ZENTRUM FÜR GEOINFORMATIK
Paris-Lodron-Universität Salzburg

Master's Thesis

INTERACTIVE TIMESERIES ANALYSIS ON RASTERDATA USING QGIS, AN OPEN SOURCE GIS

WERNER MACHO

ZENTRUM FÜR GEOINFORMATIK
Paris-Lodron-Universität Salzburg

Master's Thesis

INTERACTIVE TIMESERIES ANALYSIS ON RASTERDATA USING QGIS, AN OPEN SOURCE GIS

INTERAKTIVE RASTERDATEN ZEITREIHENANALYSE MIT QGIS, EINEM OPEN SOURCE GIS

Zur Erlangung des Grades
„Master of Science (Geographical Information Science & Systems) – MSc(GIS)"

AUTHOR: DIPL.-ING.WERNER MACHO
U102723, UNIGIS MSC JAHRGANG 2012
SUPERVISOR: AO. UNIV. PROF. DR. JOSEF STROBL
ADVISOR: DIPL.-ING. MATTEO MATTIUZZI
DR. ANJA KLISCH
DATE: GMÜND, SEPTEMBER 30, 2016

"When it comes to software, I **much** prefer **free** software, because I have very seldom seen a program that has worked well enough for my needs, and having sources available can be a life-saver."

– Linus Torvalds

DECLARATION

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Gmünd, 30. September 2016

Werner Macho

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

API	Application Programming Interface
BFAST	Breaks For Additive Season and Trend
BOKU	University of Natural Resources and Life Sciences
CSV	Comma-Separated Values
ESRI	Environmental Systems Research Institut
EXIF	Exchangeable Image File Format
FOSS	Free and Open Source Software
GEOS	Geometry Engine - Open Source
GDAL	Geospatial Data Abstraction Library
GIS	Geographic Imformations System
GNU GPL	GNU General Public License
GPS	Global Positioning System
GRASS	Geographic Resources Analysis Support System
GRIB	GRIdded Binary or General Regularly-distributed Information in Binary form
GUI	Graphical User Interface
HDF	Hierarchical Data Format
IRC	Internet Relay Chat
IVFL	Institute of Surveying, Remote Sensing and Land Information
MODIS	Moderate resolution imaging spectroradiometer
MuTAnT	Multi Temporal Analysis Tool
NCSA	National Center for Supercomputing Applications

NetCDF	Network Common Data Form
NDVI	Normalized Difference Vegetation Index
OS	Operating System
OSGeo	Open Source Geospatial Foundation
PAM	Persitant Auxiliary Metadata
PAR	Photosynthetically Active Radiation
R	R language and environment for statistical computing
SPIRITS	Software for the Processing and Interpretation of Remotely sensed Image Time Series
TIMESAT	TIMESAT
UNO	United Nations Organization
VBA	Visual Basic for Applications
WAMIS	Wide Area Monitoring Information System
XML	Extensible Markup Language
XMP	Extensible Metadata Platform
QGIS	Q Geographic Information System

ACKNOWLEDGEMENT

It is not an easy task to write a master's thesis beside being included in the normal work process and having to do day to day work. You need good friends and a supporting family to go through this time.

First of all I have to thank Matteo Mattiuzzi for creating the idea for this thesis due to his needs for an interactive raster data time-series analysis tool to detect poppy seed fields in Afghanistan.

Many thanks also to my girlfriend Verena who initially got the idea to start with UniGIS and also had a tough time keeping me focused when I was close to stop the work on this thesis.

I would also like to thank my parents Walter and Gerlinde Macho who not only made it possible for me to study but also taught me a lot about nature and how things are always depending from each other. Thanks a lot for growing me up in a healthy environment surrounded by nature and the best playgrounds (strong and tall trees) a child could ever have. A big thanks goes out to my sister Birgitt, who was the first trying to teach me the most basic things (like reading) and raising my interest for science.

I also want to thank everyone from UniGIS for being patient and understanding that it is just sometimes not possible to deliver everything on time. Thanks to Prof. Fritz Strobl and Dr. Gudrun Walentin for proof reading my thesis and giving input for some parts I might have otherwise missed.

Last but not least I want to thank everyone from the Q Geographic Information System (QGIS) project in which I am involved since 2007. Especially Tim Sutton for opening my mind about Free and Open Source Software (FOSS) and showing me the possibilities of free GIS Software. Also Peter Wells and Martin Dobias from lutraconsulting for being such good friends having an open ear for all problems that arose during writing the necessary code and testing new things and always being available on Internet Relay Chat (IRC) for questions.

I would like to dedicate this work to my grandfather Adolf Binder († 1989) who is responsible for and raised my interest in technical processes and computers. He always supported me while growing up and initiated my programmer career by buying my first Computer - a Commodore C64.

ABSTRACT

This Master thesis is about interactively looking at remote sensing data (satellite data) and especially about creating an interactive tool for remote sensing analytics. Decision makers usually want to have a quick tool giving them the information they need without having to deal with long and complicated processing and processes in the background.

There is a variety of tools available to extract the values of raster data at a given coordinate while at the same time applying algorithms to it, but usually they are neither interactive nor „real-time“ and to my knowledge especially not both at the same time. This thesis focuses on the creation of an interactive tool for raster data analysis and time-series, appearing problems and their possible solutions. Furthermore it includes a description of algorithms and how to create and add new algorithms to the analysis and interact with the time-series raster data.

Today a lot of raster data is available free of charge. Moreover there is different software to calculate graphs and apply algorithms to complete a remote sensing workflow. To my knowledge all of the available FOSS software is lacking an „interactive“ mode to take a deeper look at the values on a selectable point and immediately display this values of all loaded raster-layers underneath it in an understandable way. Not to speak from an applied algorithm at this moment. Furthermore there is the problem of bringing these values in chronological order to display a time-series in a correct way.

Having an interactive tool to investigate time-varying raster data will replace a lot of currently available time-consuming work-flows. The current tools mostly work like this: One has to select a coordinate and probably also select and apply a certain algorithm to it. The values returned are then in turn used to draw a graph. A new, currently to be written tool would incorporate these steps and let the user interactively see the data below the mouse pointer.

The aim of this thesis is to investigate the need for and create an open-source tool to make all this possible in one go. Problems detected during the creation of this tool will be documented and hopefully be solved.

The tool being developed will be a python plugin for the FOSS QGIS and will obviously also be open-source and licensed under the GNU General Public License (GNU GPL).

There are papers about some software which is already available for such kind of procedure (timeseries, rasterdata), but up to the date starting to write this thesis (and the plugin) no software could be found that solves this problem interactively. Usually one always has to follow a workflow (with probably different software tools) to achieve the final result. Given the task at hand to create a new algorithm to do all the work in one step, I am not sure if I will reach the actual state of implementing an algorithm (or more algorithms). However, the plugin will be prepared to be extended with algorithms to be able to do more analysis with the collected time-series of raster data values and probably visualize this data in a very easy way.

The thesis is based on existing research (papers), which in term will be used to develop the plugin for interactive analysis. The focus of the thesis will be the plugin which will specialise and focus on the interactive implementation to investigate time-series based raster data in real time.

The initial idea for this thesis was born at the Institute of Surveying, Remote Sensing and Land Information at the University of Natural Resources and Life Sciences (BOKU) in Vienna. There will always be the risk that the implementation will not work, literature is available to get the algorithms and some reasonable workflows of other software to have a look how things are done in the best way.

MY HOBBY: GETTING LOST ON GOOGLE MAPS SATELLITE [1]

1 INTRODUCTION

Historically earth never stood still. Since the big bang until now there is a constant change on the surface of our planet. From short term changes as for example the growing of plants and vegetation to long term changes, like surface tectonics, there is always movement and dynamic on earth surface.

With the availability of having Global Positioning System (GPS) satellites circling in orbit, taking pictures and sending them back to earth there is a huge amount of data about different changes at the surface (and sometimes also a bit below the surface) is gathered. The best way for gathering this data is obviously taking pictures, a technique that has a long history and has the ability to make snapshots of the current situation. As science evolved it was soon possible to display changes that formerly were not able to be seen by the human eye but also in the bandwith outside of the human eye (ultraviolet and infrared) to gather even more specific data about earth and its changes.

To be able to realise what is going on and probably extrapolate (or interpolate) some data for prediction or just to proof what has happened or is currently happening, one has to collect all this data and take a closer look at the changes displayed over periods of time (time series).

This analysis of data within time series should afterwards be visualized to make the interpretation of the collected data easier and more understandable. The question raised is how to get time series visualized in a proper manner and how to make it easy enough to handle for everyone.

The purpose of scientific visualization is to graphically illustrate data to enable scientists to understand and glean insight from their data.

There is a need in sciences for computational tools to integrate large spatially distributed datasets to provide insight into the spatial and temporal domains of the data while allowing visualization, analysis in the spatial and temporal dimensions, data metrics, and pattern recognition in the same application.

1.1 BACKGROUND AND GENERAL IDEA

After doing some research which will be described in chapter 3 it was soon obvious that there is some software available to do raster data analysis. Even more there is also some software already available to do multi-temporal analysis on raster data. Yet, on the market one can still get no good software to do that **interactively and easy enough** to have quick results in a good visualization to make decisions based on that results. I personally have a background from the BOKU where people are in need of an interactive analysis tool. The idea of implementing this tool by myself was soon born. Also having a background in FOSS and QGIS it was clear to me what the solution has to look like: A plugin written in python for QGIS being open-source and available for everyone to use. 0

1.1.1 *Problem*

This thesis tries to answer and solve the following research questions:

1. Is it possible to create a easy to use and most important an interactive too with FOSS software for multi-temporal raster data analysis?

2. Can multi-temporal raster data be interactively examined per coordinate (pixel) to plot a date/time graph?

3. How can this be done in an interactive way to get fast results to scientists to make it also possible to interpret that data quickly?

While the first two points are partly solved in currently available software, it is also a fact that, especially in the Geographic Imformations System (GIS) sector, this software can be quite expensive for a normal user. The third point of being interactive is (to my knowledge) yet not solved in any type of available software.

There is some software which is very good on doing analysis and dealing with multi-temporal raster data and time series (see chapter 3), but the drawback in this software is mostly the lack of functionality, interactive useability or it is bound to either a proprietary GIS package and/or it is available on only one Operating System (OS).

The main problem to solve is to find a way to do everything with one software which should be easy enough to get quick results and help scientist to interactively understand a time series of raster data. Ideally this software should be OS independent and should work with freely available software (FOSS).

1.1.2 *Motivation*

The main motivation behind this thesis and the corresponding plugin is the lack of free software available to do exactly this type of job. The University of Natural Resources and Life Sciences BOKU has a demand for this type of workflow. Being involved in the QGIS project, I want to see if it is really possible to create such a tool with free software FOSS.

Without FOSS the modern digital life would be different. FOSS projects empower *Fortune 500* companies such as Google, Amazon, Red Hat or facebook. They also are key figures in the internet, running on most of the servers, but also on smartphones or other devices. However, FOSS projects do not only play an important role in the modern software ecosystem, there is also an active development community behind them which mostly consists of volunteers, but also companies. Lowdown such as the costs, the free availability of the source code and the number of people who work on a single project are certainly positive aspects of FOSS projects. They allow to use them without big investments and risks to implement an idea quickly and effectively. That is also a reason why I choose Free and Open Source Software FOSS to solve problems.

Beside using FOSS even propagating Open Standards for exchanging data and using available Open Data is also enough motivation to create a workflow, try to prototype a program and make all this available under the rules of the GNU GPL.

Many research papers exist about the motivation of developers or contributers to FOSS projects. Lakhani [11] for example examined why people do provide free support to other developers or users. They claim that most people do offer support because it returns direct learning benefits. Lerner [12] focus more on developers.

Also Loe [14] questioned why developers would work for free and came to the conclusion that a complex interaction between several technological, social and economic factors provide reasons for a developer's motivation.

Grazzini [5] implemented a web based survey analysing the answers of over 600 developers and over 287 projects. They consider external motivational factors as implausible and propose enjoyment-based intrinsic motivation as the main motivation for professionals but also for volunteers. A similar study, but limited to the Linux kernel was done by [6]. The findings were similar, however analysed from a psychological point of view.

So FOSS seems to give volunteers and professionals an opportunity of doing what they like.

I personally aim at giving something back. I have been using QGIS since more than 8 years and have been involved into the very helpful community. This has been enough motivation for me to create something new and useful which hopefully will be used by a lot of people like I have been using FOSS for a very long time.

Now it is the time for me to create something new, learn a lot while doing it and give something back to the community that helped me a lot during the last years to solve my very own problems with GIS.

1.2 OUTLINE OF THE THESIS

CHAPTER 1 – INTRODUCTION This chapter presents an overview of the thesis and introduces the reader to the problem and motivation of this analysis.

CHAPTER 2 – THEORETICAL BACKGROUND The theory behind the plugin, some algorithms and available basic data which can be used to do this type of research are presented in this chapter along with a theoretical background and an explanation of basic knowledge to understand the problems that appear and do this kind of analysis.

CHAPTER 3 – RELATED WORK An outline of related research in the field of FOSS, the development processes and software engineering methods are presented with appropriate references.

CHAPTER 4 – METHODOLOGY The methodology used will be provided in this chapter. This includes a general explanation of methods and software already available to solve the main problems investigated in this thesis along with solutions for the main problems that appeared.

CHAPTER 5 – RESULTS Problemes during writing the plugin and the thesis will be named here as well as the overall capabilities of the resulting plugin. As the plugin will be already published during the process of writing (following the best development process - release early, release often [20]) some case studies will be mentioned here.

CHAPTER 6 – CONCLUSION AND DISCUSSION The thesis finally concludes with a summary and draws together the main findings while writing the thesis and the corresponding software plugin along with possible future directions.

2 THEORETICAL BACKGROUND

In order to analyse and compare different work-flows we take a look at the theory behind the multi-temporal analysis, explain what raster data is and in which different formats of raster data it can appear. Give some examples of multi-temporal analysis and where it is necessary and helpful as well as start with some ideas how the processing will be done and which tools are best to use to implement it.

2.1 THEORY

Before computers were even able to read picture data, the usage of raster data, especially orthophotos (see section 2.2) to analyse different scopes of data and detect areas of the same "values" has been done manually by visually looking at them and drawing areas by hand. With the upcoming faster and faster developing computer technology and the ability of computers reading and modifying pictures the speed of reading and applying algorithms to alter the data now only depends on the resolution (and of course the size) of the raster data. This is becoming better and better with every year. Computers are now able to automate this work and combine areas of pixels with the same values which is called classification. Sometimes an operator has specific questions regarding pixel values and has to find decent pixel values in a picture or he knows the area of interest and wants to know the specific pixel value at a given coordinate.

To find an answer to these questions a researcher usually has to take a look at the complete time series of for e.g. satellite images at once. The problems start when you have to search for decent increase or decrease of a certain value over time. With currently available software (see Chapter 3 on page 23) one can take a look at one coordinate - calculate the time series and plot it, but for the next plot on a different coordinate one wants to take a look at, one has to step over the procedure of entering the coordinates, calculating and plotting again.

This is very time consuming and to ease exactly this task for scientist, the idea of having an interactive look at time series data was born. The actual kickoff for this master's thesis and its corresponding plugin was to interactively find the heyday for poppy seed in satellite imagery of Afghanistan by a demand from BOKU.

Starting with satellite images from the MODIS project (which provides vegetation index data from the specified area) and some precalculation (which was done at a server at BOKU using the MODIS package (see Section 2.5 on page 20) R-MODIS from the R statistics project), the main goal is to draw time series (see section 2.1.2 on page 9) plots interactively while moving the mouse cursor over the map canvas in an Open Source GIS.

2.1.1 *NDVI*

Most of this thesis and investigations will be based on NDVI data. Therefore here is a short explanation of what NDVI data is.

NDVI stands for normalized difference vegetation index and is a simple graphical indicator that can be used to analyse remote sensing measurements typically, but not necessarily, from a space platform and assess whether the target being observed contains live green vegetation or not.

Live green plants absorb solar radiation in the Photosynthetically Active Radiation (PAR) spectral region, which they use as a source of energy in the process of photosynthesis. Leaf cells have also evolved to re-emit solar radiation in the near-infrared spectral region (which carries approximately half of the total incoming solar energy), because the photon energy at wavelengths longer than about 700 nanometers is not large enough to synthesize organic molecules. A strong absorption at these wavelengths would only result in overheating the plant and possibly damaging the tissues. Hence, live green plants appear relatively dark in the PAR and relatively bright in the near-infrared. By contrast, clouds and snow tend to be rather bright in the red (as well as other visible wavelengths) and quite dark in the near-infrared.

The pigment in plant leaves, chlorophyll, strongly absorbs visible light (from 0.4 to 0.7 µm) for use in photosynthesis. The cell structure of the leaves, on the other hand, strongly reflects near-infrared light (from 0.7 to 1.1 µm). The more leaves a plant has, the more these wavelengths of light are affected, respectively. Since early instruments of Earth Observation, such as NASA's ERTS and NOAA's AVHRR, acquired data in visible and near-infrared, it was natural to exploit the strong differences in plant reflectance to determine their spatial distribution in these satellite images.

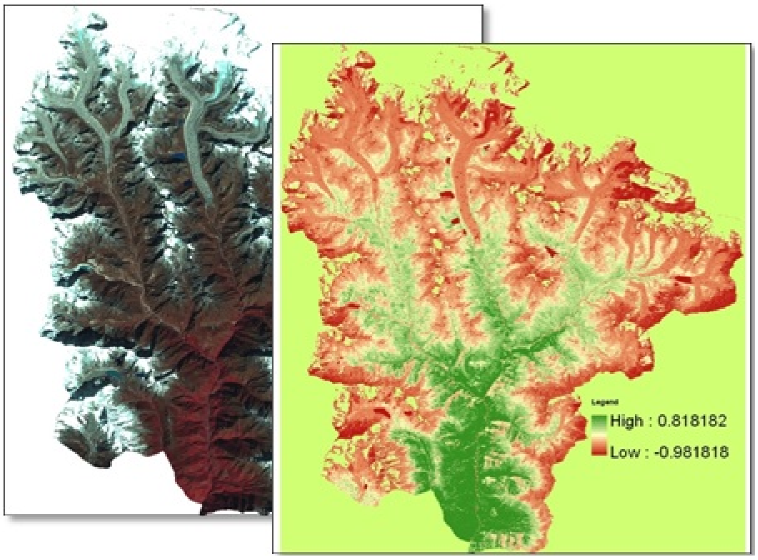

Figure 2.1: Generated Normalized Difference Vegetation Index (NDVI) from the satellite images

The NDVI is calculated from these individual measurements as follows:

$$\mathrm{NDVI} = \frac{(\mathrm{NIR} - \mathrm{VIS})}{(\mathrm{NIR} + \mathrm{VIS})} \tag{2.1}$$

where VIS and NIR stand for the spectral reflectance measurements acquired in the visible (red) and near-infrared regions, respectively (http://earthobservatory.nasa.gov/Features/MeasuringVegetation/measuring_vegetation_2.php).

These spectral reflectances are themselves ratios of the reflected over the incoming radiation in each spectral band individually, hence they take on values between 0.0 and 1.0. By design, the NDVI itself thus varies between -1.0 and +1.0. It should be noted that NDVI is functionally, but not linearly, equivalent to the simple infrared/red ratio (NIR/VIS). The advantage of NDVI over a simple infrared/red ratio is therefore generally limited to any possible linearity of its functional relationship with vegetation properties (e.g. biomass). The simple ratio (unlike NDVI) is always positive, which may have practical advantages, but it also has a mathematically infinite range (0 to infinity), which can be a practical disadvantage as compared to NDVI. [31]

2.1.2 *Time Series*

A time series is a series of data points listed (or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. To detect development in a specified area over time it is obvious that the values have to be in a time context. If there is a known starting point and time-frequency, it is enough to have only the values available. The example shown in figure 2.2 is a data set of the number of births per month in New York city, from January 1946 to December 1959. This data is available in the file http://robjhyndman.com/tsdldata/data/nybirths.dat

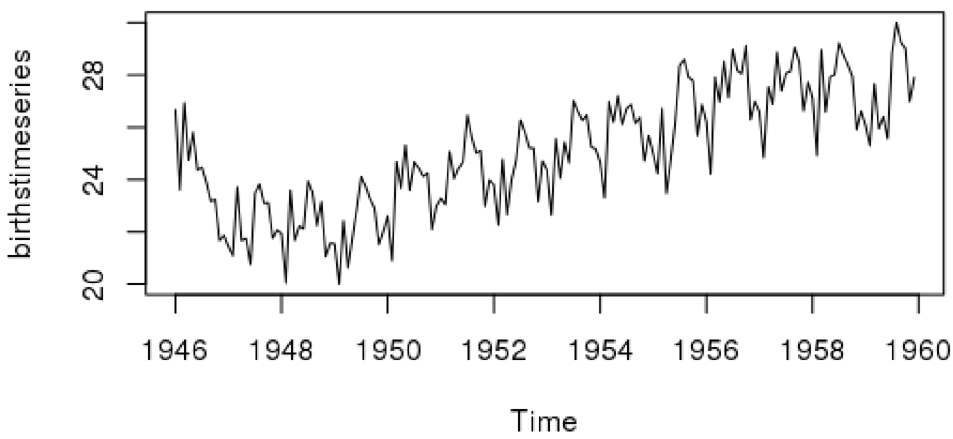

Figure 2.2: Time series plot example: Births per month from January 1946 to December 1959

In case of raster data every pixel has its value taken at exactly that time when the photo/satellite-image were taken. Obviously the raster data could also be altered manually by some kind of computer graphics program, but for this case we assume that we work with satellite images. Time series for our purposes are not always directly georeferenced, but can be indirectly georeferenced through a many-to-many relationship (many time series records can be related to many spatial features). But this referencing has to be solved in a way that sources of time from many places can be read and equally identified after classification.

2.1.3 *Remote Sensing*

Remote sensing is not really a multi-temporal analysis. Remote sensing is one of the top investigation methods to use when it comes to deal with earth surface data and satellite images. The way how remote sensing data is acquired can be seen in figure 2.3.

The use of remote sensing time series for crop and vegetation monitoring typically requires a number of processing steps that include the temporal smoothing of the cloud-affected remote sensing signal, the computation of LTA and associated variability, the computation of anomalies, the detection of plant phenology and the classification of the productivity level on the basis of seasonal performances.

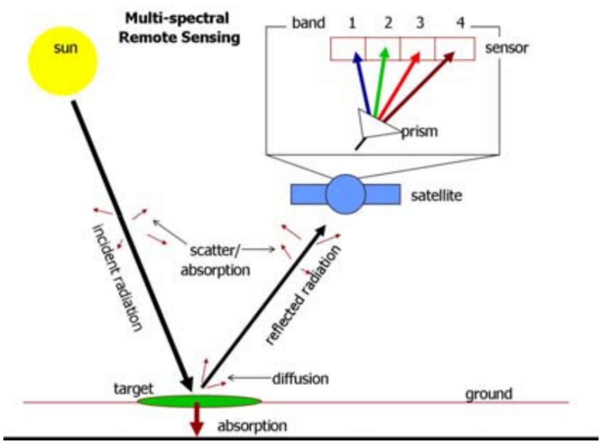

Figure 2.3: Acquiring data from satellite for remote sensing

Furthermore, the production of a crop monitoring bulletin (i.e., a report intended to illustrate the current agronomic situation to decision makers and non-specialists in general) summarizing all the information gathered from the analysis of the mentioned derived products, requires additional post-processing steps. These include for example the computation of statistics aggregated by administrative unit and by land cover class, and the generation of maps and graphs showing spatial patterns and temporal evolutions of relevant indicators [16]

After taking the remote sensing data from satellite there is still some work to do to get exactly the data out of the images one wants for investigations (see figure 2.4 on the following page). This is the processing chain - and exactly what we would like to help with with the plugin we intent to write for GIS analysis.

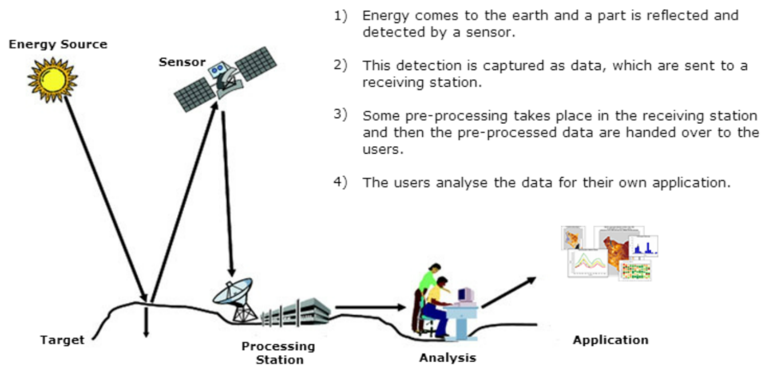

Figure 2.4: Processing chain for analysis of remote sensing data

2.1.4 *Analysis*

Time series analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data. Analysing raster data itself is not always an easy task but trying to analyse a time series of raster data is even more difficult and one can easily be confused with coordinates and values and the amount of data produced by these analysis. Every scientist has his own questions and therefore different methods available for time series analysis of raster data exists. Methods for time series analyses may be divided into two classes: frequency-domain methods and time-domain methods. The former includes spectral analysis and wavelet analysis; the latter include auto-correlation and cross-correlation analysis. As this thesis and its according software is written with some specific questions in mind , the analysis methods that are possible to be implemented to fit the goal of this thesis are mainly some sort of algorithms to analyse NDVI data (see 2.1.1 on page 7). Of course the plugin will be written with FOSS in mind and therefore will try to implement everything in the broadest possible way to give other sciences and visualisations the chance to use this software as a base for their own extensions and developments. Everyone is free and welcome to implement more analysis tools. The analysis tools used at BOKU are mentioned shortly and will probably be the first ones to be implemented:

2.1.4.1 *Interpolation*

Multi temporal data usually consists of a lot of values which sometimes have a huge differences. If you bring these values in a timely order there can be still huge gaps or there is a problem of missing values. To enable the scientist to better and easier interpret the data these differences and gaps have to be smoothed. One possibility of smoothening is interpolation, which can be done with several methods. Some of them are easy and some of them are more sophisticated.

In the mathematical field of numerical analysis, interpolation is a method of constructing new data points within the range of a discrete set of known data points.

In engineering and science, one often has a number of data points, obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable. It is often required to interpolate (i.e. estimate) the value of that function for an intermediate value of the independent variable. This may be achieved by curve fitting or regression analysis. This analysis method is possibly the easiest one and will be implemented as an example for more analysis tools for the plugin.

For reference some types of the easier interpolation methods have to be mentioned here, which are:

- linear interpolation
- cubic spline interpolation
- nearest neighbour interpolation

There are more analysis methods available to apply to NDVI data which will probably used to implement in the plugin later. The exact formulas of the algorithms can be found in specified publications, so there will only be a short description of the methods here. As an example only the Whittaker algorithm for smoothing will be described including its formulas (taken from Chountasis [2]).

2.1.4.2 *Whittaker*

The Whittaker [29] smoother can be considered as a special case of B-spline smoothing, in which the number of knots is equal to the number of data points. For a given noisy series y_i with m data points, a knot is assigned on each observed value of the signal. Then, instead of a basis matrix B that contains all the B-spline functions, a vector z is created of the same size as y. A penalized least squares algorithm is applied to minimize the quantity:

$$S = \sum_{i=1}^{m} (y_i - \hat{z}_i)^2 + \lambda \sum_{j=k+1}^{n} (\Delta^k z_j)^2 \qquad (2.2)$$

The scores of S with respect to vector z is equal to 0 giving a unique solution

$$\hat{z} = (I + \lambda D'D)^{-1} y \tag{2.3}$$

where I is the identity matrix and D is the matrix of differences. Once again λ plays a crucial role defining the smoothness of the series y_i. Very small values of λ result in wiggly fits near the data points, while on the other extreme, large values of the penalty weight will oversmooth the series, resulting in a straight line. The system of equations is easily handled when the number of observations is small, say up to 1000. For larger datasets Eilers describes an algorithm based on sparse matrices that works very efficiently. The algorithm can be easily implemented into Matlab or R with a few lines of code. The Whittaker smoother can be easily adapted to fit data with missing values and data that are arbitrarily non equally spaced [2].

Applying those algorithms to MODIS data is smoothing the graph a lot an lets see the trendline easier (see figure 2.5).

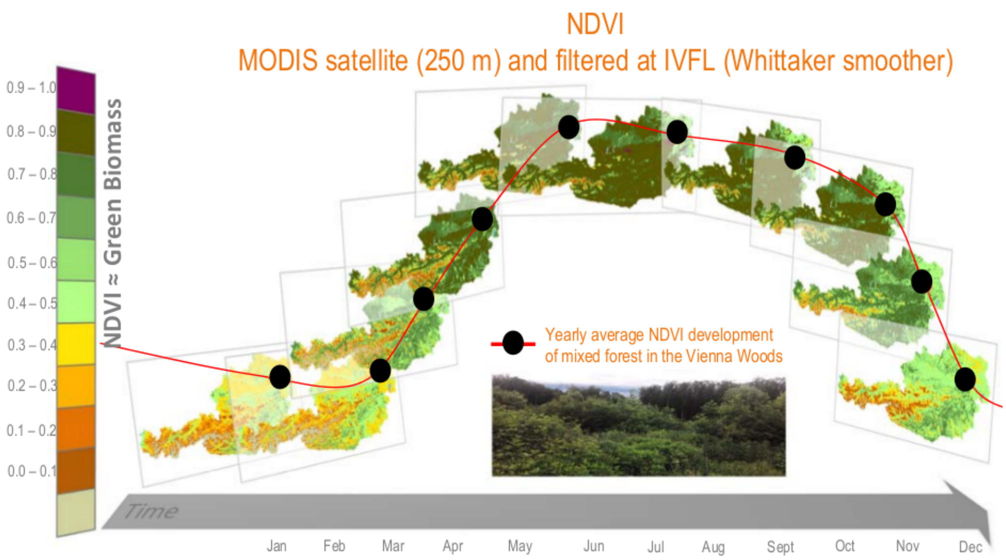

Figure 2.5: NDVI time series smoothed with Whittaker

2.1.4.3 *Double-logistic*

The logistic sigmoid curve is widely used in nonlinear regression and in binary response modeling. Problems arise from the double sigmoid behavior. This behavior has two parts - first increase to an early saturation at an intermediate level, and the second sigmoid with the eventual plateau of saturation. A double sigmoid behavior is usually achieved using additive or multiplicative combinations of logit and more complicated functions with numerous parameters. [13]

2.1.4.4 *Savitzky-Golay*

A Savitzky–Golay filter is a digital filter that can be applied to a set of digital data points for the purpose of smoothing the data, that is, to increase the signal-to-noise ratio without greatly distorting the signal [24].

This is achieved, in a process known as convolution, by fitting successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares. When the data points are equally spaced, an analytical solution to the least-squares equations can be found, in the form of a single set of "convolution coefficients" that can be applied to all data sub-sets, to give estimates of the smoothed signal, (or derivatives of the smoothed signal) at the central point of each sub-set. The method, based on established mathematical procedures, was popularized by Abraham Savitzky and Marcel J. E. Golay who published tables of convolution coefficients for various polynomials and sub-set sizes in 1964.

2.1.5 *Long time analyses*

If one takes a look at other sciences (compared to GIS) it is obvious that the analysis has to be either in a short time manner (chemistry) or a long time manner(land development). For example climate changes have to be investigated in a very long timescale. Vegetation is more or less in a yearly. And some chemical reactions would be finished within milliseconds. The scale obviously has to be adjusted to the data you look at. In terms of vegetation index data the time scale to use will surely be at least a few years to look at to identify a trend or spot the necessary values inside the raster data.

2.1.6 *Trend analysis*

Trend is a global polynomial interpolation that fits a smooth surface defined by a mathematical function (a polynomial) to the input sample points. The trend surface changes gradually and captures coarse-scale patterns in the data. Conceptually, trend interpolation is like taking a piece of paper and fitting it between raised points (raised to the height of value). A flat piece of paper will not accurately capture a landscape containing a valley. However, if you bend the piece of paper once, you will get a much better fit. Adding a term to the mathematical formula produces a similar result, a bend in the plane. A flat plane (no bend in the piece of paper) is a first-order polynomial (linear). Allowing for one bend is a second-order polynomial (quadratic), two bends a third-order (cubic), and so forth. Rarely will the piece of paper pass through the actual measured points, thus making trend interpolation an inexact interpolator. Some points will be above the piece of paper and others will be below. However, if you add up how much higher each point is above the piece of paper and add up how much lower each point is below the piece of paper, the two sums should be similar. The lower the root mean square (RMS) error, the more closely the interpolated surface represents the input points. The most common order of polynomials are one through three. Trend surface interpolation creates smooth surfaces.

2.1.7 *Visualising*

Visualising is one of the most important things for presenting data. Data can be accurate and scientific proof, but without a proper method of presentation or visualization it is useless for interpretation and getting the information one wants to get. Visualising multi-temporal data can be done in different ways but there always has to be a specific axis for the time data. Sometimes it could even be useful to present the data in a 3D matrix to extract more information and present more values.

For this thesis the main case was to visualise one band of multiband data, or the only band (for preprocessed satellite imagery you will get from MODIS). Time series animations work wonders, especially visualizing events. Imagine watching the melt of polar ice caps or the spread of a forest fires or diseases. Showing these types of events captures your audience by viewing obviously geographic changes in the landscape with time. While it is doable to create an animation with time series data a scientist would rather prefer a graph with decent values to have a clear picture of what he is looking at. And the shape of a graph always references to a recognisable event which gives the possibility to categorise the visualised data to an already investigated event. So this thesis and its corresponding plugin will try to visualise the output (which will be many tuples of date/value) in a more or less simple XY-Axis plot where the value will be on Y-Axis and the date will be drawn on the X-Axis.

2.2 RASTER DATA

Raster is a method for the storage, processing and display of spatial data. A raster dataset is composed of rows (running across) and columns (running down) of pixels (also know as cells). Each pixel represents a geographical region, and the value in that pixel represents some characteristic of that region. Each cell must be rectangular in shape, but not necessarily square. Each cell within this matrix contains location co-ordinates as well as an attribute value. The spatial location of each cell is implicitly contained within the ordering of the matrix, unlike a vector structure which stores topology explicitly. Areas containing the same attribute value are recognised as such, however, raster structures cannot identify the boundaries of such areas as polygons. The values inside the raster data can describe a lot of different things taken from real life. In figure 2.6 there is an example of how a raster data is built out of the distribution of points, where each point counts as one value. The darker (higher value) the square area in the raster, the more points are inside this square area.

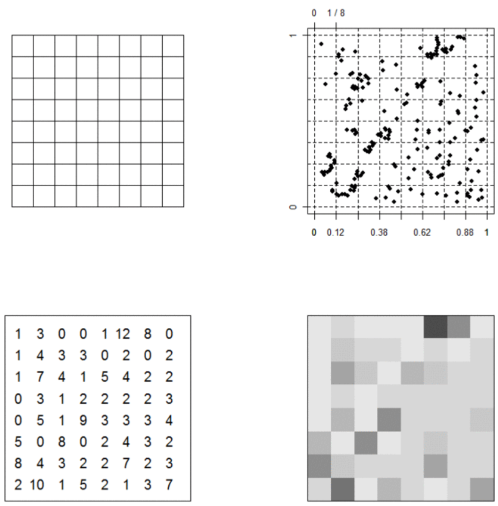

Figure 2.6: Example of Raster data © by Wikipedia User Ldecola licensed as CC BY-SA 3.0

Raster data is an abstraction of the real world where spatial data is expressed as a matrix of cells or pixels, with spatial position implicit in the ordering of pixels. With the raster data model, spatial data is not continuous but divided into discrete units. This makes raster data particularly suitable for certain types of spatial operation, for example overlays or area calculations.

Raster structures may lead to increased storage in certain situations, since they store each cell in the matrix regardless of whether it is a feature or simply 'empty' space.

2.2.1 *Raster Series*

Raster Series are a collection of rasters indexed by time. Each raster is a "snapshot" of the environment at some instant in time. Grouping a series of rasters can describe how the environment changes over time. This is called a time series. Raster series are useful for describing the dynamics of spatially continuous phenomena, like ponded depth in the Everglades, or rainfall measured by NEXRAD or in this case the growth of vegetation in a certain area. Some useful examples where analysis of time series of raster data is also useful can be seen in chapter 2.4 on page 19.

2.3 MULTI-TEMPORAL GIS FORMATS

There are a lot of existing GIS data formats around. Some are for vector formats and others are for raster formats. There are also databases which can store vector and raster formats but dealing with multi-temporal formats is a new challenge storing data on a computer. Temporal data has a time component attached to it. For example: Weather data usually uses temporal GIS data formats because time is an important factor related to weather and its development. Other examples of temporal data are demographic trends, land use patterns, lightning strikes and a lot more. For GIS there are some specific developed data formats to contain temporal data which are the following:

2.3.1 *NetCDF (Network Common Data Form)*

The Network Common Data Form (NetCDF) GIS format is an interface for array-oriented data for storing multi-dimensional variables. An example of a multi-dimension NetCDF could be temperature, precipitation or wind speed over time. It's commonly used for scientific data involved in the oceanic and atmospheric community as a GIS data storage format. The ArcGIS multidimensional toolbox and the QGIS NetCDF Browser both offer support for NetCDF files. The QGIS NetCDF Browser is a QGIS plugin (`https://plugins.qgis.org/plugins/netcdfbrowser/`) like the one we would like to develop for visualising multi-temporal raster data coming from "normal" satellite sources (see section 4.5.1 on page 39).

2.3.2 *GRIB (GRIdded Binary or General Regularly-distributed Information in Binary form)*

Similar to NetCDF, GRIB files are commonly used in meteorology to store historical and forecast weather data. It's a multidimensional file with the advantages of self-description, flexibility and expandability. GRIdded Binary or General Regularly-distributed Information in Binary form (GRIB) is standardized by the World Meteorological Organization's Commission and in operation since 1985. Currently, there are three versions of GRIB files (GRIB 0, 1 and 2). There are tools to convert GRIB into rasters such as grb2grid (`http://www.cpc.ncep.noaa.gov/products/wesley/grb2grid.html`) and QGIS software.

2.3.3 *HDF (Hierarchical Data Format)*

Hierarchical Data Format (HDF) was designed by the National Center for Supercomputing Applications (NCSA) to manage extremely large and complex scientific data. It is a versatile data model with no limit on the number of size of data objects in the collection. Its file format is designed to efficiently store and organize large amounts of numerical data, including satellite images. It is already used for a number of remote sensing data sources (e.g., Meteosat Second Generation, SPOT-VEGETATION and Proba-V provided through the Copernicus program) and will likely be adopted by other sensors (see section 4.5.1 on page 39). ArcGIS for example is capable of reading HDF4 and HDF5 data. The free open source Geospatial Data Abstraction Library (GDAL) (command-line) tools supports the conversion of HDF files to GeoTIFF.

2.4 examples for multi-temporal analysis

Multi temporal information is used for change detection, but it also provides a good tool to take phenological information into account when doing vegetation classification. In the following I am describing some practical usage of investigating time series data with GIS. There cannot be any prediction for future development without a look back into the past and an analyses of some time series data.

2.4.1 *Climate Change*

Many studies deal with the effects on society of the so called "greenhouse effect" climatic changes and a lot of effort is done to look at some climate-related time series to see if the effects of any such change can be identified. Series of lake levels and river flows are analysed for linear trends, periodicities, autoregressions and random residuals and possible physical causes are given for those components found to be significant. As climate change is a very long term process to investigate, it is even more important to be able to look at time series data to get a deeper understanding of the effects of climatic changes in the long term.

2.4.2 *Landcover Change*

Land-use and land-cover changes affect local, regional and global climate processes. Choices about land-use and land-cover patterns have affected and will continue to affect our vulnerability to the effects of climate change. Mapping land-use and land-cover change (LULCC) over large areas at regular time intervals is a key requisite to improve our understanding of dynamic land systems.

2.4.3 *Damaged forest*

Insect damage is a general problem that disturbs the growth of forests, causing economic loss and affecting carbon sequestration. Coarse-resolution data from satellites are potentially useful for national and regional mapping of forest damage, but the accuracy of these methods has not yet been fully examined. With time series and specific footprints on forests one can see if there is a potential risk for forest areas to insect damages.

2.5 SOFTWARE FOR IMPLEMENTING AN INTERACTIVE ANALYSIS

As this thesis relies on FOSS and aims at producing an open source tool for analysis we have to take a look at the possibilities of software available in order to make a decision for a OS platform and a programming language.

2.5.1 *QGIS*

QGIS is a user friendly Free and Open Source Geographic Information System (GIS) licensed under the GNU General Public License. QGIS is an official project of the Open Source Geospatial Foundation (OSGeo). It runs on Linux, Unix, Mac OS X, Windows. *BSD as well as Android and supports numerous vector, raster, and database formats and functionalities.

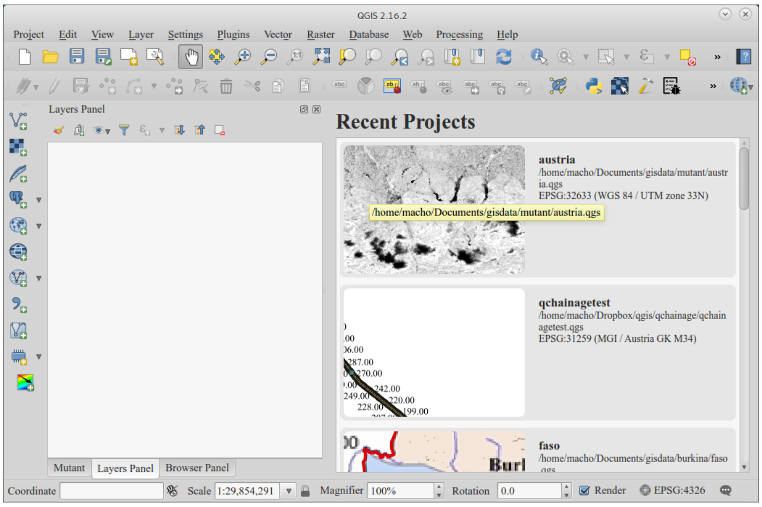

Figure 2.7: The user interface of a freshly started QGIS Desktop

Initiated in the year 2002 by Garry Sherman QGIS has since been growing every year and has currently reached a status where one can say it will not disappear anytime soon. QGIS also has a growing community and very good documentation. Having started as a viewer for PostGIS (`http://www.postgis.net/`) data QGIS has evolved into a fully functional and expandable GIS software which is fully FOSS compliant

2.5.2 *GRASS*

Geographic Resources Analysis Support System (GRASS) GIS is a free and open source Geographic Information System (GIS) software suite used for geospatial data management and analysis, image processing, graphics and maps production, spatial modeling, and visualization. GRASS GIS is currently used in academic and commercial settings around the world, as well as by many governmental agencies and environmental consulting companies.

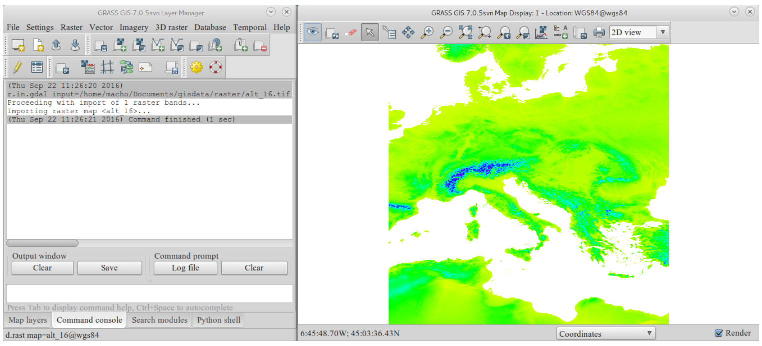

Figure 2.8: The user interface of a GRASS Desktop

It is a founding member of the Open Source Geospatial Foundation (OSGeo). Though GRASS is also scriptable with python scripts the functionality and integration with the GIS software itself (by creating a Graphical User Interface (GUI) is better solved in QGIS).

2.5.3 *Python*

Due to QGIS ability to be extended by plugins written in Python it is very likely that the software of choice for implementing the functionality will be Python. Even a lot of other big players in the GIS software industry are supporting python as a language to easily extend the functionality of their software (for e.g. Environmental Systems Research Institut (ESRI), GRASS). Not only because there is a Python Application Programming Interface (API) available for QGIS but also it is easy to learn, easy to read and general the recommended way of starting a plugin one should learn to program in Python. ArcGIS from ESRI have ceased their Visual Basic for Applications (VBA) implementation to also support Python in their GIS Desktop software.

2.5.4 *Combination of QGIS and Python*

As mentioned in 2.5.3 on the previous page QGIS is the perfect tool for the implementation of the ideas of this thesis. It offers a Python API and the community support is great. There is help readily availble in either reading the documentation or using one of the communication channels with its developers. Furthermore there are already some plugins available which integrate parts of the functionality I would like to achieve.

In the idea of FOSS I will not reinvent the wheel (as you can read in the cite by Linus Torvalds I used at the beginning of this thesis), but rather use already available parts and add the functionality I need to the already available software to hopefully solve the tasks this thesis tackles. In chapter 3 on the following page I will give a brief overview of the already existing parts and plugins for this type of workflow.

3 RELATED WORK

Research has shown that there are already software packages available which analyse timeseries. The available software with algorithms to investigate the phenology of NDVI data is already listed in [10]

There are also some currently available QGIS plugins with multi-temporal capabilities. Some QGIS plugins are known to address the work with multi-temporal raster data.

1. Value Tool [1]
   - Data access local only
   - Basic capabilities
   - development abandoned, not maintained anymore
2. QGIS Time Series [2]
   - Data access on remote server only
   - Not working on QGIS later than 1.8
3. Multi-temporal Analyser [3]
   - Not suited for many files as the date has to be manually assigned.
   - Could not be tested because of this limitation.
4. Temporal/Spectral Profile Tool [4]
   - Current status 'broken'. Could not be tested.

As the development for our own tool has already started we have to complete this list with our own tool here:

- MuTAnT (forked from Value Tool 0.8.5) [5]
  - Data access local only

1 https://plugins.qgis.org/plugins/valuetool/
2 http://wamis.meraka.org.za/time-series-viewer/qgis-time-series-plugin
3 https://plugins.qgis.org/plugins/multitemporalAnalizer/
4 https://plugins.qgis.org/plugins/temporalprofiletool/
5 https://plugins.qgis.org/plugins/mutant/

– Dedicated Time Series capabilities

– Development stage, but stable and useable

3.1 KNOWN ANALYSIS SOFTWARE

In this section I will concentrate on some of the already available software packages (see table 3.1 on page 32) and describe the range of their application spectrum. The most important information is already summarised in the table by [10].

Unfortunately most of the existing software that is capable of what we would like to achieve is free, but not in terms of FOSS. Furthermore there is not a single software out there that lets someone interactively do the analysis, so this part of being interactive would be one of the main goals to tom be archieved. Another goal is to be expandable due to integrate more algorithms for further investigations later (or even better let others implement more algorithms).

3.1.1 *TIMESAT*

TIMESAT (TIMESAT) is a Software Package for Time-Series Processing and Assessment of Vegetation Dynamics which is one of the main purposes the plugin has to deal with. The TIMESAT package consists of routines developed in Matlab and Fortran. It has been developed under Windows and tested also under Linux. TIMESAT is a free software package for processing satellite time-series data in order to investigate problems related to global change and monitoring of vegetation resources. The assumptions behind TIMESAT are that the sensor data represent the seasonal vegetation signal in a meaningful way, and that the underlying vegetation variation is smooth. A number of processing steps are taken to transform the noisy signals into smooth seasonal curves, including fitting asymmetric Gaussian or double logistic functions, or smoothing the data using a modified Savitzky-Golay filter (see 2.1.4.4 on page 14).

The methods incorporate qualitative information on cloud contamination from ancillary datasets. The resulting smooth curves are used for extracting seasonal parameters related to the growing seasons. The methods are implemented in a computer program, TIMESAT, and applied to NASA/NOAA Pathfinder AVHRR Land Normalized Difference Vegetation Index data over Africa, giving spatially coherent images of seasonal parameters such as beginnings and ends of growing seasons, seasonally integrated NDVI and seasonal amplitudes. Based on general principles, the TIMESAT program can be used also for other types of satellite-derived time-series data [9].

TIMESAT can adapt to the upper envelope of the data, accounting for negatively biased noise, and can take missing data and quality flags into account. The software enables the extraction of seasonality parameters, like the beginning and end of the growing season, its length, integrated values, etc.

TIMESAT has been used in a large number of applied studies for phenology parameter extraction, data smoothing, and general data quality improvement. To enable efficient analysis of future Earth Observation data sets, developments of TIMESAT are directed towards processing of high-spatial resolution data from e.g. Landsat and Sentinel-2, and use of spatio-temporal data processing methods. [4]

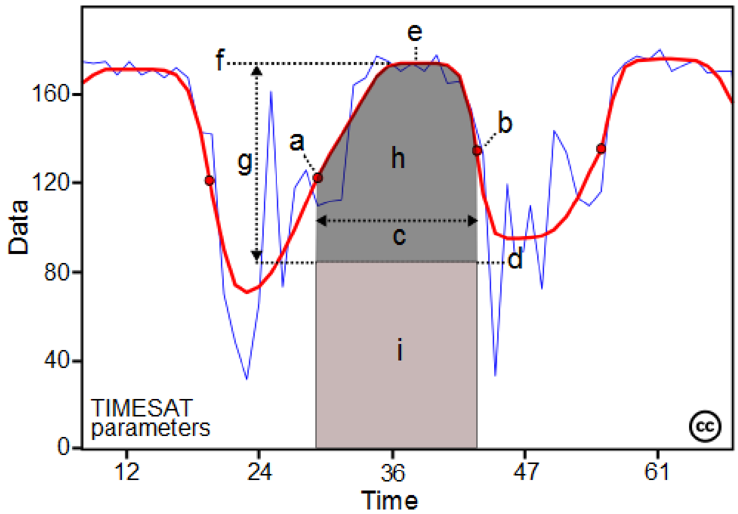

Figure 3.1: Some of the seasonality parameters generated in TIMESAT: (a) beginning of season, (b) end of season, (c) length of season, (d) base value, (e) time of middle of season, (f) maximum value, (g) amplitude, (h) small integrated value, (h+i) large integrated value.

Some of the main points to remark on TIMESAT software:

- Highly focused on vegetation data
- Available (only) for Windows and Linux

- Freely available but only for non-commercial academic research
- Not really open source software (in terms of FOSS) due to some restrictions in their distribution policy
- Lots of algorithms applied (by using matlab and Fortran) but there is no interactivity available

TIMESAT is available for download at
http://web.nateko.lu.se/timesat/timesat.asp

3.1.2 *BFAST*

Breaks For Additive Season and Trend (BFAST) integrates the decomposition of time series into trend, season, and remainder components with methods for detecting and characterizing change within time series.

BFAST iteratively estimates the time and number of abrupt changes within time series, and characterizes change by its magnitude and direction.

BFAST can be used to analyse different types of time series (e.g. Landsat, MODIS) and can be applied to other disciplines dealing with seasonal or non-seasonal time series, such as hydrology, climatology, and econometrics. The algorithm can be extended to label detected changes with information on the parameters of the fitted piecewise linear models.

BFAST monitor provides functionality for monitoring disturbances in time series models (with trend/season/regressor terms) at the end of time series (i.e., in near real-time). Based on a model for stable historical behaviour abnormal changes within newly acquired data can be detected. Different models are available for modeling the stable historical behavior. A season-trend model (with harmonic seasonal pattern) is used as a default in the regresssion modelling. You can find a sample output of BFAST in figure 3.2 on the next page.

Some of the main points to remark on BFAST software:

- One of the few programs being available under a free license (GNU GPL)
- Module for the R language and environment for statistical computing (R) programming language
- Not interactive on the input data

BFAST is available for download at http://r-forge.r-project.org/projects/bfast/

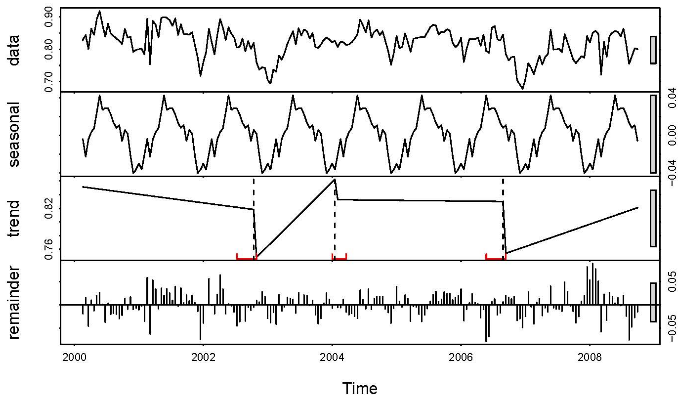

Figure 3.2: BFAST sample output

3.1.3 *TimeStats*

TimeStats is a free tool for the analysis of multi-temporal equidistant georeferenced remote sensing data archives, such as MODIS, AVHRR, MERIS and SPOT-Vegetation. Key features include parametric and non-parametric methods for trend detection, generalized-least square regression, distributed lag models, cross spectra analysis, windowed trend and frequency analysis, continuous wavelet transform, empirical mode decomposition and extraction of phenological indexes (peaking times and magnitudes). TimeStats is programmed in the Interactive Data Language®(IDL) and freely distributed with the IDL virtual machine®. Generated raster output files are saved in the standard ENVI®format with appropriate header files and are portable to common geospatial satellite imaging processing software packages. Software binaries and an extended user manual can be obtained from the author.

3.1.4 *PhenoSat*

PhenoSat is a software tool that allows to extract satellite VI metrics related to vegetation phenology in a simple and easy way. PhenoSat receives as input the yearly vegetation index images to process, and outputs the data processing steps and the phenological information. PhenoSat extracts information of seven phenological stages for the main growing season. It is also possible to record information for a double growth season or regrowth occurrence. This option allows obtain-ing information about the start and maximum of this period. Adverse weather conditions such as unseasonal snow, extreme heat or irregular precipitation could result in a false vegetation regrowth. The selection of a sub-interval, based on vegetation dynamics knowledge, could help to deal with the false report of regrowth in particular natural environments (high latitudes or boreal regions), leading to a better analysis and more consistent results. For the previous reasons, a new feature was added to PhenoSat that allows the possibility to select an annual time-series sub-interval. This selection reduces the volume of data to be processed, improves the fitting process and leads to more reliable results.

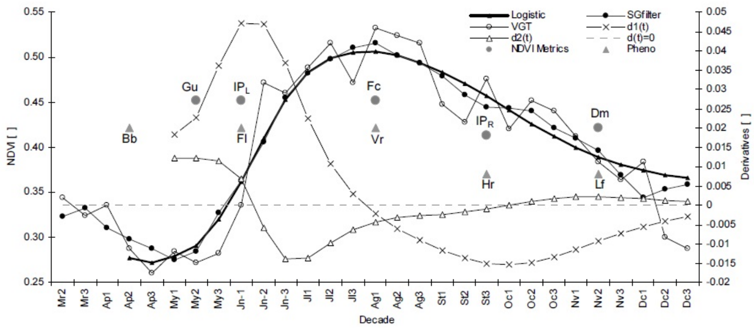

Figure 3.3: PhenoSat sample output

Different experiments have already been conducted with PhenoSat, using the vineyard, semi-natural meadows and low shrublands vegetation types. In all experiments, PhenoSat presented accurate and consistent results, compared with ground-based measures. This tool proved capable of solving some limitations present in other software tools, such as: the detection of a double growth season, with the extraction of phenological parameters for this period; and the possibility to select an in-season region of interest. This last feature proved to be a valuable tool for vineyard monitoring and can enlarge the PhenoSat application to crops with discontinuous canopy, like forestry and deciduous fruit trees. PhenoSat has been developed in Matlab and it is available for free.

PhenoSat is available for download at
http://www.fc.up.pt/PhenoSat/software.html

3.1.5 *SPIRITS*

The Software for the Processing and Interpretation of Remotely sensed Image Time Series (SPIRITS) is an integrated and flexible free software environment for analysing satellite derived image time series in crop and vegetation monitoring.

The software was originally developed as a toolbox for crop monitoring with remote sensing, but has evolved into an independent tool for the processing and analysis of time series of raster data. It can be used to perform and to automatize many spatial and temporal processing steps on time series and to extract spatially aggregated statistics. Vegetation indices and their anomalies can be rapidly mapped and statistics can be plotted and interpreted in seasonal graphs to be shared with analysts and decision makers.

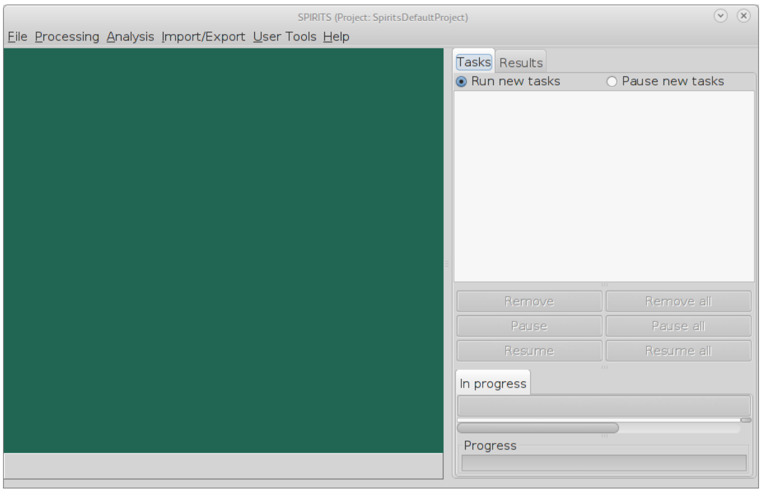

Figure 3.4: SPIRITS Desktop

SPIRITS has been developed to answer the specific needs of the agriculture monitoring community, with the objective to support the whole chain of image time series processing steps and the production of synoptic visual analysis outputs in a user-friendly and flexible manner. Since its first release in 2012 the software has evolved into an advanced and comprehensive time series processing tool.

SPIRITS is written in java programming language but incorporates a lot of other software (which includes using some FOSS as well) (see 3.4).

Additional improvements are under development. A major direction of development will also be the integration of SPIRITS in larger information systems, for example using PostgreSQL as reference database.

SPIRITS can be downloaded at:
http://spirits.jrc.ec.europa.eu/

3.2 PROBLEMS WITH EXISTING SOFTWARE

As already described in chapter 1, the main reason for doing this work was the lack of already existing software to do **interactive raster data analysis**.

The lack of an established standard poses another problem. Extracting information of creation dates from raster data is a hard task because every raster format either has no date somewhere or is hiding it in some uncommon places like images from satellite data (see 4.5.1). This problem is awaiting a solution.

Beside the fact that with different type of satellite data the conventions must be further investigated to come to a common solution which fits everything and in the best terms of FOSS will also fit for future solutions.

3.2.1 *Multi-temporal analysis*

The development of effective methodologies for the analysis of multi-temporal data is one of the most important and challenging issues that the remote sensing community will face in the coming years. Its importance and timeliness are directly related to the ever-increasing quantity of multi-temporal data provided by the numerous remote sensing satellites that orbit our planet. The synergistic use of multi-temporal remote sensing data and advanced analysis methodologies results in the possibility of solving complex problems related to the monitoring of the Earth's surface and atmosphere at different scales. However, the advances in the methodologies for the analysis of multi-temporal data have been significantly under-illuminated with respect to other remote sensing data analysis topics. In addition, the link between the end-users' needs and the scientific community needs to be strengthened [25]

To be able to do a multi-temporal analysis one either has to have a "date" field or some other knowledge about the date when the data was taken or produced.

Even though this thesis focuses on satellite images the plugin must still be open to all different kind of date sources to be able to be used as widely as possible for all different kind of tasks. One has to search for a good way to read the date and time when the raster data was taken or produced (see section 4.4 on page 38).

As one can see there are already a lot of existing solutions which all give proper analysis results but mostly lack the interactive part one would like to have.

It is always the best solution to take the best parts of existing solutions and combine them in a way that the fitted parts together give a better user experience than the former separated parts. The advantages of all investigated available solutions should be combined.

The result of this thesis as a QGIS plugin will be based on the already existing Valuetool plugin and, due to the lack of interactivity of all existing solutions will focus on interactivity. In this way everything should be under the hood of one program and there will be no need to change the GIS program for analysis or switch between different programs to search for a coordinate and investigate it in another program.

Table 3.1: Available software comparision for analysis method specifically to phenology taken from [10] and extended with more software

TOOL	PURPOSE	METHOD	SOURCE CODE	REFERENCE
TIMESAT	Smoothing, LSP extraction	Savitzky-Golay, Gaussian, double-logistic, Relative threshold	FORTRAN90 MATLAB free	[8]
BagS	Smoothing, LSP extraction	"4253H smoother, twice", Gaussian model, detecting and isolating peaks and mins on first numerical derivative	C++ free	[15]
TiSeG	Smoothing	Stepwise interpolation	IDL free on request	[3]
TSPT/PPET	Smoothing, LSP extraction	Interpolation, Savitzky-Golay, Relative threshold	MATLAB -	[19], [18], [23]
BFAST	Trends and seasonal analysis	-	R GPL	[27]
Ndvits	Smoothing, LSP extraction, TS analysis	Savitzky-Golay, Absolute threshold, BFAST	R GPL (not maintained since 2014)	[1]
TimeStats	Smoothing, TS analysis	Savitzky-Golay	IDL free on request	[26]
PhenoSat	Smoothing, LSP extraction	Double-logistic, Savitzky-Golay, 1st derivative of curvature	Matlab free (code protected)	[22]
SPIRITS	Smoothing, LSP extraction	MEAN, BISE, SWETS, Whitthaker	free to use (but code protected)	[21]

4 METHODOLOGY

Based on scientific research methods explained in chapter 2 on page 6, the chapter methodology will give an overview about the idea and the history, explains the research choices that have been done as well as the approach to the different projects used.

4.1 THE IDEA

The basic idea is to be able to interactively get a table or plot of all values of all loaded raster files at a interactively specified coordinate, pointed out by the mouse pointer. Furthermore this coordinate must be selectable with the mouse cursor and it must be possible to extract the creation date of the raster data. Afterwards all loaded raster data can be put into a correct time frame, which makes it possible to see the development of the values at a selected coordinate over time.

The whole idea started at the Institute of Surveying, Remote Sensing and Land Information (IVFL) (`https://www.rali.boku.ac.at/en/ivfl/`) while doing a project for surveying and testing different software for analysis. As one can see in chapter 3 on page 23 there is a lot of software available for analysis and get some good graphs with algorithms on a known coordinate but scientists still have to search for the correct coordinates if they do not know them yet.

What we want to archive is a tool for analysis with which one can search for the correct coordinate at the same time while applying some algorithms. The tool obviously has to be interactive and give a good feedback about what is happening at a specified coordinate point.

At BOKU they always had to search for a specific NDVI graph to identify a poppy seed field in Afghanistan. So the main focus here will be investigation of NDVI indices.

Of course the project still wants to be open for other scientific approaches as well and wants to give the software more the type of a general tool to look at (mainly) satellite data.

Some of the possible usages of the tool you can find in section 2.4 on page 19.

With a good GUI this plugin will be possibly the only tool you need for investigation and applying scientific algorithms to satellite raster data. The plugin will be developed with being extensibility in mind and due to the nature of being written in the python (see section 2.5.3 on page 21) programming language it will also be easy for people to do so.

4.2 HISTORY OF DEVELOPMENT

After searching and investigating and having a look at some of the tools, it soon turned out that at the time of writing this (September 30, 2016) there is no available tool with the possibility to do an interactive analysis on satellite raster data. Speaking from interactivity there is already a tool available as a QGIS plugin that is able to display values of raster data interactively: Valuetool (see chapter 3 on page 23). So in the sense of FOSS I took this tool and tried to understand the python code in it. After realising that this tool (named Valuetool) is not actively developed any more and more or less unmaintained I decided to take it and maintain it for the future and add my own wishes to what was already available. Valuetool will be the base for our tool but we have to enhance it a lot to get everything into the code we want to have. Most important thing will be to extend the base with the ability for time series analysis.

4.3 USED SOFTWARE

The big advantage of FOSS software is its possibility to get the source code and adjust it to your needs. Moreover a lot of tools are developed open source that you can just take and use to enhance it in the way you want or to create completely new tools out of if if you combine it in a good way.

The software used to develop QGIS and as a result the plugin is based on is listed here:

- Qt
- Qwt
- Python
- matplot
- PyQtGraph

4.3.1 *QGIS*

QGIS is a Geographic Information System (GIS) that manages, analyses, and displays databases of geographic information. QGIS supports shape file viewing and editing, spatial data storage with PostgreSQL/-PostGIS, projection on-the-fly, map composition, and a number of other features via a plugin interface. QGIS also supports display of various georeferenced raster and Digital Elevation Model (DEM) formats including GeoTIFF, Arc/Info ASCII Grid, and USGS ASCII DEM.

QGIS is the perfect choice for our task. It can read a lot of different GIS raster data and has a very good raster data management.

4.3.2 *Qt*

Qt is a cross-platform application framework that is widely used for developing application software that can be run on various software and hardware platforms with little or no change in the underlying codebase, while still being a native application with native capabilities and speed. Qt is currently being developed both by The Qt Company, a company listed on the Nasdaq Helsinki Stock Exchange and the Qt Project under open-source governance, involving individual developers and firms working to advance Qt. Qt is available with both commercial and open source GPL 2.0, GPL 3.0, and LGPL 3.0 licenses.[32]

Qt is also the GUI base and the overall grounding of QGIS

4.3.3 *Qwt*

The Qwt library contains GUI Components and utility classes which are primarily useful for programs with a technical background. Beside a framework for 2D plots it provides scales, sliders, dials, compasses, thermometers, wheels and knobs to control or display values, arrays, or ranges of type double.

Qwt is used for several enhanced graphic abilities in QGIS as well as the fastest library for plotting time/value pairs because it is written in C++. Unfortunately for our GIS purposes the future of using this library is very unclear because the bindings to the python programming language are not actively developed any more. For displaying values of GIS coordinates it is the most reliable and fastest library for usage with Qt.

4.3.4 *Python*

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Python and its bindings for Qt and QGIS are the base for successful plugin development. Without python easy plugin development and fast results would hardly be possible. So it is the ideal choice for newcomers or part time programmers to use and its already developed modules are easily installed and great to use especially if you want to have access for GIS purposes to specific algorithms (see sections 2.1.4.2 on page 12 and 2.1.4.4 on page 14).

4.3.5 *matplotlib*

matplotlib is a 2D plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB. SciPy makes use of matplotlib. matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell, web application servers, and six graphical user interface toolkits. matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code. For simple plotting the pyplot interface provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.[7]

For our purposes matplotlib is the perfect choice because it offers ways to deal with date/time data which is exactly the GIS data we want to extract out of raster data and analyse it on a multi-temporal base.

4.3.6 *PyQtGraph*

PyQtGraph is a graphics and user interface library for Python that provides functionality commonly required in engineering and science applications. Its primary goals are

1. to provide fast, interactive graphics for displaying data (plots, video, etc.) and

2. to provide tools to aid in rapid application development (for example, property trees such as used in Qt Designer).

PyQtGraph makes heavy use of the Qt GUI platform (via PyQt or PySide) for its high-performance graphics and uses the python module numpy for fast heavy number crunching. In particular, PyQtGraph uses Qt's GraphicsView framework which is a highly capable graphics system on its own. Despite being written entirely in python, the library is very fast due to its heavy leverage of numpy for number crunching and Qt's GraphicsView framework for fast display. PyQtGraph is distributed under the MIT open-source license and is known to run on Linux, Windows, and OSX PyQtGraph is a pretty fast python library and allows us to deal with bigger data.

There are a lot more FOSS software projects involved, included or linked with QGIS like GDAL, Proj4, Geometry Engine - Open Source (GEOS), SQLITE, Spatialite and of course bindings for python (PyQt). Describing each of this software is too much for this thesis but they should all at least be mentioned here.

4.3.7 *MODIS R*

MODIS R is a R software package which is written and maintained by Matteo Mattiuzzi from BOKU. The software itself is a additional package for R and provides functions for download, mosaic, re-sample, re-project, analyse and visualise of MODIS grid data. Special attention is given to spatio-temporal change detection and phenological metric extraction. This software is already used to download and prepare the raster data (downloaded from the MODIS project) software in R. [17]

4.4 searching for 'standards'

At the time of writing the plugin there is no standardized format available being able to somehow store time/date information inside a GIS raster data format.

In terms of technical possibilities there are some picture(raster)-formats that make it possible to store additional Metadata inside the file. This so called "EXIF-Data" (Extended Information) is widely used in digital cameras to store the GPS position where the picture was taken and of course the pictures time and date. Unfortunately georeferenced raster data does not use these possibility of using EXIF Data to store its information inside the data itself.

The common convention through nearly all Georaster data formats I could find was the form of storing the information about the Satellite, the time and date of the image and some more information inside the file name of the image.

In most cases it is obvious how the naming is done in the file name of the provided images. Nearly all of the data providers have web pages where they explain how the scheme of naming is done inside their satellite image products and where one can download the image data.

4.5 different data

As already explained in section 2.2 raster data can appear in a lot of different formats. Only a few of them have the ability to store some kind of date/time information (for e.g. GeoTIFF). As a tool for analysis should be usable for as many different cases as possible, we have to look for a common way to get date/time information out of every possible format coming along with raster data.

As we will mainly work with NDVI data provided by MODIS we have to look where MODIS comes from and where its recorded date of the picture is stored. If we process the satellite images before taking a deeper look at their multi-temporal data we can decide by ourself how the date is stored. It shows that taking the file name as the place for storing the date is still one of the better places because one can see by looking at the file when it was taken. For later usage of probably data from different sources with other storage places (even different inside the filename) it is better to use the Metadata to read everything regarding time and date from one standardized place.

4.5.1 *Satellite sources*

Large volumes of data from satellite sensors with high time-resolution exist today, e.g. Advanced Very High Resolution Radiometer (AVHRR) and MODIS, calling for efficient data processing methods. There are a lot of different satellite systems in the orbit. Some of them are military only, but there are also a lot available for civil research. These data is the most valuable resource for using it with the interactive tool.

The availability of satellite data is nowadays very good. A lot of data is made available for free (but you still have to take a look at the license if you want to use it or create some new commercial data out of it). In most cases you can download the data from the webpages of their respective data providers. Most of the free available satellite data uses the system of storing the date when the picture was taken inside the filename. For creating multi-temporal plots the first problem to solve is how to get the date from the filename.

To name some of the systems that provide data:

- Different Landsat systems (available since 1972)
- Meteosat (First (MFG) and second (MSG) Generation)
- MODIS
- SPOT
- SENTINEL

The longest history of available satellite data has the Landsat system launching its first satellite on july 23 1972. At that time it was named "Earth Resources Technology Satellite".

4.5.1.1 *Costs of satellite data*

A lot of these Satellite image data is available for free download. In some cases one has to register first at the webpage to be able to download the satellite data. To get the data you sometimes have to register at the webpage of the satellite data provider but this registration is usually also for free. To make it even easier there are already some existing plugins for QGIS to automate the process of downloading image data for a specific area (which in selectable)

Sometimes there are restrictions on downloading the data like the one the SPOT system has: Pricing Policy: Standard 10-day synthesis products older than three months are now available free of charge to all customers. Sometimes there are also special conditions for scientific users. In case of the SPOT program, and according to their webpage it is like this: "Scientific users and programme partners can obtain all S and P products upon payment of the cost of medium, packaging, and shipping. All requests however are subject to approval by the VEGETATION Programme and orders are processed according to availability of production capacity. P products older than 3 months are available on SDLT and DVD only. Other products can be downloaded via FTP."

In general we can say that a lot of data is available for free. For hostorical data you sometimes have to pay, but there are also exceptions where the data is available for downloading for free. For "real-time" data you often have to pay (but this is obviously aimed at providing weather forecast stations with data). For investigation of vegetation in the area of NDVI we usually do not need real time data.

But for historical investigations and creating a timeline for the last years real time data is usually not necessary and we can stick to just download the data from the webpages.

4.5.2 *MODIS*

MODIS Vegetation Index Products (NDVI and EVI)

MODIS vegetation indices, produced on 16-day intervals and at multiple spatial resolutions, provide consistent spatial and temporal comparisons of vegetation canopy greenness, a composite property of leaf area, chlorophyll and canopy structure. Two vegetation indices are derived from atmospherically-corrected reflectance in the red, near-infrared, and blue wavebands; the normalized difference vegetation index (NDVI), which provides continuity with NOAA's AVHRR NDVI time series record for historical and climate applications, and the enhanced vegetation index (EVI), which minimizes canopy-soil variations and improves sensitivity over dense vegetation conditions. The two products more effectively characterize the global range of vegetation states and processes. The vegetation indices are retrieved from daily, atmosphere-corrected, bidirectional surface reflectance. The VI's use a MODIS-specific compositing method based on product quality assurance metrics to remove low quality pixels. From the remaining good quality VI values, a constrained view angle approach then selects a pixel to represent the compositing period (from the two highest NDVI values it selects the pixel that is closest-to-nadir). Because the MODIS sensors aboard Terra and

Aqua satellites are identical, the VI algorithm generates each 16-day composite eight days apart (phased products) to permit a higher temporal resolution product by combining both data records. The MODIS VI product suite is now used successfully in all ecosystem, climate, and natural resources management studies and operational research as demonstrated by the ever increasing body of peer publications.

4.5.3 *Landsat*

The Landsat program is the longest-running enterprise for acquisition of satellite imagery of Earth. On July 23, 1972 the Earth Resources Technology Satellite was launched. This was eventually renamed to Landsat. The most recent, Landsat 8, was launched on February 11, 2013. The instruments on the Landsat satellites have acquired millions of images. The images, archived in the United States and at Landsat receiving stations around the world, are a unique resource for global change research and applications in agriculture, cartography, geology, forestry, regional planning, surveillance and education, and can be viewed through the USGS 'EarthExplorer' website (http://earthexplorer.usgs.gov/). Landsat 7 data has eight spectral bands with spatial resolutions ranging from 15 to 60 meters; the temporal resolution is 16 days. Landsat images are usually divided into scenes for easy downloading. Each Landsat scene is about 115 miles long and 115 miles wide (or 100 nautical miles long and 100 nautical miles wide, or 185 kilometers long and 185 kilometers wide) [30].

All Landsat scene identifiers are based on the following naming convention:

LXSPPPRRRYYYYDDDGSIVV

L=Landsat	PPP=WRS path	DDD=Julian day of year
X=Sensor	RRR=WRS row	GSI=Ground station identifier
S=Satellite	YYYY=Year	VV=Archive version number

Examples:

From the examples we can easily recognise that YYYYDDD is the part we would need for our date extraction.

Table 4.1: Landsat naming convention

NAME	SATELLITE
LC80390222013076EDC00	(Landsat 8 OLI and TIRS)
LO80390222013076EDC00	(Landsat 8 OLI only)
LT80390222013076EDC00	(Landsat 8 TIRS only)
LE70160392004262EDC02	(Landsat 7 ETM+)
LT40170361982320XXX08	(Landsat 4 TM)
LM10170391976031AAA01	(Landsat 1 MSS)

4.5.4 *SENTINEL2*

The full SENTINEL 2 mission comprises twin polar orbiting satellites in the same orbit, phased at 180° to each other. The mission monitors variability in land surface conditions, and its wide swath width and high revisit time (10 days at the equator with one satellite, and 5 days with 2 satellites under cloud free conditions which results in 2-3 days at mid latitudes) and will support monitoring of changes to vegetation within the growing season. The coverage limits are from between latitudes 56° south and 84° north.

The Sentinel 2 products follows the naming convention defined in figure 4.1 on the next page):
MMM_CCCC_FFFFDDDDDD_<Instance_ID>.<FORMAT>
A short overview about whats inside this naming convention can be found in table 4.2 on the following page. A brief description about the naming is available at `https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/naming-convention`

4.5.5 *Metadata*

As you can see in sections 4.5.3 on the previous page and 4.5.4 the naming of the satellite images are usually quite the same, but there are still differences between them, which obviously also depend from the type of data inside the image. If we want to have a common format for all images we have to "invent" one. Luckily we do not really have to invent the whole type of storage of the data. There are already some ways which are kind of standard, and we only have to use them and adjust

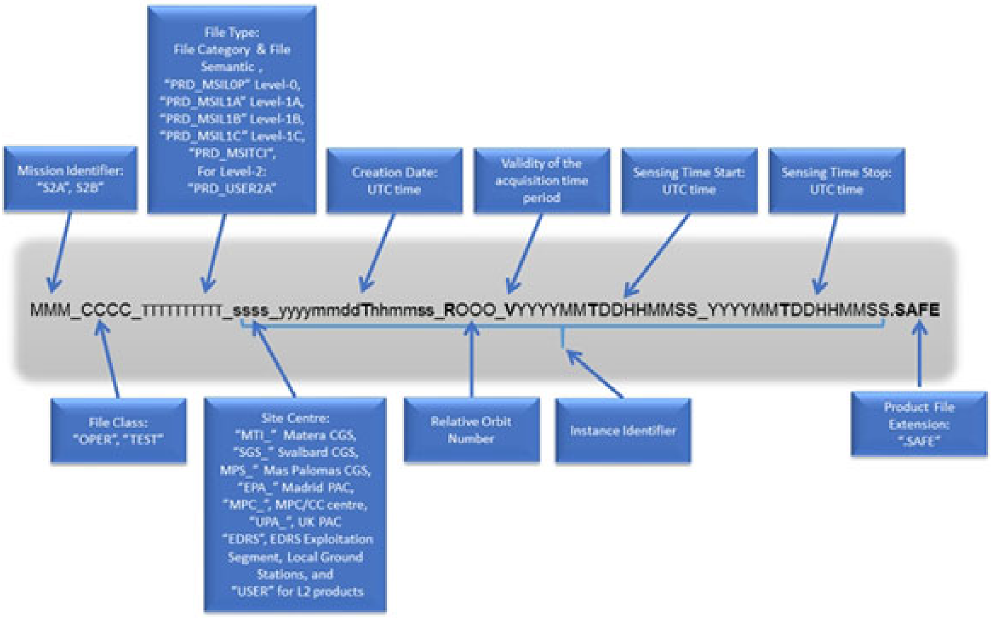

Figure 4.1: SENTINEL naming convention, graphic taken from https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/naming-convention

Table 4.2: SENTINEL2 naming convention

PART	DESCRIPTION	COMMENT
MMM	Mission Identifier	S2A; S2B
CCCC	File Class	OPER, TEST
TTTT	File Type (Category and Semantic)	PRD_USER2A
Instance_ID	contains yyyymmddThhmmss	
FORMAT	Product Format (SAFE)	

them for our way of storing date and time data which accompanies an image format.

4.5.5.1 *Metadata and GDAL*

GDAL can deal with the following baseline TIFF tags as dataset-level metadata :

TIFFTAG_DATETIME which is the date and time of image creation. The format is: "YYYY:MM:DD HH:MM:SS", with hours like those on a

24-hour clock, and one space character between the date and the time. The length of the string, including the terminating NUL, is 20 bytes.

Unfortunately the satellite images available for download usually do not use this Exchangeable Image File Format (EXIF) data so we have to search for another way to store the date permanently and make it available for all purposes.

On the other hand fortunately other non standard metadata items can be stored in a TIFF file created with the profile GDALGeoTIFF. Those metadata items are grouped together into a XML string stored in the non standard TIFFTAG_GDAL_METADATA ASCII tag. When BASELINE or GeoTIFF profile are used, those non standard metadata items are stored into a Persitant Auxiliary Metadata (PAM) .aux.xml file.

Starting with GDAL version 1.9.0, Extensible Metadata Platform (XMP) metadata can be extracted from the file, and will be stored as Extensible Markup Language (XML) raw content in the xml:XMP metadata domain.

Starting with GDAL version 1.10, EXIF metadata can be extracted from the file, and can be stored in the EXIF metadata domain.

As we have seen in section 4.4 on page 38 there is no standard for storing date and time of creation of an image. Merely we have to say that there are more standards which incorporate storing it in some metadata inside the file, storing it in metadata outside the file or use the filename to store this kind of data.

We have several possibilities to read and store. To provide a tool for all purposes, which is still our intention, we have to create an approach that deals with all of this storage possibilities. But after all we need one place to read from and this has to be integrated into the plugin too.

5 RESULTS

This thesis lead to the programming of the MuTAnT plugin which itself is derived from the abandoned Valuetool plugin for QGIS.

The result of the development efforts can be found at either `http://plugins.qgis.org/plugins/mutant/` or `https://github.com/mach0/mutant` and supersedes the Valuetool plugin and will be maintained by me also in the future.

A brief description of the tool and its usage and installation can be found in the appendix of this thesis (see appendix .1 on page 63. This is kept like this to easily split the manual off of the thesis and use it on its own.

At this point as a result of the developing efforts I also wanted to know where the plugin is already in usage and sent some mails to people known to me where there is possible use for the plugin in the real GIS world. You can find one answer in the appendix (see .3 on page 79) which proof that the tool is used but that there is also a lot work ahead to get FOSS GIS tools at the same level and acceptance than commercial product.

Unfortunately the most important part of the thesis - the interactivity - can hardly be shown in a written thesis. As one moves the mouse cursor over the loaded satellite raster data you can see the change. Depending on which view of the MuTAnT plugin you choose either really fast columns of digits which can be exported to Comma-Separated Values (CSV) or as lines in a nicely shaped graph. More information about what to enable or disable in the plugin can be found in the Appendix (see appendix .1 on page 63)

5.1 PROBLEMS WHILE WRITING

During writing the plugin and finding the best way of doing multi-temporal analysis on satellite raster data some problems appeared which have to be solved as part of writing the thesis. This problems are listed in the following sections.

5.1.1 *Extracting date and time*

At the time of writing there was no existing standard for reading and storing time/date data inside or outside satellite images (see section 4.4 on page 38). We need to extract the time/date data from the images to create a time line as X-axe and to apply algorithms.

The approach used to combine possibly all sources of data into one place is:

1. Read the time/date from every possible place.
2. Convert it into a format which is usable for our multi-temporal analysis
3. Store it into one place as an XMP file next to the original image as a metadata file (.aux.xml).

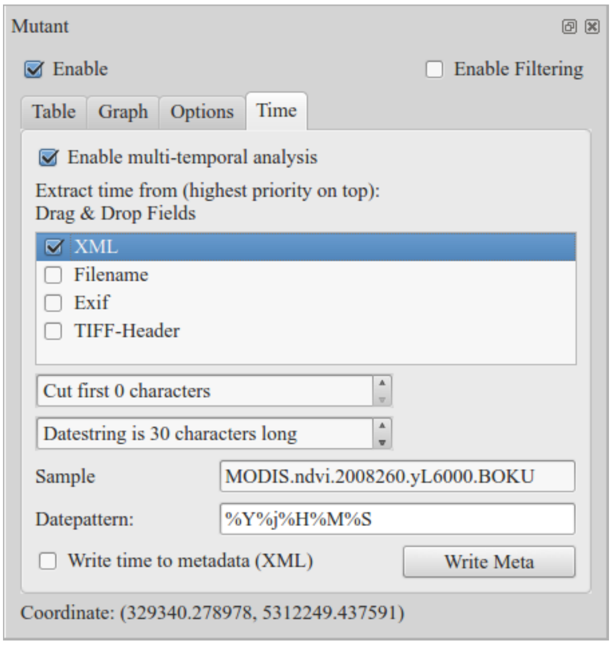

Figure 5.1: Multi-temporal Options inside MuTAnT

As one can see there are several possibilities to read the time or date when the image was taken from (XML, Filename, EXIF, TIFF-Header). Not all are yet implemented in the plugin, but the plugin is prepared to be easily extended to be able to read from all this sources. To the authors knowledge all possible places where time/date data can be stored have been included in the plugin. At least an easy extendable preparation of the plugin is prepared to include possibly missing places to get time/data data from.

For our purposes of NDVI investigations from MODIS data the biggest part to solve was to find some useable standard (see section 4.4 on page 38) therefore we can now read from different sources and store them all at the same level (XML) as one can see in the left bottom corner.

5.1.2 *Additional TAB for interactive timeseries analysis*

As the MuTAnT plugin is based on the Valuetool plugin there was some time to invest on the existing code and to rewrite parts of it to get better integration of the timeseries. To access the time based data we also needed to extend the GUI by adding a new TAB to enable this timebased analysis. We needed to do this to still provide the usual way of analysis for people who need that kind of "normal" type to investigate, but also enable the time series analysers to get the tool fully functional for them.

The newly added Time TAB (see figure 5.1 on the previous page) does not only enable the new type of multi-temporal analysis but also includes functionality to extract the time/date from filenames and write this time/date information to a metadata file (see section 5.1.1 on the preceding page).

5.1.3 *Graphics Libraries*

There were several possible graphic libraries available to choose from. All had their advantages and disadvantages for displaying the multi-temporal data. In figure 5.2 on the next page you can see the available graphical libraries you can select from.

A short description of the libraries and its pros and cons for GIS analysis you can find in sections 4.3.3 on page 35, 4.3.5 on page 36 and 4.3.6 on page 37.

An ideal situation would be to have to speed of Qwt (which is written in C++), the functionality of matplotlib (which is unfortunately very slow with big data) and the easyness of integration of PyQtGraph (because it is written in Python and also has some speed).

PyQtGraph would be the ideal library but is also lacking the functionality of time and date which i implemented by myself. But after implementing it, I saw that PyQtGraph ceased its development in favour of a new library called VisPy (`http://vispy.org/` which will probably be used when I start to rework and rewrite big parts of MuTAnT.

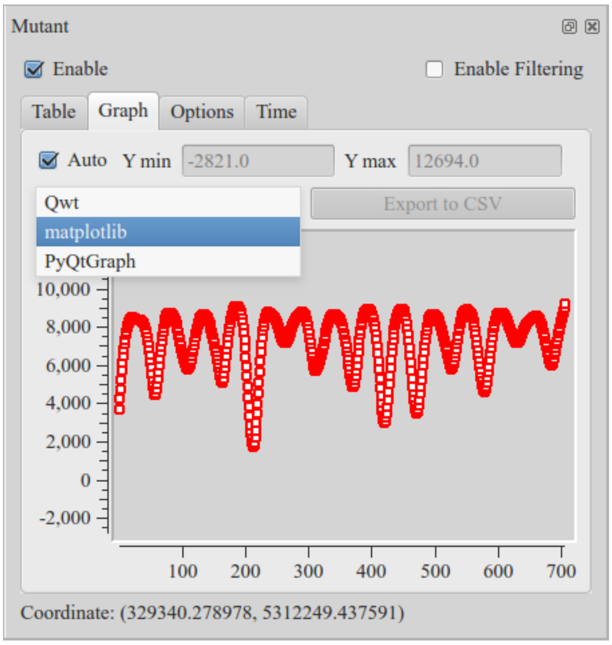

Figure 5.2: Different Graph libraries available in MuTAnT

5.2 CAPABILITES OF MUTANT

Now lets take a look at what we have achieved so far: For deeper insight about the functionality take a look at the appendix .1 on page 63.

- Added date and time for multi-temporal analysis
- Visualising of multi-temporal data with 3 different graphic libraries
- Interactive visualising by hovering over the canvas with the mouse cursor
- Switch-able interactivity via the options menu
- Display analysis data not only as a graph but also as a data sheet
- Export data sheet including the point coordinates where the data comes from as CSV
- Enables to add algorithms for further data analysis and displaying as a graph.

By looking at this we created a whole processing chain for interactive multi-temporal satellite raster image. We solved the problem of interactivity, which was the main part, because at the time of starting this thesis there was no tool available to do this interactively. And all this was done using FOSS and publish our developed tool also as FOSS under the license of the GNU GPL. This all makes it possible for others to jump in and help with development, add their own analysis tools and use them interactive on raster data. And as one can see, as the tool already got published during development, there are people out there already using it on a daily base in real projects (see 5.3).

5.3 CASE STUDIES

As the plugin was initiated by DI Matteo Matiuzzi and Dr. Anja Klisch at the University of Natural Resources and Life Sciences in Vienna (BOKU), they both worked a lot with the created software already during its development time. Furthermore, as the software is open source and available within an open source GIS program it has been used by the United Nations Organization (UNO) to help them identifying poppy seed fields in Afghanistan. And obviously the software has also helped the anti-drug special forces in Mexico to identify also drug fields by searching for the characteristic NDVI graph with MuTAnT.

Unfortunately, and this is a bit sad to mention, people tend to forget about what they are getting for free. So after I asked for case studies especially at the UN I got an email back which you can find in the Appendix at .3 on page 79.

So probably there are lots of case studies and usages of the plugin, but due to its nature of open source I will probably never find out where it was or where it still is being used in different kind of usage possibilities.

6 CONCLUSION AND DISCUSSION

It is doable to create a FOSS plugin for QGIS to interactively analyse multi-temporal raster data. Though the plugin took some time to write I see many possibilities to apply this plugin in real world use cases. Some can already be found in 5.3 on the preceding page.

It is possible to do all the necessary tasks with FOSS and the first steps were taken towards a workflow that is fully capable of doing interactive analysis on raster data. Some certain amount of time still has to be put into developing the analysis capabilities and apply them to the data arrays. The developed software itself is already published inside the main QGIS plugin repository (`http://plugins.qgis.org`) and therefore available for everyone to download.

Since the availability in the repository and the demand from the BOKU it has already been downloaded several thousand times being used as the tool for detecting poppy seed fields in Afghanistan by the United Nations and helping in drug detecting with the Mexican special forces for drug prevention.

6.1 THE FUTURE

Software is never good enough to be "ready". Looking at what has been done up to the current status as of September 30, 2016(see section 5.2 on page 48), there is already a lot work done, but during doing the work one is usually blind for immediate enhancements and looking at the "finished" work one can always see room for improvement.

6.1.1 *List of topics and potential improvements:*

I stopped programming the plugin some time ago to finish the thesis, otherwise I think I would have never finished. But I can already see some a lot of things that should be improved and rewritten to better fit into GIS analysis. First of all there should be a refactoring and rewriting of the data handling where it is currently too difficult to apply more complicated algorithms and that has to be done much easier and implemented in other ways. While doing this the whole plugin can be updated to support the upcoming version 3.0 of QGIS which also supports python in version 3.

6.1.1.1 *Possible visual plot enhancements*

- The plot misses lines that are indicating the year.
  This is required for the orientation in time, to quickly see where a cycle is located and how many per year and how intense they are (see figure 6.1). This could be important for long time series and multiple cycles per year.

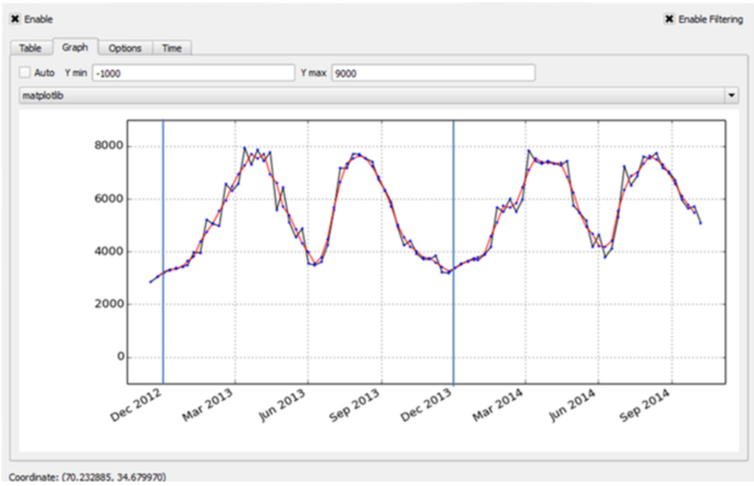

Figure 6.1: Line for year separation

- Add a set of filter the make the interpretation of noisy data easier (see figure 6.2)

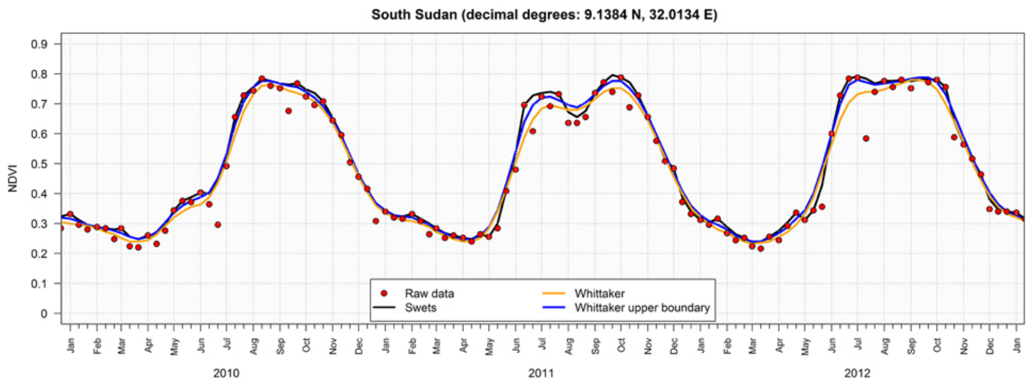

Figure 6.2: Adding more filter algorithms to the data

It could also be implemented to draw different years in a different colour or linestyle as an overlay, or even better not only years but do a custom separation on data (for e.g. daily, monthly, yearly).

6.1.1.2 *Possible functionality enhancements*

- Zoom to a specified period (x-axis) by pushing the cursor on the plot.
  This is already possible with using the PyQtGraph library but is highly dependend from the used library. As PyQtGraph will be replaced by PyVis this could be a possible improvement during implementation.
- Interact with the plot by drawing a vertical line on a specific date (e.g. date of NDVI maximum) and to keep this line as reference for the other locations. This could help to compare other locations have a similar phenological stage.
- The reproduction of the vertical line periodically (e.g. each 1st of May) over multiple years. This could help to understand the phenology between the years (see figure 6.3.

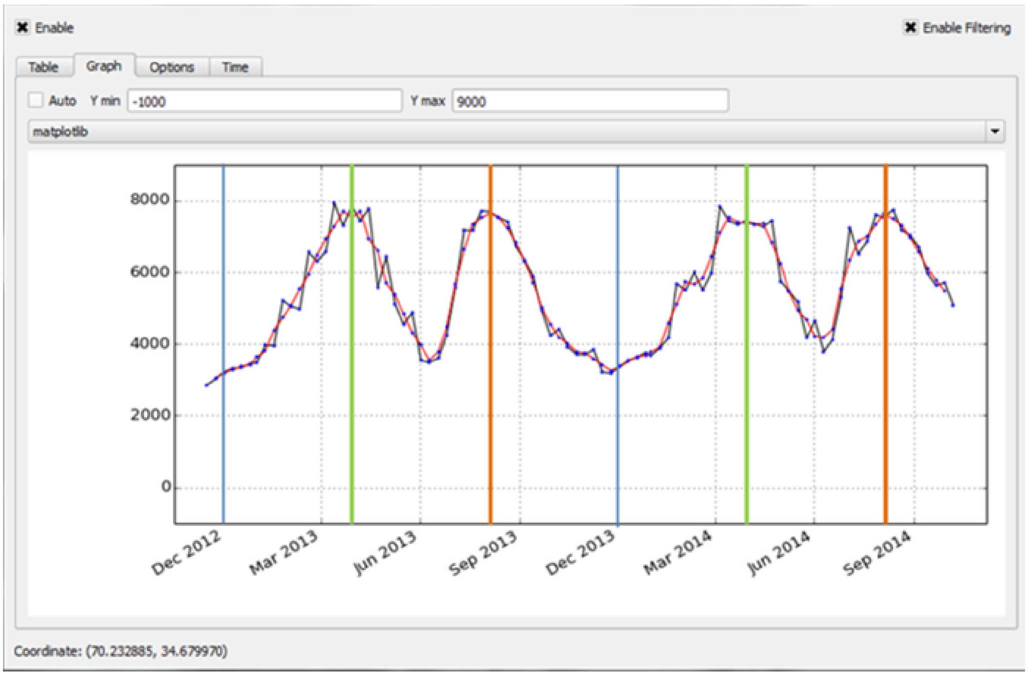

Figure 6.3: Periodically repeating lines to help the identification of crop periodicity over years

- By pointing on a location in the graph it could help to return date (x-axis) and NDVI value (y-axis) (see figure 6.4). This might also be possible to archieve by selecting the correct graphics library and will include some more work to enhance the visualisation.

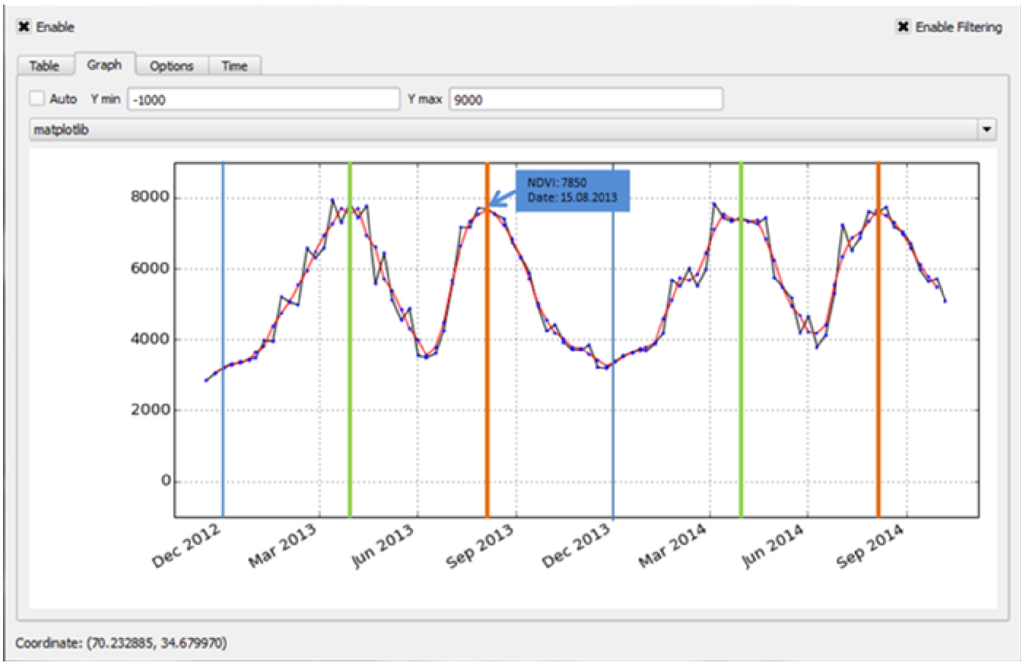

Figure 6.4: Interactively read NDVI and date on the in the graph pointed location.

- Plot multiple indicators (e.g. NDVI + moisture index, or multi-year statistics). This functionality can be adapted from the source code in 'QGIS Time Series Plugin' of Wide Area Monitoring Information System (WAMIS) (see figure 6.5 on the following page).

6.1.1.3 *Enhance dealing with big data*

Currently the time series raster data is simply loaded into QGIS from a folder as the regular files. Up to a certain number of pixel and layers it is easily handled by QGIS. Using MODIS (230m ground resolution) and not to many years of time series it is fairly easy to cover at least half of Afghanistan in one run. But considering the need of a higher resolution data like Landsat (30m) or the upcoming Sentinel2 (10m), the preprocessing and loading of data becomes a very challenging matter.

1 http://wamis.meraka.org.za/time-series-viewer/qgis-time-series-plugin

Figure 6.5: Plotting multiple indicators at once (screenshot from WAMIS [1])

To be able to cover large areas and periods with high resolution data (approximately 5-50m ground resolution) it is required to process and store the data in a dedicated environment. Usually such environment is a (local or remote) server interface that handles the huge dataset and returns the temporal information each time it is queried by a mouse click in the QGIS canvas. For this the 'QGIS Time Series Plugin' of WAMIS can probably be used as a base and example as it has the capability to interact with a remote server.

A second solution is finding a graphics library that can deal with big datasets and/or improve the data handling inside the plugin for faster applying of algorithms. This is one of the points that has to be done in a next step.

6.1.1.4 *Enhanced filtering/processing capabilities*

Generally speaking the filtering (noise reduction) is not so important for the visual interpretation of the graphic as only very few cloudy pixels are present. The current available filtering capabilities are just a prove of concept (see figure 6.6).

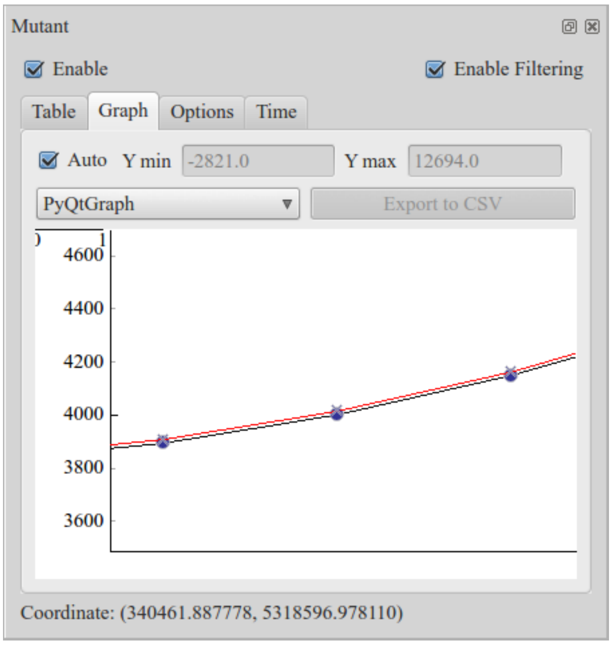

Figure 6.6: Current available filtering

A simple moving mean function is applied on the queried location reducing the noise.

Certainly it is planned to add a set of 'state of the art' time series filters (see section 6.2 on page 51). Currently the plugin does only provide help with visualization, but no real processing is available right now.

Masking by threshold (see figure 6.7) could considerably help the process of interpretation.

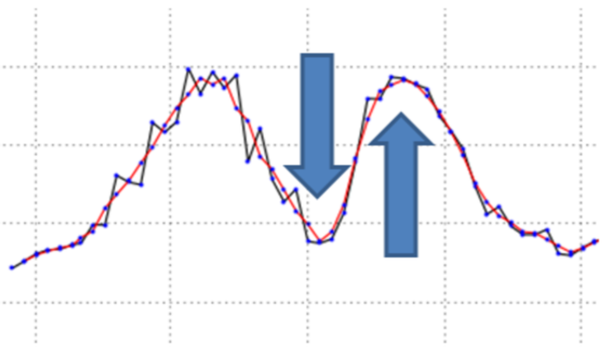

Figure 6.7: Select pixel that have a second crop

For example the interpreter cloud select pixels that have a given pattern, e.g. forcing to have a minimum or maximum NDVI on certain dates.

6.1.1.5 *Adding Algorithms*

As you can see in section 2.5.3 on page 21 QGIS works well together with Python. As Python itself is a very actively developed language there are also packages/modules available to enhance Python with capabilities that can afterwards easily be used.

One of this packages, named SciPy (`https://www.scipy.org/`), is a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization, statistics and much more. For example the Savitzky-Golay Filter from section 2.1.4.4 on page 14 is present in this collection and by including this package it should be easily possible to add not only this but much more filter enhancements to MuTAnT.

As a conclusion providing and using open source makes it extremely easy to enhance the tools you use for GIS and you do not have to reinvent the wheel over and over again, or at least write a whole filtering algorithm by yourself.

6.1.1.6 *Graphics libraries*

As mentioned in 4.3.6 on page 37 we discover that PyQtGraph is a fast library for displaying GIS data. Unfortunately there are yet no capabilities in the library for displaying date/time axes data. Even more during the development of the plugin some authors of several python graphic libraries decided to work together on a much faster and better library, among them also the author of PyQtGraph.

6.2 DISCUSSION

It has been a long way to come to the result of analysing and interactively viewing multi-temporal data with possibly algorithms applied. Not every problem has yet been solved and there are a lot of work in front to deal with them. But as a conclusion we have shown that there is a way to do it.

We can interactively analyse multi-temporal raster data from nearly any kind of source and apply algorithms to it. Moreover everyone dealing with this type of tasks now can do it too due to making everything available under the GNU GPL license.

Development will not stop here and by using the plugin on daily base and getting feedback from other users there will be continued development to make interactive analysis of raster data available for free for everyone to use.

BIBLIOGRAPHY

[1] M. E. Brown, K. B. Beurs, and A. C. Vrieling. „The response of African land surface phenology to large scale climate oscillations“. In: *Remote Sensing of Environment* 114 (2010), pp. 2286–2296.

[2] S. Chountasis et al. „The Whittaker smoother and the Moore-Penrose inverse in signal reconstruction.“ In: *Applied Mathematical Sciences (Ruse)* 6.25-28 (2012), pp. 1205–1219. ISSN: 1312-885X.

[3] R. R. Colditz et al. „TiSeG: A Flexible Software Tool for Time-Series Generation of MODIS Data Utilizing the Quality Assessment Science Data Set“. In: *IEEE Transactions on Geoscience and Remote Sensing* 46.10 (2008), pp. 3296–3308. DOI: .921412.

[4] L. Eklundh and P. Jönsson. „TIMESAT: A Software Package for Time-Series Processing and Assessment of Vegetation Dynamics“. In: *Remote Sensing Time Series: Revealing Land Surface Dynamics*. Ed. by C. Kuenzer, S. Dech, and W. Wagner. Cham: Springer International Publishing, 2015, pp. 141–158. ISBN: 978-3-319-15967-6. DOI: 10.1007/978-3-319-15967-6_7. URL: http://dx.doi.org/10.1007/978-3-319-15967-6_7.

[5] J. Grazzini. „The Open Source Phenomenon“. Working Paper. University of Turin, 2009.

[6] G. Hertel, S. Niedner, and S. Herrmann. „Motivation of Software Developers in Open Source Projects: An Internet Based Survey of Contributors to the Linux Kernel“. In: *Research Policy* 32.7 (2003), pp. 1159–1177.

[7] J. D. Hunter. „Matplotlib: A 2D graphics environment“. In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[8] P. K. Jönsson and L. Eklundh. „Seasonality extraction by function fitting to time series of satellite sensor data“. In: *IEEE Transactions on Geoscience and Remote Sensing* 40 (2002), pp. 1824–1832.

[9] P. Jönsson and L. Eklundh. „TIMESAT—a program for analyzing time-series of satellite sensor data“. In: *Computers & Geosciences* 30 (2004), pp. 833–845. DOI: 10.1016/j.cageo.2004.05.006.

[10] A. Klisch. „Review of methods for extracting phenology metrics from satellite time series data“. 2012.

[11] K. Lakhani and E. von Hippel. „How Open Source Software Works: "Free" User-to-User Assistance“. In: *Research Policy* 32.6 (2002), pp. 923–943.

[12] J. Lerner and J. Tirole. *The Simple Economics of Open Source*. Tech. rep. 2000.

[13] S. Lipovetsky. „Double logistic curve in regression modeling“. In: *Journal of Applied Statistics* 37.11 (2010), pp. 1785–1793. DOI: 10.1080/02664760903093633. eprint: http://dx.doi.org/10.1080/02664760903093633. URL: http://dx.doi.org/10.1080/02664760903093633.

[14] L. E. Loe et al. „Climate Predictability and Breeding Phenology in Red Deer: Timing and Synchrony of Rutting and Calving in Norway and France“. English. In: *Journal of Animal Ecology* 74.4 (2005), pp. 579–588. ISSN: 00218790. URL: http://www.jstor.org/stable/3505437.

[15] P. Mahau. *BAgS Beginning of Agricultural Season*. FAO Research, Extension, Training Division Environment, and Natural Resources Service (SDRN) Agrometeorology Group. 2004.

[16] M. Massart et al. „The use of remote sensing data and meteorological information for food security monitoring, examples in East Africa“. In: vol. Advances in Earth Observation for Global Change. 1. Chuvieco, Emilio, Li, Jonathan, and Yang, Xiaojun, 2010. Chap. 15. DOI: 10.1007/978-90-481-9085-0. URL: http://www.springer.com/earth+sciences+and+geography/remote+sensing/book/978-90-481-9084-3.

[17] M. Mattiuzzi et al. *MODIS: MODIS download and processing package. Processing functionalities for (multi-temporal) MODIS grid data*. First International Workshop on Temporal Analysis of Satellite Images Mykonos Island, Greece, May 23-25, 2012 (Poster). 2012.

[18] R. McKellip et al. *Phenological Parameters Estimation Tool*. Tech. rep. NASA Tech Briefs, 2010. URL: http://www.techbriefs.com/component/content/article/8481.

[19] R. McKellip et al. *Remote-Sensing Time Series Analysis, a Vegetation Monitoring Tool*. Tech. rep. NASA Tech Briefs, 2008. URL: http://www.techbriefs.com/component/content/article/2719.

[20] E. S. Raymond. *The Cathedral and the Bazaar*. Ed. by T. O'Reilly. 1st. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 1999. ISBN: 1565927249.

[21] F. Rembold et al. „Remote sensing time series analysis for crop monitoring with the SPIRITS software: new functionalities and use examples". In: *Frontiers in Environmental Science* 3 (2015), p. 46. ISSN: 2296-665X. DOI: 10.3389/fenvs.2015.00046. URL: http://journal.frontiersin.org/article/10.3389/fenvs.2015.00046.

[22] A. Rodrigues, A. R. Marcal, and M. Cunha. „PhenoSat — A tool for vegetation temporal analysis from satellite image data". In: *2011 6th International Workshop on the Analysis of Multi-temporal Remote Sensing Images (Multi-Temp)*. 2011. DOI: 10.1109/Multi-Temp.2011.6005044.

[23] K. W. Ross, B. A. Spiering, and M. T. Kalcic. „Monitoring phenology as indicator for timing of nutrient inputs in northern gulf watersheds". In: *Proc. MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges OCEANS 2009*. 2009, pp. 1–4. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5422234&isnumber=5422059.

[24] A. Savitzky and M. J. E. Golay. „Smoothing and Differentiation of Data by Simplified Least Squares Procedures". In: *Analytical Chemistry* 36 (1964), pp. 1627–1639.

[25] P. Smits and L. Bruzzone. *Proceedings of the Second International Workshop on the Analysis of Multi-Temporal Remote Sensing Images: Multitemp 2003, Joint Research Centre, Ispra, Italy, 16-18 July 2003*. Series in remote sensing. World Scientific, 2004. ISBN: 9789812389152. URL: https://books.google.at/books?id=tR4miGZhibIC.

[26] T. Udelhoven. „TimeStats: A Software Tool for the Retrieval of Temporal Patterns From Global Satellite Archives". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 4.2 (2011), pp. 310–317. DOI: 10.1109/JSTARS.2010.2051942.

[27] J. Verbesselt et al. „Detecting trend and seasonal changes in satellite image time series". In: *Remote Sensing of Environment* 114 (2010), pp. 106–115.

[28] F. Vuolo et al. „Data service platform for MODIS Vegetation Indices time series processing at BOKU Vienna: current status and future perspectives“. URL: http://www.rali.boku.ac.at/fileadmin/data/H03000/H85000/H85700/publications/33_Vuolo_et_al_2012_SPIE.pdf.

[29] E. T. Whittaker. „On a new method of graduation“. In: *Proceedings of the Edinburgh Mathematical Society*. Vol. 41. 1923, pp. 63–75.

[30] Wikipedia. *Landsat program — Wikipedia, The Free Encyclopedia*. [Online; accessed 27-September-2016]. 2016. URL: https://en.wikipedia.org/w/index.php?title=Landsat_program&oldid=740330431.

[31] Wikipedia. *Normalized Difference Vegetation Index — Wikipedia, The Free Encyclopedia*. [Online; accessed 26-September-2016]. 2016. URL: https://en.wikipedia.org/w/index.php?title=Normalized_Difference_Vegetation_Index&oldid=737273986.

[32] Wikipedia. *Qt (software) — Wikipedia, The Free Encyclopedia*. [Online; accessed 26-September-2016]. 2016. URL: https://en.wikipedia.org/w/index.php?title=Qt_(software)&oldid=740906139.

APPENDIX

.1 THE MUTANT MANUAL

This manual is trying to describe the functionality of the MuTAnT plugin in its current state as of September 30, 2016. It will be a basic manual for MuTAnT and only a start for further development because the development of MuTAnT will not stop at this time. During writing this thesis and trying out new algorithms and functionality it becomes more and more clear that in some parts of the plugin a complete rewrite of the code is necessary to provide better functionality and compatibility for future functions and enhancements.

.2 A BRIEF INTRODUCTION TO MUTANT

At the time of writing, version v0.3.1 of MuTAnT is the latest official published version available in the online QGIS plugin repository.

There will surely be some further development on MuTAnT and the user interface might slightly change over time as also QGIS itself will surely change during development as QGIS is currently in front of a major change with updating the libraries it is based on from Qt4 to Qt5 and Python2 to Python3.

The basic functionality will surely be the same, but new features (like for e.g. more algorithms to filter data) will be implemented as well as some rewrites to the used graphic libraries (especially Pyqtgraph, which will maybe be replaced by its successor VisPy).

Lets start this manual with a figure of QGIS with loaded MuTAnT plugin and some loaded data (see figure .8 on the next page)

So what is MuTAnT?

MuTAnT is a plugin for a FOSS GIS system called QGIS.

MuTAnT stands for "MUlti Temporal ANalysis Tool". The plugin installation inside QGIS is straight forward.

.2.1 *Installation inside QGIS*

Make sure you are running the latest stable version of QGIS. There are 2 choices available you have to install QGIS.

1. Install the "QGIS Standalone Installer Version" Version for newcomers or

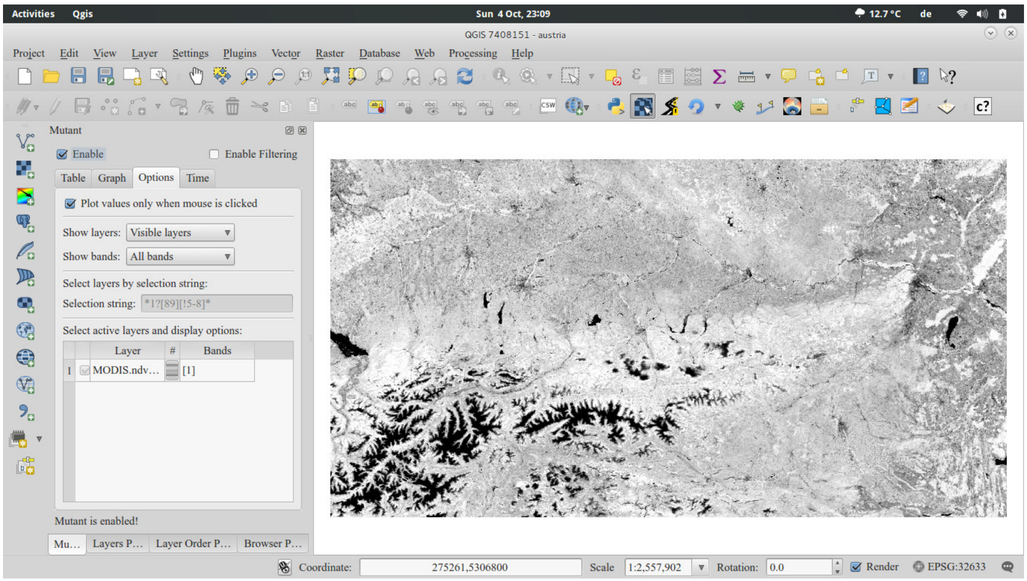

Figure .8: QGIS with activated Mutant plugin

2. use the more advanced OSGeo4W (which means OSGeo for Windows) installer which includes a whole environment of different FOSS software for GIS.

In this case we assume you are an advanced user and want to download more software for analysis later.

To download and install QGIS and dependencies for windows go to: `http://trac.osgeo.org/osgeo4w/` and follow the instructions found in 'Quick Start for OSGeo4W Users'. Run this installer multiple times until no updates/packages are found.

.2.1.1 *Other OS than Windows*

For MacOSX or Linux users there are prepared packages available within the QGIS Webpage (provided as links or howto) or in the package manager of the linux distribution of your choice.

.2.2 *MuTAnT (MUlti Temporal ANalysis Tool)*

The MuTAnT plugin is based on the Value Tool version v0.8.5. It uses the capabilities of efficient raster data access and the general graphical scheme. The development of MuTAnT is done to satisfy the requirements of multi-temporal data visualisation and processing. MuTAnT is currently still in a development stage but it already has implemented the crucial functionalities. The current state allows a reliable and seamless use of the plugin, but for the mid to long term plans, mainly the plot capabilities, such as the interaction with the plots, and times series processing capabilities are planed.

Also the access to more sophisticated data sources such as querying remote servers it planned for later versions. Some of this capabilities are available in already existing plugins and as far as possible the solutions found in those plugins will be integrated into MuTAnT.

.2.2.1 *Installation of the plugin*

The installation is done through the default QGIS Plugins installation system.

Follow the steps below to install and start the plugin:

1. Open QGIS
2. Click on 'Plugins' (see figure .9)

Figure .9: Open Plugin manager

3. Click on "Manage and Install Plugins"
4. Make sure plugins are regularly updated by using: Settings -> Check for updates on startup (see figure .10)

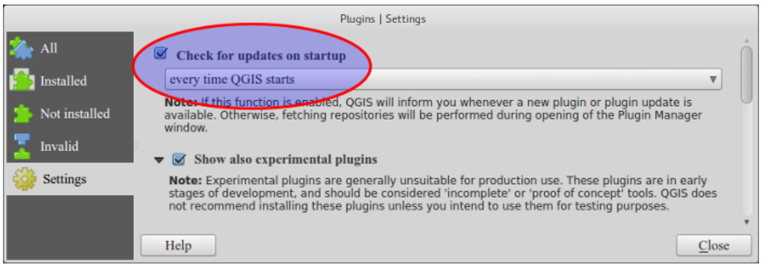

Figure .10: Make sure your plugins are regularly updated

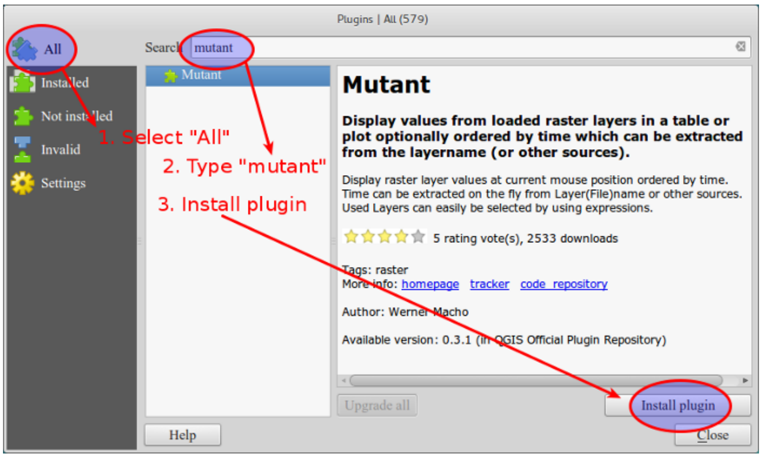

Figure .11: Install MuTAnT

5. Click on All -> Search: 'MUTANT' -> Install Plugin (see Figure .11)

6. Close the plugin manager

7. Activate the MuTAnT plugin (see Figure .12) Either click on the Icon

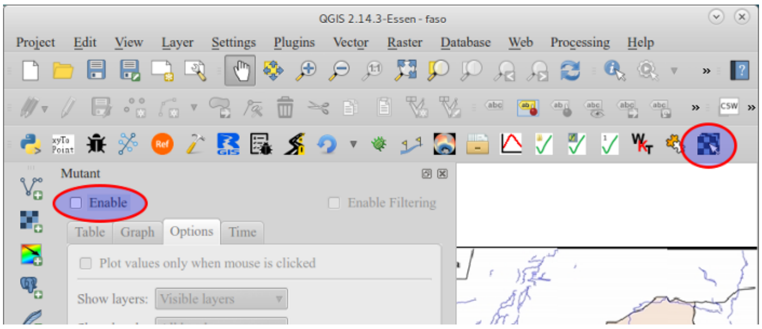

Figure .12: Enable MuTAnT

or activate MuTAnT with its enable clickbox

.2.3 *Ordering a MODIS time series*

To start working with this plugin you can order a multi-temporal MODIS dataset from the following site: `http://ivfl-info.boku.ac.at/index.php/eo-data-processing/dataprocess-global` (see .13 on the following page[28].

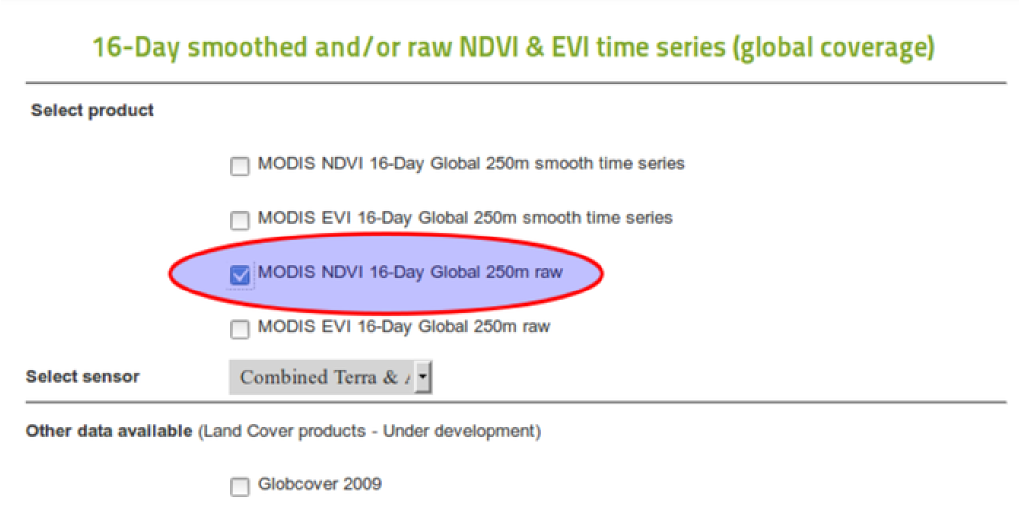

Figure .13: Downloading MODIS data from BOKU

As you can see in figure .14 on the next page you are able to select the are of interest by moving the upper left and lower right corner of the rectangle in the webpage.

After this you have to enter the correct data to be able to generate and prepare the MODIS data for downloading. (see figure .15 on page 69)

After submitting you will get an email with submission details. The time for processing depends on data availability and spatial and temporal extent, but usually it completes within 24 hours and you will be noticed after the process is completed. To see the progress stage or download the processed data click on 'Results' or access the following link: `http://ivfl-info.boku.ac.at/index.php/eo-data-processing/dataprocess-global/master-global`

.2.4 *Easy download of test data*

Recently a new QGIS plugin was released which perfectly fits in the workflow to just quickly test how MuTAnT works.

This plugin is called Oceancolor Downloader and allows easy download of oceancolor and sea surface temperature data from NASA Oceancolor `http://oceancolor.gsfc.nasa.gov/cms/`.

It downloads either global level 3 mapped chlorophyll-a concentrations or sea surface temperatures within a defined time range, and resolution. The data is saved in GeoTiff format and can be added to the QGIS canvas once downloaded.

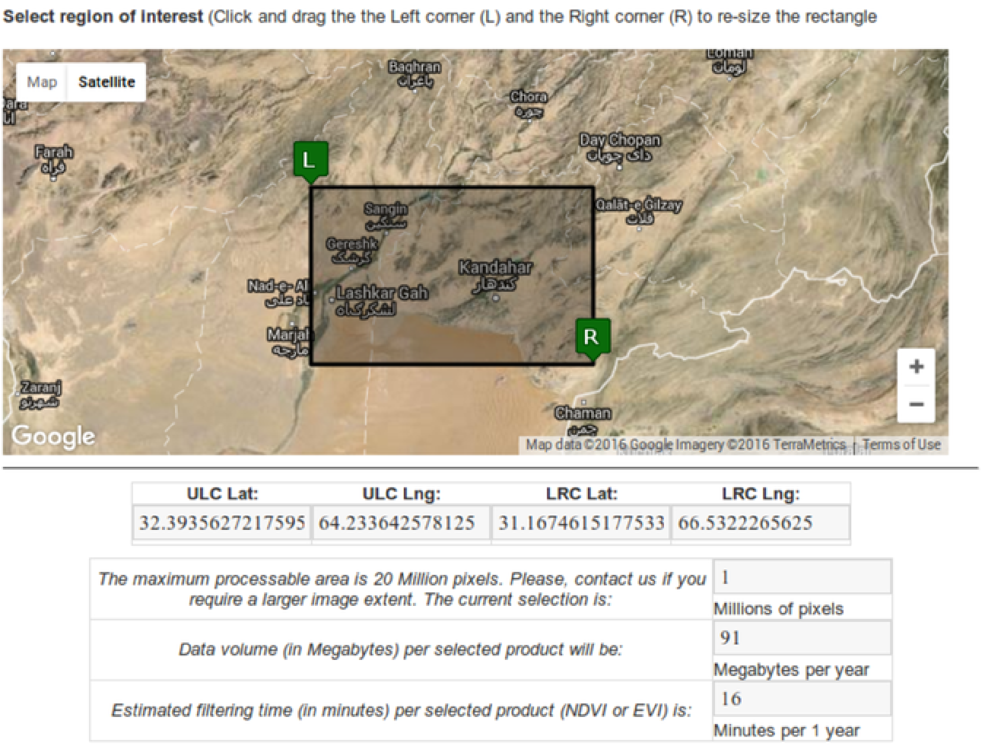

Figure .14: Selecting a region of interest

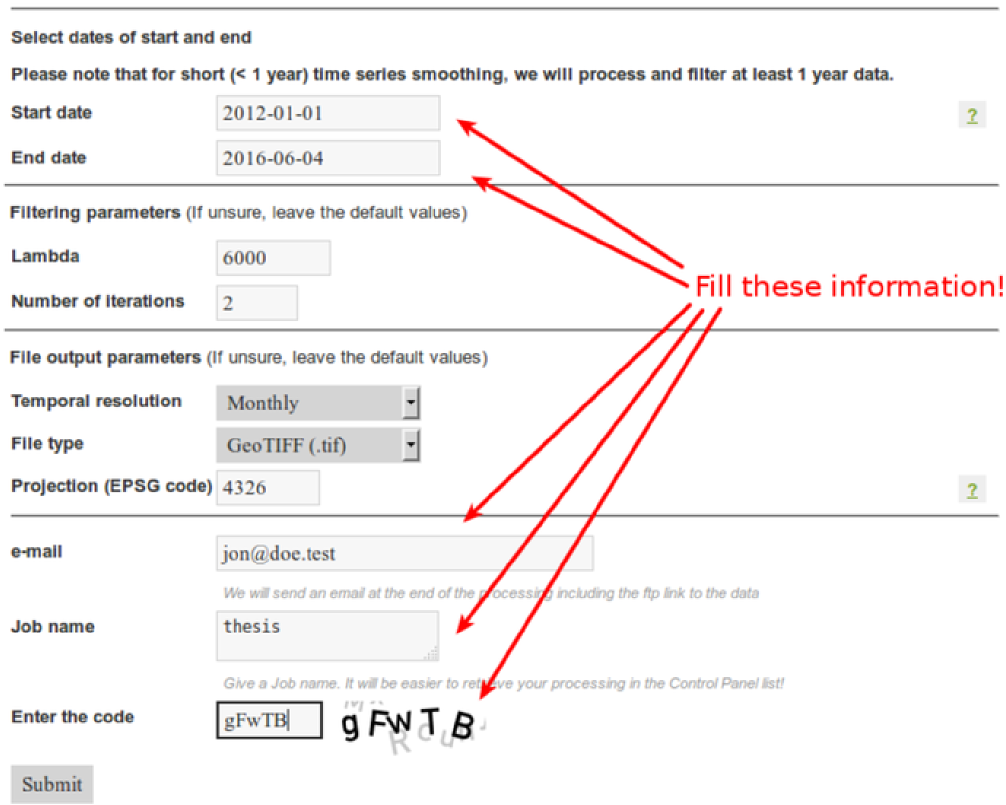

Figure .15: Filling the correct Information

All data is sourced from NASA Oceancolor. Full details on the input data can be found at http://oceancolor.gsfc.nasa.gov/cms/, and includes algorithm descriptions (http://oceancolor.gsfc.nasa.gov/cms/atbd.

The plugin currently provides access to three datasets:

- MODIS AQUA CHL-a concentration
- SeaWiFS CHL-a concentration
- MODIS AQUA Night Sea Surface Temperatures

The suggestion is to download the annual sea surface temperature data which will create a lot of new layers when you enable the "Add to canvas" switch. You surely also want to select a directory for the downloaded data.

After having downloaded all the new layers you can enable the MuTAnT plugin (obviously you already have downloaded MuTAnT by now).

After enabling the MuTAnT plugin go to the options tab and switch to "show all layers" and "show all bands". If you switch to the "Graph" Tab now and move the mouse over the canvas you can see a plot representing all values under the mouse of all layers (ordered more or less after how they have been loaded into the layers tab). But this plot still has only the number of layers a x-axis.

Now you can switch to the "Time" Tab and enable the multi temporal analysis. First move the "Filename" using drag&drop to the first position in the list. All times will than be taken directly from the filename. As all layers have the same content and were downloaded from the same source they all have (and should have) the same layername. You can see one sample taken from the filenames. Layernames should look somewhat like "A20090012009365.L3m_YR_NSST_9" which means we have to shape the layernames a bit to extract the time part.

First you have to decide which date you would like to choose. As we downloaded annual data it seems that it is created with two dates in the filename. In our example this is 2009001 and 2009365 which means the first day of 2009 and the last day of 2009

Using the two lines above Sample you are able to cut the first character (the A) and the length of the date string is 7 characters. According to the date time from python your Date pattern should be %Y%j: %Y is the year with 4 characters and %j is the day of the year with 3 characters. Notice the background of the "Date pattern" field switching from red to green when you enter a valid extraction pattern.

After having extracted the datetime stamp from the filename you are now able to switch back to the "Graph" Tab and you have your multi temporal raster analysis on the correct order.

Now you can take a look how the ocean temperature changed in the downloaded timeframe by moving the mouse across the canvas (see figure .16). That is real interactivity in finding multi-temporal data plots.

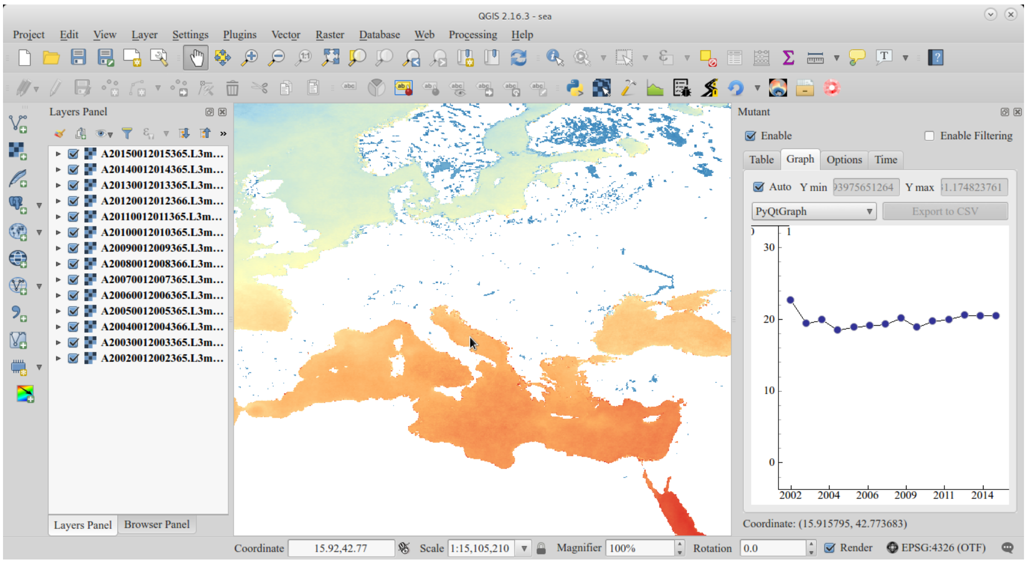

Figure .16: Example of downloaded ocean temperatur data

.2.5 *Loading raster into QGIS*

Loading raster data into QGIS is done the usual way by clicking on the "Add Raster Layer" icon in the toolbar or using the menu "Layer" -> "Add Layer" -> "Add Raster Layer...". Both lets you select the path where your raster layers are stored and lets you load them into QGIS

.2.6 *Enable the MuTAnT plugin*

You have to possibilities to enable MuTAnT. Either by clicking its icon in the toolbar or by enabling (and disabling) it on its own switch in the panel that is activated when you install MuTAnT (see figure .12 on page 66)

.2.7 *MuTAnT options in detail*

The following section will give a description about all the options available within the MuTAnT plugin.

.2.7.1 *The Table TAB*

In the Table TAB (see figure .17) you can see (also interactively) the Layername, the actual value at the mouse cursor (depending on what is enabled in the Options TAB (see section .2.7.3 on the following page))the Date (if enabled in the Time TAB (see section .2.7.4 on page 74)) and additionally the coordinates of the point of the mouse cursor. Enabling the Clickbox for the Decimals means to set the maximum numer of digits after the comma in the Values. This is sometimes important if you want to export the whole table with the "Export to CSV" button.

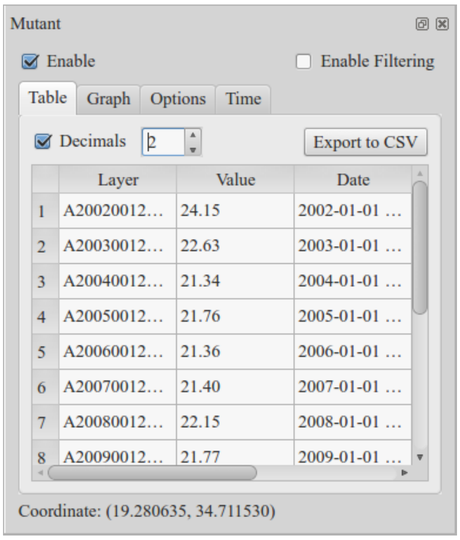

Figure .17: Options of the Table TAB

.2.7.2 *The Graph TAB*

The Graph TAB is the most important TAB inside the MuTAnT plugin (see figure .18 on the following page). Depending where your mouse cursor is and the settings in the options it will give you a complete graphical overview of all values of all loaded rasterfiles at the mouse cursor coordinate.

The Minimum and Maximum Value is set to "Auto" per default but you can disable auto and enter the range of the values to display yourself. Important to know is that the "Auto" setting is always choosing the minimum respectively the maximum value of all loaded raster data in all layers.

With the dropdown box you can select between 3 different graphical libraries to use (see section 5.1.3 on page 47 to read about the advantages and disadvantages of each of them).

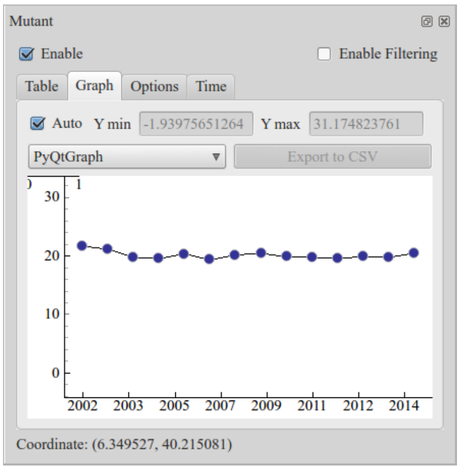

Figure .18: Options of the Graph TAB

.2.7.3 *The Options TAB*

At the options TAB (see figure .19 on page 75).you can set all necessary options for MuTAnT. The top clickbox is self explaining. With "plot Values only when mouse is clicked" you can switch between only displaying values when you click with the left mouse button while hovering over the canvas or, if you disable the clickbox, it is the fully interactive mode where every change of raster value below the mouse button is afterwards getting displayed either in the Graph TAB or in the Table TAB.

With the dropdown box at "Show layers" you can select the layers you want to show the values from independent from the setting you did in the Layers Panel in QGIS.

The options you can select from are:

- Visible layers
- All layers
- Manual selection

- by selection string

Visible layers means to take the values from all layers which are set to be visible in the Layers Panel from QGIS. All layers means literally all layers that are currently loaded.

Manual selection activates the bottom data sheet where all loaded layers are displayed. There you can select the layers manually by clicking the clickbox in the first column next to the layer names.

With "by selection string" the "Select layers by selection string" input box becomes active. There you can select the layers you want to be included in the graph by entering a selection string which is a bit hard to learn at first, but once you are used to it it also becomes very handy and fast. Leaving the mouse pointer over the input field gives you a short manual howto use the selection strings after a few seconds.

The "Show bands" drop-down box works similar to the "Show layers" drop-down box.

The options you can select from are also very similar:

- All Bands
- Active Bands
- Selected Bands

If you work with multiband raster layers this becomes handy when you want to use only some bands of the available. All Bands is self explaining. Active are the ones already set in the bottom box inside the column "Bands". There you also can select the Bands which becomes active if you select "Selected Bands" in the Dropdown box.

.2.7.4 *The Time TAB*

In figure .20 on the next page you can see the options which are available to extract and store time/date data from the loaded raster data. With the clickbox at the top you can enable and disable the multi temporal analyses and switch on time extraction.

In the Value Field with XML, Filename, Exif, TIFF-Header you can enable and disable the method to extract the date and drag and drop prefered method to the top to prioritise it. The 2 Boxes with Cut characters and the lenght of datestring are already described in section .2.4 on page 67.

Here three fields are relevant:

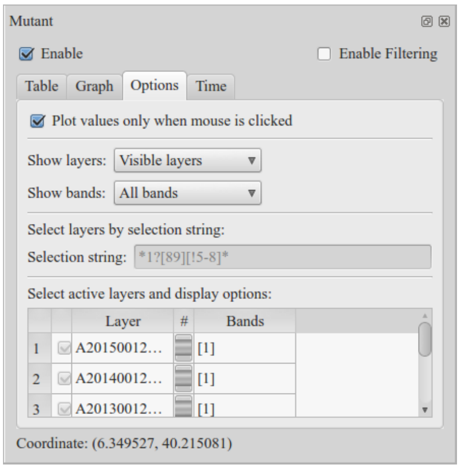

Figure .19: Options available in the Options TAB

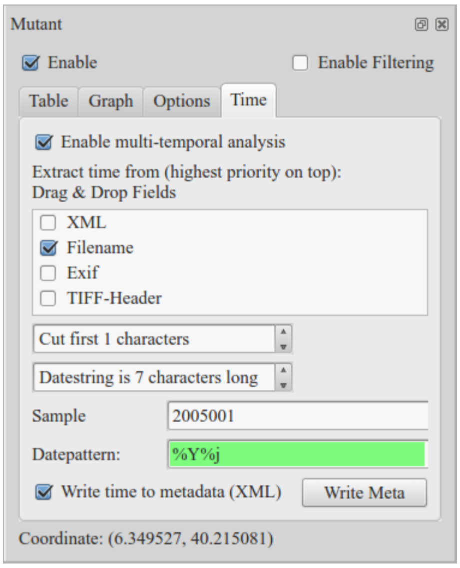

Figure .20: Options of the Time TAB

- 'Cut first x characters' indicates the start position of the date in the file name.
- 'Datestring is x characters long' indicates how many characters the date is composed of.
- 'Datepattern' indicates how the date is formatted in the filename. '%Y%j' is equivalent to a 'YYYYDDMM' format and in most cases the right choice.

The information about what the YYYYDDMM means is available at https://docs.python.org/3/library/datetime.html#strftime-and-strptime-behavior

The procedure has to be done only once for one of the files (if they are all named in the same way). If the datetime value is stored to the according XML file all you have to do is to enable reading from XML afterwards and you could also compare files with different sources. After the first time the date is saved into the file metadata (XML) of all loaded files and remains available for further use with the switch 'read from Metadata' (see section 4.5.5 on page 42).

Without assigning the file date information (see figure .21) the plugin is not able to sort the layers accordingly.

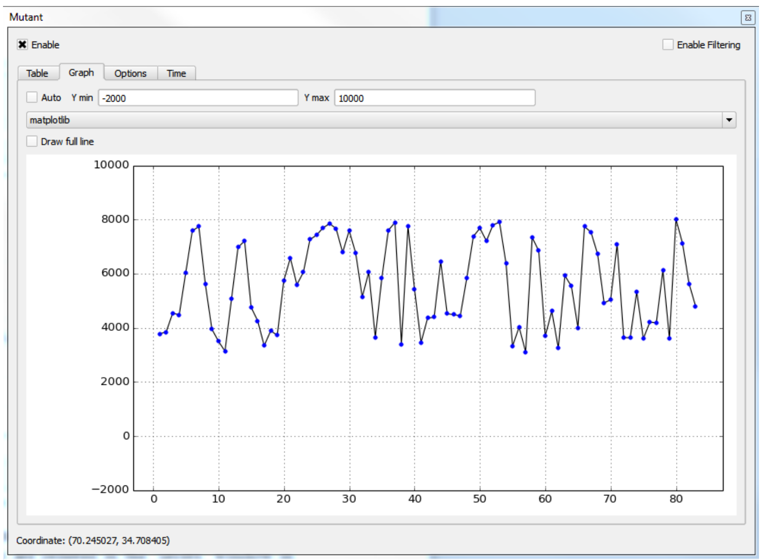

Figure .21: Unsorted data with time disabled

It takes the files as they are ordered in the 'layers' window in QGIS (left side of figure .21) and the x-axis is just a numeration of that order. Here the files have been loaded in an unsorted order by purpose to better show the problem.

After indication the date of the files graph the temporal behavior:

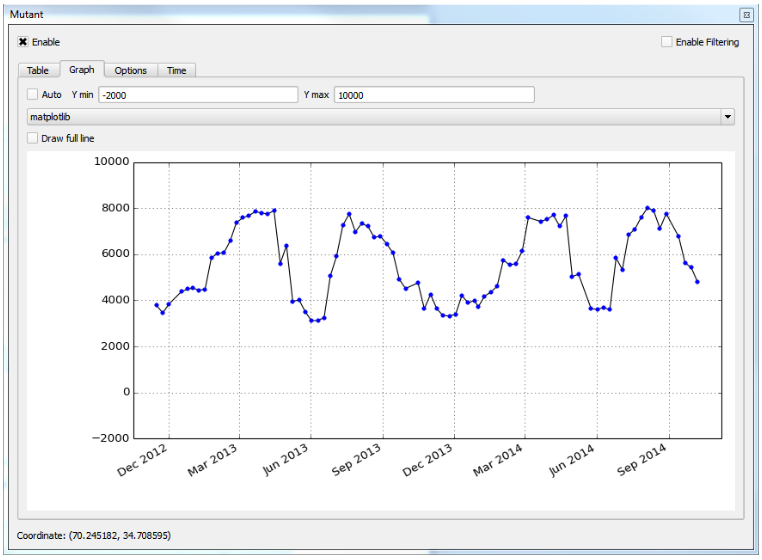

Figure .22: Sorted data with time enabled

In figure .22 you can see the same Time series plot after enabling the time information from files. The values are now ordered according the specific file date and the x-axis is now a precise date.

In figure .22 it is clearly visible the two crop cycles per year, both peaks of one year are above 6000 very high (y-axis is NDVI*10000), this location is a irrigated high production agricultural land and with typical signature of potential cannabis presence (second peak in each year).

Additionally a simple smoothing approach is available by enabling the filter tickbox in the top right of the plugin panel. By enabling filtering (see figure .23 on the following page) a local mean filter reduces the impact of noise in the plot, making the visual interpretation easier.

The local mean filter is just a prove of concept more sophisticated approaches will follow. Also the appearance is temporary as the lines thickness and color will become selectable.

.2.8 *Some helpful hints*

Here you can find some hint to speed up your work with the MuTAnT plugin.

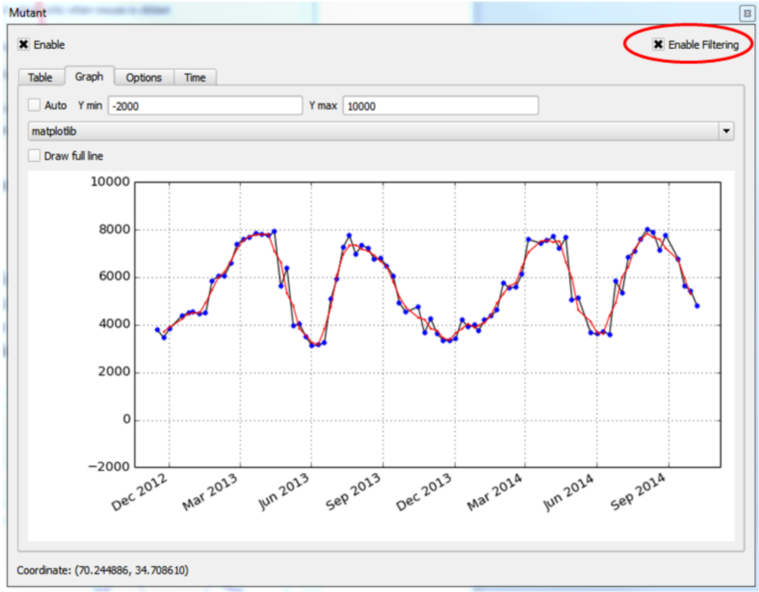

Figure .23: Enabled filter as a first step for adding more algorithms

.2.8.1 *Improving rendering speed*

Every time a change is done in the GIS canvas the content has to be rendered, this takes more time if more files are loaded and leads to a frustrating situation. The solution is to disable the visualization of the time series files by disabling their visualization in the Layers Panel. This is easily done by clicking the icon with the eye above the Layers Panel and select "Hide all Layers" or using the keycombination "CTRL-Shift-H". For orientation purposes you can now reactivate the topmost Layer by clicking the empty rectangle on the left side of the layer name. Now QGIS is only rendering one layer (the topmost) while all other layers do not render when paning or zooming the canvas. Still all data from all other layers is available to the MuTAnT plugin and can be displayed by selecting them properly in the MuTAnT options (see figure .24 on the next page)

.2.8.2 *Moving the MuTAnT Panel*

You can pick the MuTAnT Panel and either move it to its own screen for better view on the GIS data to investigate or even drag it to the right side of the main QGIS window to have better and easier access to the layers (or any other) menu (see also figure .24 on the following page).

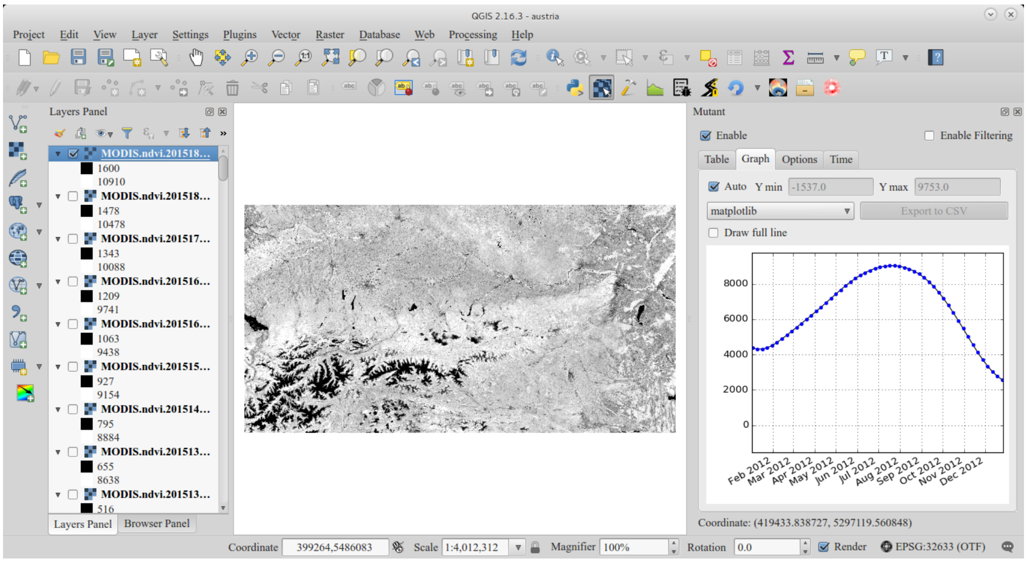

Figure .24: Selecting only the first layer for speed improvements

.3 A LETTER FROM UNO

```
Delivered-To: werner.macho@gmail.com
To: Werner Macho <werner.macho@gmail.com>
From: Coen BUSSINK <coen.bussink@unodc.org>
Date: Wed, 29 Jun 2016 16:00:23 +0200

Dear Werner,

Great to get to know you and thanks for your help in designing
the plugin. Yes, Mutant was helpful particularly for Mexico,
where we analysed the cropcycles with Landsat time series.
Last week the survey report on Mexico came out and mentions
the tool.
Unfortunately the name of the tool was not included, something
that we could do for the next publication.

  https://www.unodc.org/unodc/en/crop-monitoring/index.html?tag=
3DMexico

Also for our Afghanistan survey the tool was used (with MODIS
images), but the report actually doesn't mention explicitly
the use of the tool yet,something that we should do for the
next one as well.
```

https://www.unodc.org/documents/crop-monitoring/Afghanistan/_Afghan_
opi=um_survey_2015_web.pdf

I don't think that Matteo already used the tool for this research, but the analysis he did was the base for the development I think, you can check with Matteo:

https://www.schweizerbart.de/papers/pfg/detail/2014/84423/Analysing_
Phenological_Characteristics_Extracted_from_Landsat_NDVI_Time_
Series_to_Identify_Suitable_Image_Acquisition_Dates_for_Cannabis_
Mapping_in_Afghanistan?l=3DDE

Best regards,

Coen

____

Coen Bussink (Mr.)
Programme Officer (Research) and Team Leader of the Illicit Crop Monitoring Programme
Programme Development and Management Unit / Research and Trend Analysis Branch
United Nations Office on Drugs and Crime (UNODC)
PO Box 500, 1400 Vienna
Austria

Tel: (+43-1) 26060-4165, Fax: (+43-1) 26060-74165

Email: coen.bussink@unodc.org
http://www.unodc.org/unodc/en/crop-monitoring/index.html