side=4)

        fit_zoo <- as.zoo(ts(x$Fit, start=as.Date(time(x$Data)[1]),
            frequency=frequency(x$Data)))
        lines(fit_zoo, lwd=2, col="blue")

        if(exists("Start", where=x)){
            if(length(x$Start)>0) {

```



```

        for(t in 1:length(x$Start)) {
            segm <- fit_zoo[
                x$Start[t]:x$End[t]]
            if(x$ChangeType[t] == 1) {
                lines(segm, lwd=2,
                    col="red")
            } else {
                lines(segm, lwd=2,
                    col="orange")
            }
        }
    }
}

par(mar=c(8,5,0,3))
f_local_zoo <- as.zoo(ts(x$f_local, start=as.Date(
    time(x$Data)[1]), frequency=frequency(x$Data)))
plot(f_local_zoo, main=NULL, type = "n", yaxt = "n",
    bty="n", ylab = "", xlab="")

par(new=T)
barplot(f_local_zoo, main=NULL, ylab="Trend Local Change",
    xaxt='n', xlab="", cex.lab=0.8, border='red',
    col='red', space=100)
box()
}

if(figure==2 & exists("Seasonal", where=x)) {

    par(mfrow=c(4,1))
    par(mar=c(0,5,3,3))
    plot(x$Data, main="DBEST fig.2", ylab="Data", xaxt='n',
        yaxt='n')
    axis(side=4)

    par(mar=c(0,5,0,3))
    trend_zoo <- as.zoo(ts(x$Trend,
        start=as.Date(time(x$Data)[1]),
        frequency=frequency(x$Data)))
    plot(trend_zoo, type="l", ylab="Trend", xaxt='n',
        main=NULL)

    par(mar=c(0,5,0,3))
    seasonal_zoo <- as.zoo(ts(x$Seasonal, start=as.Date(
        time(x$Data)[1]),
        frequency=frequency(x$Data)))
    plot(seasonal_zoo, ylab="Seasonal", xaxt='n', yaxt='n',
        main=NULL)
    axis(side=4)

    par(mar=c(3,5,0,3))
    remainder_zoo <- as.zoo(ts(x$Remainder,
        start=time(x$Data)[1],
        frequency=frequency(x$Data)))
    plot(remainder_zoo, main=NULL, type = "n", yaxt = "n",
        bty="n", ylab = "")

    par(new=T)
    barplot(remainder_zoo, ylab="Remainder", main=NULL,

```

```

                                xaxt='n', border='#808080',
                                col='#808080', space=100)
                                box()
                                }
}

```

## Filename: Trend\_local\_change.R

```

Trend_local_change <- function(data_series,epsilon,vec) {

  Length <- length(data_series);
  n_turning <- 0
  Distance_points <- 999
  Turning <- c()
  f_sign <- c()
  f_local <- c()

  if(epsilon<=0) {
    n_turning <- Length-1
    f_sign <- rep(1,Length-1)
    f_sign[1] <- 0
    Turning <- seq(Length-1)
    RL_DIST <- rep(0,(Length-1))
  } else {
    delta1 <- 0
    delta2 <- 0
    ck <- 2
    for(k in 2:(Length-1)) {

      delta1[ck]=data_series[ck]-data_series[ck -1]
      delta2[ck]=data_series[ck + 1]-data_series[ck]

      if( sign(delta1[ck]) != sign(delta2[ck]) ) {
        f_sign[ck] <- 1
        n_turning <- n_turning + 1
        Turning[n_turning] <- ck
      } else {
        f_sign[ck] <- 0
      }
      ck <- ck + 1
    }

    f_sign[1] <- 1
    n_turning <- n_turning + 1
    f_sign[Length] <- 0

    while(length(Distance_points) != 0) {

      vertexs <- c(1,Turning,Length)

      righth_dist1 <- c()
      left_dist1 <- c()
      righth_dist2 <- c()
      left_dist2 <- c()

      Righth_Dist <- c()
      Left_Dist <- c()
      righth_dist_max <- 0
      left_dist_max <- 0

      for(i in 1:n_turning) {

```

```

v1_x <- vertexs[i]
v2_x <- vertexs[i+1]
v1_y <- data_series[v1_x]
v2_y <- data_series[v2_x]
points_within <- v2_x - v1_x

if(points_within>1) {
  line_slope <- (v2_y-v1_y)/
                points_within

  for(j in 1:(points_within-1)){

    point_slope <- (
      data_series[
        v1_x+j]-
        v1_y)/j
    if(point_slope<=
      line_slope) {
righth_dist1[j] <- v1_x+j
righth_dist2[j] <- Point_to_line( c(v1_x+j,
  data_series[v1_x+j],0),
  c(v1_x,v1_y,0), c(v2_x,v2_y,0) )
righth_dist1[is.na(righth_dist1)] <- 0
righth_dist2[is.na(righth_dist2)] <- 0
    } else {
left_dist1[j] <- v1_x+j
left_dist2[j] <- Point_to_line(c(v1_x+j,
  data_series[v1_x+j],0),
  c(v1_x,v1_y,0), c(v2_x,v2_y,0))
left_dist1[is.na(left_dist1)] <- 0
left_dist2[is.na(left_dist2)] <- 0
    }

  } ##end for

  if( length(righth_dist1) > 0 ) {
righth_mx_idx <- which(righth_dist2==max(
  righth_dist2))
righth_dist_max <- cbind(righth_dist1[
  righth_mx_idx], righth_dist2[
  righth_mx_idx])
righth_dist1 <- c()
righth_dist2 <- c()
Righth_Dist <- rbind(Righth_Dist,
  righth_dist_max)
  }

  if( length(left_dist1) > 0 ) {
left_mx_idx <- which(
  left_dist2==max(left_dist2))
left_dist_max <- cbind(left_dist1[
  left_mx_idx],
  left_dist2[left_mx_idx])
left_dist1 <- c()
left_dist2 <- c()
Left_Dist <- rbind(Left_Dist, left_dist_max)
  }

} ## end points_within

```

```

    } ## end for

    RL_Dist <- rbind(Rigth_Dist,Left_Dist)

    if(length(RL_Dist) != 0) {
      RL_DIST <- RL_Dist[,2] # /case sensitivity
    } else {
      RL_DIST <- c()
    }

    Distance_points <- NULL
    size_RL_Dist <- dim(RL_Dist)

    for(i in 1:size_RL_Dist[1]) {
      if(RL_Dist[i,2]>epsilon) {
        Distance_points <-
          c(Distance_points,RL_Dist[i,1])
      }
    }

    n_turning <- n_turning + length(Distance_points)
    f_sign[Distance_points] <- 1
    Turning <- sort(c(Turning,Distance_points))

  } ## end while

  if(length(vec)>0) {
    for(i in 1:length(vec)){
      if(f_sign[vec[i]]!=1){
        f_sign[vec[i]] <- 1
        n_turning <- n_turning+1
        Turning <- c(Turning,vec[i])
      }
    }
  }

  Turning <- sort(c(1,Turning))
} ## end else

f_diff <- rep(0,Length)

for(i in 1:n_turning-1) {
  M_current <- data_series[Turning[i]]
  M_after <- data_series[Turning[i+1]]
  f_diff[Turning[i]] <- M_after - M_current
}

f_diff[Turning[n_turning]] <- data_series[length(data_series)] -
  data_series[Turning[n_turning]]

for(k in 1:(Length-1)) {
  f_local[k] <- f_sign[k] * f_diff[k]
}
f_local[Length] <- 0

TLC.values <- list(
  "n_turning" = n_turning,
  "f_local" = f_local,
  "f_sign" = f_sign,
  "Turning" = Turning,
  "RL_DIST" = RL_DIST

```

```

)
class(TLC.values) <- "TLC"
return(TLC.values)
}

```

Filename: f\_level\_shift.R

---

```

f_level_shift <- function(data_series,threshold1,threshold2,duration) {

  diff1 <- data_series[-1] - data_series[-length(data_series)]
  Index <- which(abs(diff1)>threshold1)
  diff2 <- rep(0, length(Index))
  number_temp <- 0
  number <- 0
  LS_candidate <- c()
  LS_mag <- c()
  LS_point <- c()

  if(length(Index) > 0) {
    for(i in 1:length(Index)) {
      Up <- Index[i] + duration
      Down <- Index[i] - duration

      if ( (Up > length(data_series)) || (Down < 1) ) {
        diff2[i] <- 0
      } else {
        diff2[i] <- mean( data_series[
          Index[i]:Up] ) -
          mean( data_series[Down:Index[i]] )
      }

      if( (abs(diff2[i]) > threshold2) &&
          (sign(diff1[
            Index[i]]) == sign(diff2[i])))){

        number_temp <- number_temp + 1
        LS_candidate[number_temp] <- Index[i]
        LS_mag[number_temp] <- diff2[i]
      }
    }
  }

  if(number_temp > 0) {
    absLS_mag <- abs(LS_mag)
    LS_mag_sorted <- absLS_mag[order(-absLS_mag)]
    index_sort <- order(-absLS_mag)
    number <- 1
    LS_point[number] <- LS_candidate[index_sort[1]]

    if(number_temp>1) {
      ci <- 2
      for(i in 2:number_temp) {
        t <- 0
        for(j in 1:number) {
          if( abs(LS_candidate[
            index_sort[ci]] -
            LS_point[j]) > duration ) {

```

```

        t <- t + 1
    }
    }
    if (t == number) {
        number <- number + 1
        LS_point[number] <- LS_candidate[
            index_sort[ci]]
    }
    ci <- ci + 1
}
}
}
}
}

flevel.values <- list(
    "no" = number,
    "LS_point" = LS_point
)
class(flevel.values) <- "flevel"
return(flevel.values)
}

```

### Filename: Compute\_epsilon.R

```

Compute_epsilon <- function(pixeldata,epsilon_max,vec) {

    TLC <- Trend_local_change(pixeldata,epsilon_max,vec)
    turning_no_max <- TLC$n_turning
    f_local_max <- TLC$f_local
    f_sign_max <- TLC$f_sign
    Turning_max <- TLC$Turning
    RL_DISTmax <- TLC$RL_DIST
    DTSBP <- DBEST_timeseries_BP(pixeldata,turning_no_max,f_local_max,
        turning_no_max,f_sign_max,Turning_max)

    RSS <- DTSBP$RSS
    RMSE <- DTSBP$RMSE
    MAD <- DTSBP$MAD
    RMSE_max <- RMSE
    epsilon <- 3*(RMSE_max)
    return(epsilon)
}

```

### Filename: DBEST\_timeseries\_BP.R

```

DBEST_timeseries_BP <- function(data_series,n_BP,f_local,
    n_turning,f_sign,Turning){

    Length <- length(data_series)
    f_local_abs <- abs(f_local)
    ith_max <- c()
    BP_points <- c()
    B1 <- c()
    B2 <- c()
    B3 <- c()
    B4 <- c()
    detrend_data_series <- c()
    f_sorted <- c()
    Index <- c()
    Start <- c()
    End <- c()
}

```

```

Duration <- c()
Magnitude <- c()
BP_order <- c()
BP_mark <- c()
Tempnew <- f_local_abs[Turning]
f_sorted <- Tempnew[order(-Tempnew)]
Index <- order(-Tempnew)
Index <- Turning[Index]
ith_max <- matrix(0,Length,n_BP)

if(n_BP>0) {
  for(i in 1:n_BP) {
    ith_max[which(f_local_abs==f_sorted[i]),i] <- 1
  }
}

BP_points <- rep(0,length(data_series))

if(n_BP>0) {
  BP_points[Index[1:n_BP]] <- 1
}

if(n_BP==n_turning) {
  DETREND_DS <- DETREND_SJ(data_series,'linear',
                           which(BP_points==1))
  detrend_data_series <- DETREND_DS$y
  param_no <- DETREND_DS$param_no
  p <- DETREND_DS$p
  R <- DETREND_DS$R
  dfsJ <- DETREND_DS$dfsJ
  normr <- DETREND_DS$normr
} else {
  B1 <- which(BP_points==1)
  if(n_BP > 0){
    for(i in 1:n_BP){
      B2 <- which(f_sign==1)
      B3 <- B2[B2>B1[i]]
      if(length(B3)==0) {
        B4 <- B4
      } else {
        B4 <- c(B4,B3[1])
      }
    }
    DETREND_DS <- DETREND_SJ(data_series,'linear',
                             c(which(BP_points==1),B4))
  } else {
    DETREND_DS <- DETREND_SJ(data_series,'linear')
  }

  detrend_data_series <- DETREND_DS$y
  param_no <- DETREND_DS$param_no
  p <- DETREND_DS$p
  R <- DETREND_DS$R
  dfsJ <- DETREND_DS$dfsJ
  normr <- DETREND_DS$normr
}

```

```

}

temp <- sort(c(which(BP_points==1), B4))
b <- unique(temp)
m1 <- match(b,temp)
c1 <- sort(m1)
d1 <- match(c1,m1)
b <- b[d1]
shared <-b
RSS <- sum(detrend_data_series^2)
trend_data_series <- data_series - detrend_data_series

if(n_BP>0){
  Start<-which(BP_points==1)

  for(i in 1:n_BP) {
    if(Turning[length(Turning)] == Start[i]) {
      End[i] <- Length
    } else {
      End[i] <- Turning[which(
        Turning==Start[i])+1]
    }
  }
  # end for

  Duration <- End-Start
  Magnitude <- trend_data_series[End]-
    trend_data_series[Start]

  absMagnitude <- abs(Magnitude)
  af <- absMagnitude[order(-absMagnitude)]
  bf <- order(-absMagnitude)
  Start <- Start[bf]
  End <- End[bf]
  Magnitude <- Magnitude[bf]
  Duration <- Duration[bf]
}

BP_mark <- matrix(0,Length,n_BP)

if(n_BP>0) {
  for(i in 1:n_BP) {
    BP_mark[,i] <- data_series * ith_max[,i]
  }
}

BP_order <- BP_mark
BP_mark[BP_mark==0] <- NaN
RMSE= sum(abs(detrend_data_series)^2)^(1/2) /
  sqrt(length(detrend_data_series))
MAD <- max(abs(detrend_data_series))
DTSBP.values <- list(
  "Start" = Start,
  "End" = End,
  "Duration" = Duration,
  "Magnitude" = Magnitude,
  "BP_mark" = BP_mark,
  "BP_order" = BP_order,
  "trend_data_series" = trend_data_series,
  "Param_no" = param_no,
  "RSS" = RSS,
  "RMSE" = RMSE,
  "MAD" = MAD,

```



```

        "p" = p,
        "R" = R,
        "dfSJ" = dfSJ,
        "normr" = normr,
        "shared" = shared
    )

    class(DTSBP.values) <- "DTSBP"
    return(DTSBP.values)
}

```

## Filename: DBEST\_timeseries\_Threshold.R

```

DBEST_timeseries_Threshold <- function(data_series,beta,f_local,
                                       n_turning,f_sign,Turning,BIC_no){

    Length <- length(data_series)
    f_local_abs <- abs(f_local)
    ith_max <- c()
    BP_points <- c()
    B1 <- c()
    B2 <- c()
    B3 <- c()
    B4 <- c()
    detrend_data_series <- c()
    f_sorted <- c()
    Index <- c()
    Start <- c()
    End <- c()
    Duration <- c()
    Magnitude <- c()
    BP_order <- c()
    BP_mark <- c()
    Tempnew <- f_local_abs[Turning]
    f_sorted <- Tempnew[order(-Tempnew)]
    Index <- order(-Tempnew)
    Index <- Turning[Index]
    n_BP <- length(f_local_abs[which(f_local_abs>beta)])

    if(n_BP>BIC_no){
        n_BP <- BIC_no
    }

    ith_max <- matrix(0,Length,n_BP)

    if(n_BP > 0) {
        for(i in 1:n_BP) {
            ith_max[which(f_local_abs==f_sorted[i]),i] <- 1
        }
    }

    BP_points <- rep(0,length(data_series))

    if(n_BP > 0) {
        BP_points[Index[1:n_BP]] <- 1
    }

    if(n_BP==n_turning) {

```

```

DETREND_DS <- DETREND_SJ(data_series, 'linear',
                          which(BP_points==1))
detrrend_data_series <- DETREND_DS$y
param_no <- DETREND_DS$param_no

} else {

  B1 <- which(BP_points==1)

  if(n_BP > 0) {
    for(i in 1:n_BP){
      B2 <- which(f_sign==1)
      B3 <- B2[B2>B1[i]]
      if(length(B3)==0) {
        B4 <- B4
      } else {
        B4 <- c(B4,B3[1])
      }
    }

    DETREND_DS <- DETREND_SJ(data_series, 'linear',
                             c(which(BP_points==1),B4))
  } else {

    DETREND_DS <- DETREND_SJ(data_series, 'linear')
  }

  detrend_data_series <- DETREND_DS$y
  param_no <- DETREND_DS$param_no
}
temp <- sort(c(which(BP_points==1), B4))
b <- unique(temp)
m1 <- match(b,temp)
c1 <- sort(m1)
d1 <- match(c1,m1)
b <- b[d1]
shared <-b
RSS <- sum(detrrend_data_series^2)
trend_data_series <- data_series - detrend_data_series

if(n_BP>0){
  Start<-which(BP_points==1)

  for(i in 1:n_BP) {
    if(Turning[length(Turning)] == Start[i]) {
      End[i] <- Length
    } else {
      End[i]= Turning[which(Turning==Start[i])+1]
    }
  }
  # end for

  Duration <- End-Start
  Magnitude <- trend_data_series[End]-
                trend_data_series[Start]

  absMagnitude <- abs(Magnitude)
  af <- absMagnitude[order(-absMagnitude)]
  bf <- order(-absMagnitude)
  Start <- Start[bf]
  End <- End[bf]
  Magnitude <- Magnitude[bf]
  Duration <- Duration[bf]
}

```

```

}

BP_mark <- matrix(0,Length,n_BP)

if(n_BP > 0) {
  for(i in 1:n_BP) {
    BP_mark[,i] <- data_series * ith_max[,i]
  }
}

BP_order <- BP_mark
BP_mark[BP_mark==0] <- NaN
RMSE <- sum(abs(detrend_data_series)^2)^(1/2) /
sqrt(length(detrend_data_series))
MAD <- max(abs(detrend_data_series))
DTSTH.values <- list(
  "n_BP" = n_BP,
  "Start" = Start,
  "End" = End,
  "Duration" = Duration,
  "Magnitude" = Magnitude,
  "BP_mark" = BP_mark,
  "BP_order" = BP_order,
  "trend_data_series" = trend_data_series,
  "Param_no" = param_no,
  "RSS" = RSS,
  "RMSE" = RMSE,
  "MAD" = MAD,
  "shared" = shared
)
class(DTSTH.values) <- "DTSTH"
return(DTSTH.values)
}

```

Filename: BIC\_BP.R

```

BIC_BP <- function(pixeldata,f_local,turning_no,f_sign,turning) {

  BICBP <- c()
  N <- length(pixeldata)
  DETREND_DS <- DETREND_SJ(pixeldata,'linear',c())
  detrend_data_series <- DETREND_DS$y
  param_no <- DETREND_DS$param_no
  RSS <- sum(detrend_data_series^2)
  BICBP <- c(BICBP,param_no*log(N)+N*(log(2*pi*RSS/N)+1))
  BICBP_Idx <- c(0)
  for(i in 1:turning_no) {

    DTSBP <- DBEST_timeseries_BP(pixeldata,i,f_local,
                                turning_no,f_sign,turning)

    Param_no <- DTSBP$Param_no
    RSS <- DTSBP$RSS
    BICBP_Idx <- c(BICBP_Idx, i)
    BICBP <- c(BICBP,Param_no*log(N)+N*(log(2*pi*RSS/N)+1))
  }

  sortBICBP <- sort(BICBP)
  BIC_BP_ord <- order(match(BICBP,sortBICBP))
  BIC_BP_no <- BICBP_Idx[BIC_BP_ord[1]]
  return(BIC_BP_no)
}

```

### Filename: Point\_to\_line.R

```
Point_to_line <- function(pt, v1, v2) {  
  
  a <- v1 - v2  
  b <- pt - v2  
  
  dim(a) <- c(1,3)  
  dim(b) <- c(1,3)  
  
  cprd <- c()  
  cprd[1] = (a[2]*b[3]) - (b[2]*a[3])  
  cprd[2] = (b[1]*a[3]) - (a[1]*b[3])  
  cprd[3] = (a[1]*b[2]) - (b[1]*a[2])  
  
  dim(cprd) <- c(1,3)  
  d <- max(svd(cprd)$d) / max(svd(a)$d)  
  return(d)  
}
```

### Filename: Pvalue\_ith\_coeff.R

```
Pvalue_ith_coeff <- function(coef_vec,R,normr,dfSJ,i) {  
  
  VC <- (solve(R)) %*% t(solve(R)) * (normr^2)/dfSJ  
  VCi <- sqrt(VC[i,i])  
  statistic <- (coef_vec[i])/VCi  
  p <- 2 * (1-pt(abs(statistic),dfSJ))  
  return(p)  
}
```

### Filename: Trend\_local\_change.R

```
DETREND_SJ <- function(x, tt = 'linear', bp = c()) {  
  
  if (!is.numeric(x) && !is.complex(x))  
    stop("'x' must be a numeric or complex vector or matrix.")  
  trendType <- pmatch(tt, c('constant', 'linear'), nomatch = 0)  
  
  if (is.vector(x))  
    x <- as.matrix(x)  
  n <- nrow(x)  
  if (length(bp) > 0 && !all(bp %in% 1:n))  
    stop("Breakpoints 'bp' must elements of 1:length(x).")  
  
  if (trendType == 1) { # 'constant'  
    if (!is.null(bp))  
      warning("Breakpoints not used for 'constant'  
              trend type.")  
    y <- x - matrix(1, n, 1) %*% apply(x, 2, mean)  
  } else if (trendType == 2) { # 'linear'  
    if(length(bp)==0) {  
      bp <- 1  
    }  
  
    bp <- sort(unique(c(1, c(bp), n))) ## both ends  
    lbp <- length(bp)  
    lb <- length(bp) - 1  
  }  
}
```

```

a <- matrix(0, n, lbp)
a[1:n,1] <- as.matrix(1:n)/n
if(lbp>2) {
  ktmp <- 2

  for(kt in 2:(lbp-1)) {
    m <- n - bp[ktmp]
    a[(bp[ktmp]+1):n,ktmp] <- as.matrix(1:m)/m
    ktmp <- ktmp + 1
  }

}
a[1:n,lbp] <- 1
y <- x - a %**% mldivide(a, x)
A <- matrix(0,n,lbp)

if(lbp>1) {
  bp[length(bp)] <- n-1 ##added later
  for(k in 1:(lbp-1)) {
    A[(bp[k]+1):bp[k+1],k] <- 1:(bp[k+1]-bp[k])
    A[(bp[k+1]+1):n,k] <- A[bp[k+1],k]
  }
}

A[1:n,lbp] <- 1
z <- qr(A)
Q <- qr.Q(z)
R <- qr.R(z)
p <- qr.solve(R, (t(Q)%**%x) )
#r <- x - A%**p
param_no <- length(p)
H <- A %**% solve( t(A) %**% A) %**% t(A)
dfSJ <- length(x) - 1.25 * sum(diag(H)) + 0.5
normr <- max(svd(y)$d)

} else {
  stop("Trend type 'tt' must be 'constant' or 'linear'.")
}

DTSJ.values <- list(
  "y" = y,
  "param_no" = param_no,
  "p" = p,
  "R" = R,
  "dfSJ" = dfSJ,
  "normr" = normr
)

class(DTSJ.values) <- "DTSJ"
return(DTSJ.values)
}

```



Series from Lund University

Department of Physical Geography and Ecosystem Science

**Master Thesis in Geographical Information Science**

1. *Anthony Lawther*: The application of GIS-based binary logistic regression for slope failure susceptibility mapping in the Western Grampian Mountains, Scotland. (2008).
2. *Rickard Hansen*: Daily mobility in Grenoble Metropolitan Region, France. Applied GIS methods in time geographical research. (2008).
3. *Emil Bayramov*: Environmental monitoring of bio-restoration activities using GIS and Remote Sensing. (2009).
4. *Rafael Villarreal Pacheco*: Applications of Geographic Information Systems as an analytical and visualization tool for mass real estate valuation: a case study of Fontibon District, Bogota, Columbia. (2009).
5. *Siri Oestreich Waage*: a case study of route solving for oversized transport: The use of GIS functionalities in transport of transformers, as part of maintaining a reliable power infrastructure (2010).
6. *Edgar Pimiento*: Shallow landslide susceptibility – Modelling and validation (2010).
7. *Martina Schäfer*: Near real-time mapping of floodwater mosquito breeding sites using aerial photographs (2010)
8. *August Pieter van Waarden-Nagel*: Land use evaluation to assess the outcome of the programme of rehabilitation measures for the river Rhine in the Netherlands (2010)
9. *Samira Muhammad*: Development and implementation of air quality data mart for Ontario, Canada: A case study of air quality in Ontario using OLAP tool. (2010)
10. *Fredros Oketch Okumu*: Using remotely sensed data to explore spatial and temporal relationships between photosynthetic productivity of vegetation and malaria transmission intensities in selected parts of Africa (2011)
11. *Svajunas Plunge*: Advanced decision support methods for solving diffuse water pollution problems (2011)
12. *Jonathan Higgins*: Monitoring urban growth in greater Lagos: A case study using GIS to monitor the urban growth of Lagos 1990 - 2008 and produce future growth prospects for the city (2011).
13. *Mårten Karlberg*: Mobile Map Client API: Design and Implementation for Android (2011).
14. *Jeanette McBride*: Mapping Chicago area urban tree canopy using color infrared imagery (2011)
15. *Andrew Farina*: Exploring the relationship between land surface temperature and vegetation abundance for urban heat island mitigation in Seville, Spain (2011)
16. *David Kanyari*: Nairobi City Journey Planner An online and a Mobile Application (2011)

17. *Laura V. Drews*: Multi-criteria GIS analysis for siting of small wind power plants - A case study from Berlin (2012)
18. *Qaisar Nadeem*: Best living neighborhood in the city - A GIS based multi criteria evaluation of ArRiyadh City (2012)
19. *Ahmed Mohamed El Saeid Mustafa*: Development of a photo voltaic building rooftop integration analysis tool for GIS for Dokki District, Cairo, Egypt (2012)
20. *Daniel Patrick Taylor*: Eastern Oyster Aquaculture: Estuarine Remediation via Site Suitability and Spatially Explicit Carrying Capacity Modeling in Virginia's Chesapeake Bay (2013)
21. *Angeleta Oveta Wilson*: A Participatory GIS approach to *unearthing* Manchester's Cultural Heritage 'gold mine' (2013)
22. *Ola Svensson*: Visibility and Tholos Tombs in the Messenian Landscape: A Comparative Case Study of the Pylion Hinterlands and the Soulima Valley (2013)
23. *Monika Ogden*: Land use impact on water quality in two river systems in South Africa (2013)
24. *Stefan Rova*: A GIS based approach assessing phosphorus load impact on Lake Flaten in Salem, Sweden (2013)
25. *Yann Buhot*: Analysis of the history of landscape changes over a period of 200 years. How can we predict past landscape pattern scenario and the impact on habitat diversity? (2013)
26. *Christina Fotiou*: Evaluating habitat suitability and spectral heterogeneity models to predict weed species presence (2014)
27. *Inese Linuza*: Accuracy Assessment in Glacier Change Analysis (2014)
28. *Agnieszka Griffin*: Domestic energy consumption and social living standards: a GIS analysis within the Greater London Authority area (2014)
29. *Brynja Guðmundsdóttir*: Detection of potential arable land with remote sensing and GIS - A Case Study for Kjósarhreppur (2014)
30. *Oleksandr Nekrasov*: Processing of MODIS Vegetation Indices for analysis of agricultural droughts in the southern Ukraine between the years 2000-2012 (2014)
31. *Sarah Tressel*: Recommendations for a polar Earth science portal in the context of Arctic Spatial Data Infrastructure (2014)
32. *Caroline Gevaert*: Combining Hyperspectral UAV and Multispectral Formosat-2 Imagery for Precision Agriculture Applications (2014).
33. *Salem Jamal-Uddeen*: Using GeoTools to implement the multi-criteria evaluation analysis - weighted linear combination model (2014)
34. *Samanah Seyedi-Shandiz*: Schematic representation of geographical railway network at the Swedish Transport Administration (2014)
35. *Kazi Masel Ullah*: Urban Land-use planning using Geographical Information System and analytical hierarchy process: case study Dhaka City (2014)
36. *Alexia Chang-Wailing Spitteler*: Development of a web application based on MCDA and GIS for the decision support of river and floodplain rehabilitation projects (2014)



37. *Alessandro De Martino*: Geographic accessibility analysis and evaluation of potential changes to the public transportation system in the City of Milan (2014)
38. *Alireza Mollasalehi*: GIS Based Modelling for Fuel Reduction Using Controlled Burn in Australia. Case Study: Logan City, QLD (2015)
39. *Negin A. Sanati*: Chronic Kidney Disease Mortality in Costa Rica; Geographical Distribution, Spatial Analysis and Non-traditional Risk Factors (2015)
40. *Karen McIntyre*: Benthic mapping of the Bluefields Bay fish sanctuary, Jamaica (2015)
41. *Kees van Duijvendijk*: Feasibility of a low-cost weather sensor network for agricultural purposes: A preliminary assessment (2015)
42. *Sebastian Andersson Hylander*: Evaluation of cultural ecosystem services using GIS (2015)
43. *Deborah Bowyer*: Measuring Urban Growth, Urban Form and Accessibility as Indicators of Urban Sprawl in Hamilton, New Zealand (2015)
44. *Stefan Arvidsson*: Relationship between tree species composition and phenology extracted from satellite data in Swedish forests (2015)
45. *Damián Giménez Cruz*: GIS-based optimal localisation of beekeeping in rural Kenya (2016)
46. *Alejandra Narváez Vallejo*: Can the introduction of the topographic indices in LPJ-GUESS improve the spatial representation of environmental variables? (2016)
47. *Anna Lundgren*: Development of a method for mapping the highest coastline in Sweden using breaklines extracted from high resolution digital elevation models. (2016)
48. *Oluwatomi Esther Adejoro*: Does location also matter? A spatial analysis of social achievements of young South Australians. (2016)
49. *Hristo Dobrev Tomov*: Automated temporal NDVI analysis over the Middle East for the period 1982 - 2010. (2016)