Labwork Remote Sensing

How to use and to extract information from remote sensing images for land management ?

Mathieu Fauvel

2015-2016

Contents

1 Introduction

```
FORMATION: Presential
DURATION: 0:30
SEQUENCE: 1
```

1.1 Objectives of the labworks

The main objective of these labworks is to

be able to use and to extract information from remote sensing images for land management.

Information can be any knowledge of a given landscape (landcover, land-use, humidity, ...) that is used to understand the configuration and/or the evolution of landscape.

In terms of *competences*, you should be able to master at the end of the sessions the items listed in tables 1, 2, 3, 4 and 5. Each of them is organized as a set of tasks that should be mastered progressively.

Table 1: Choose images with properties adapted to your problematic.

Remember	Properties of remote sensing images.
Understand	Physical meaning of each sampling of a given image.
Apply	Open and visualize a remote sensing image, extract its properties.
Analyze	Describe a remote sensing image. Recognize specific object.
Evaluate	Choose the good image adapted to what you are looking for.
Create	Create a set of properties needed for your problematic.

Table 2: Compute and use spectral indices.

Remember	Definition of spectral indices in general and of the NDVI in particular.
Understand	What (and why) does the NDVI emphasize.
Apply	Perform the computation of spectral indices.
Analyze	Analysis of the vegetation cover using NDVI.
Evaluate	Choose the right spectral index.
Create	Select from the literature a set of possible indices.

Table 3: Define, identify and analyze radiometric behavior.

Remember	Spectral signature for *vegetation* and *water* object.
Understand	Histogram of images.
Apply	Compute an histogram.
Analyze	Extract radiometric properties of some classes.
Evaluate	Choose relevant spectral bands and/or indices for the segmentation of several classes.
Create	Perform a segmentation using radiometric statistics on one or many spectral variables and/or indices.

1.2 Remote sensing software

In these labworks, free and open sources softwares will be used to visualize remote sensing images, to process them and to implement processing chains. In the following, each software/tools will be briefly described. Interested reader can find more information on the associated website. In particular, the installation process is not detailed. However, they can be freely download and installed on many operating systems from their official website.

1.2.1 Orfeo ToolBox (OTB)

OTB is a C++ library for remote sensing images processing. It has been developed by the CNES (French space agency) during the ORFEO program to *prepare, accompany and promote the use and the exploitation*

Table 4: Do and analyze pixel-wise classification of image

Remember	Definition of pixel-wise supervised classification.
Understand	The parameters of several classification algorithm, how the spatial sampling of the ground truth data influence the training and validation steps.
Apply	Classification algorithms.
Analyze	Interpret the confusion matrix and the thematic map, the quality of a ground truth.
Evaluate	Compare two classification maps/results.
Create	Choose the most appropriate classifier for one given application, build good ground truth data.

Table 5: Define and implement a processing chain

Remember	How to combine several bands, apply a given function to train a classifier or to predict a thematic map.
Understand	The different inputs and outputs of `OTB` functions, how to use their corresponding documentation.
Apply	Apply a set of different functions in a pipeline.
Analyze	Define the different processing needed to perform a given task.
Evaluate	Evaluate the accuracy of the given processing, check for errors.
Create	Shell scripts that automatize most the processes, in order to apply them on a large set of images or to apply several embedded processes.

of the images derived from Pleiades satellites (PHR). Processing tools from OTB are appropriated to big images. When possible, processes are paralyzed and tiled automatically for users. Many applications derived from OTB and called *OTB-Applications* are directly usable for most of the common processing, they are described here. For advanced users, it is possible to develop program based on the OTB library (not considered in these labworks).

Monteverdi2 is *graphical user interface* that allows users to visualize and process remote sensing images with *OTB-Applications*. It is also developed by the CNES during the ORFEO program.

1.2.2 QGIS

QGIS is a *Geographic Information System* (GIS). It is used to open, visualize and process digital map. It includes several spatial analysis tools working mainly on vector data. QGIS can be extended by several plugin (`https://plugins.qgis.org/`) and modules, such as the OTB applications.

1.2.3 Geospatial Data Abstraction Library (GDAL)

GDAL is a library for the processing of raster and vector data. Similar to OTB, it has several applications that can be used directly. For advanced users, it is possible to develop program based on the GDAL library (not considered in these labworks).

1.2.4 Python

Pyhton is a programming language. It has several programming capabilities, such as *object-oriented*, *functional programming*, *dynamic type* and *memory management* that make it widely used in several applications:

- Web and internet development,
- Scientific and numeric computing,

- Software development.

It has a large number of available packages that can be used in many applications. For instance, it is possible to call *OTB-Applications* or *GDAL* from Python.

1.3 During the labworks

For the *presential* sequences, you won't have to do any report. But you will have to write your personal material on remote sensing. You are encouraged to write it progressively during the sessions. **It will be the only document approved for the exam** (with those on moodle). The length of each sequence should let you enough time to write the report.

For the *non presential* sequences, you will be asked to write a document that describe briefly the results and how you obtained them. Discussion between all groups will be done during the next session.

2 Data sets

```
FORMATION: Presential
DURATION: 0
SEQUENCE:
```

2.1 Pleiades images

These images were acquired over the Fabas forest in 2013. Images were acquired the <*2013-10-12 Sat*> and the <*2013-12-10 Tue*>, respectively. A true color composition is given in Figure 1.

Figure 1: Fabas image acquired the <*2013-10-12 Sat*>.

Images are stored using the GeoTIFF format. It is an extended version of the TIFF format, which allows to embed geospatial information within the file. GeoTIFF can be read by most of the remote sensing and GIS software.

3 Visualization of remote sensing data

```
FORMATION: Presential
DURATION: 1:10
SEQUENCE: 1
```

3.1 Vizualization of remote sensing image

The vizualisation of remote sensing images can be done either with Monteverdi2 or QGIS[1]. QGIS might be a more efficient when it comes to visualize several images, or for the vizualisation of vector layers. It will be used in these labworks.

Most of the information regarding the vizualisation of raster data with QGIS can be found online http://docs.qgis.org/2.14/en/docs/user_manual/working_with_raster/raster_properties.html.

More generally, to use raster data with QGIS is described here http://docs.qgis.org/2.14/en/docs/user_manual/working_with_raster/index.html.

In this labwork, a few properties will be reviewed and you are encouraged to check (at least) the given references.

3.1.1 Vizualization of grayscale image

Open the image *fabas_10_12_2013.tif* with QGIS. The default view is a colour composition, with the bands/channels association given in Table 6. To start easy, we just open one band at a time: right click on the name of the opened image in the *Layer* pane et select *Properties*. Then select the tab *Style* and *Band rendering*. In the *render type*, select *Singleband gray* and the band you want to display.

You surely have to do *Contrast enhancement*. Check the doc for that.

Table 6: Bands and channels default association in QGIS (if there is not a set of specified spectral bands in the metadata).

Band	Channel
1	Red
2	Green
3	Blue

Work 3.1:

1. Visualize each spectral band of the data, and look at the differences in terms of graylevel between spectral bands.
2. Zoom in/out: use the mousse's wheel to zoom into the image. What do you observe ?

3.1.2 Vizualization of colour image

Now you can visualize a colour images, by selecting three spectral bands among those available from the data. Again, *Contrast enhancement* should be done.

[1]The library `matplotlib` of python is not adapted to visualize remote sensing image and should be avoided.

Work 3.2:

1. Do a "true colours" and "false colours" compositions and compare what is easily seen on each of them.
2. Get spectral values for several pixels corresponding to different materials (water, grassland, forest and bare soil). For that, use the tool *Identify features*, see `http://docs.qgis.org/2.14/en/docs/user_manual/introduction/general_tools.html` for detail.
3. Fill the *collaborative spreadsheet* with your pixel values:
   - `https://framacalc.org/fauvel_rs_water`
   - `https://framacalc.org/fauvel_rs_grassland`
   - `https://framacalc.org/fauvel_rs_forest`
   - `https://framacalc.org/fauvel_rs_baresoil`

3.2 Get data information

Before opening a remote sensing data, it is possible to get some information about its properties. For instance, using `gdalinfo` it is possible to extract several information. It can be used as

```
gdalinfo fabas_10_12_2013.tif
```

Help on the function can be obtained using the command alone or by doing :

```
man gdalinfo
```

Equivalently, it is possible to get the same information using the function `otbcli_ReadImageInfo` from the *OTB-Applications*:

```
otbcli_ReadImageInfo -in fabas_10_12_2013.tif
```

Work 3.3:

On the *Fabas* data set, get the following information.

1. Number of lines, columns and bands,
2. Size of each pixel,
3. Numerical types for coding pixel values,
4. Position of the upper left pixel,
5. Projection.

4 Spectral indices: *Normalized Difference Vegetation Index*

```
FORMATION: Presential
DURATION: 01:00
SEQUENCE: 2
```

Among the available radiometric indices, only the NDVI is considered in this labwork. NDVI is widely used for vegetation monitoring because it can be related to chlorophyll content and photosynthesis.

Work 4.1:

1. Compute the NDVI for each *Fabas* image. You can compute the NDVI using several ways, using either *OTB-Applications* or the *Raster Calculator* http://docs.qgis.org/2.14/en/docs/user_manual/working_with_raster/raster_analysis.html#raster-calculator. For a per band analysis, both methods are equivalent. Using QGIS provides the Graphical user interface, which can be convenient for processing few images, while *OTB-Applications* allow to process large number of images using *shell* programming.

   Using the raster calculator, the following formula can be used (for the Fabas image):

   ```
   ("fabas_12_10_2013@4"-"fabas_12_10_2013@1")/("fabas_12_10_2013@4"+"fabas_12_10_2013@1")
   ```

   Using the *OTB-Applications*, it is possible to use `otbcli_BandMath`. The syntax is similar, since we need to define the image, the bands used and the expression of our processing:

   ```
   otbcli_BandMath -il fabas_12_10_2013.tif -out ndvi_fabas.tif -exp "(im1b4-im1b1)/(im1b4+im1b1)"
   ```

2. Compare the two NDVI and explain your results.

5 Segmentation of remote sensing images

```
FORMATION: Presential
DURATION: 2:00
SEQUENCE: 2
```

5.1 Radiometric analysis

Work 5.1:

For the *near infra red* band and the NDVI of the image *[2013-10-12 Sat]*, do

1. Look at the histogram and identify the local maxima. For each local maximum, try to identify the corresponding pixels in the image,

2. Keep track of the characteristics of each identified maximum (position and width).

5.2 Segmentation of 1D histogram

In this part, the extraction of image's pixels sharing the same *radiometric behavior* is considered. The analysis of the histogram is used to estimate this *behavior*. When only one material is segmented, the output is a binary image (image with value `0` or `1`), where pixels having value `1` are from the same material. Figure 2 gives an example of such outputs. When several material are considered, the output is an images with integer values (`1`, `2`, `3` ...), depending on the number of materials.

A usual work-flow is proposed in this part. First, QGIS is used to analyze the data and set-up the processing (parameters *etc*). Then, the *OTB-Applications* are used to automatize the processing.

Work 5.2:

For the *near infra red* band and the NDVI, segment the identified material. For that, you need to define interval of pixel values for which a specific action is done (*e.g.*, set the value to 0 or 1). Implement the processing using the `BandMath` application or the Raster calculator.

Figure 2: Binary image for Water.

5.3 Graphical Modeler

For the segmentation of the NVDI, two processings are required

1. First, the computation of the NDVI from the original image,
2. Second, the definition of the interval of values to extract the relevant pixels.

With the graphical modeler, it is possible to define your workflow, to automatize complex tasks. Take a look at `http://docs.qgis.org/2.14/en/docs/user_manual/processing/modeler.html`.

Work 5.3:

Define your model to perform the segmentation of intro three classes of the NDVI.

6 Change detection: *Detection of floods*

```
FORMATION: Presential
DURATION: 01:00
SEQUENCE: 2
```

Change detection in remote sensing consists in detecting differences between two images, or a set of images. It can be used to detect changes in vegetation properties or in land cover. It is also used in disaster management, to detect impacted areas. In this labwork, we are dealing with floods. In Figure 3 is shown two quickbird images, before and after a flooding. The objective is to identify the impacted area to provide a map of these zones

Work 6.1:

1. Characterize the impacted zones in terms of radiometric behavior, *i.e.*, what is the variation in terms of spectral values. And why ?
2. Define the processing chain to extract these areas.
3. Implement the processing chain with the graphical modeler.

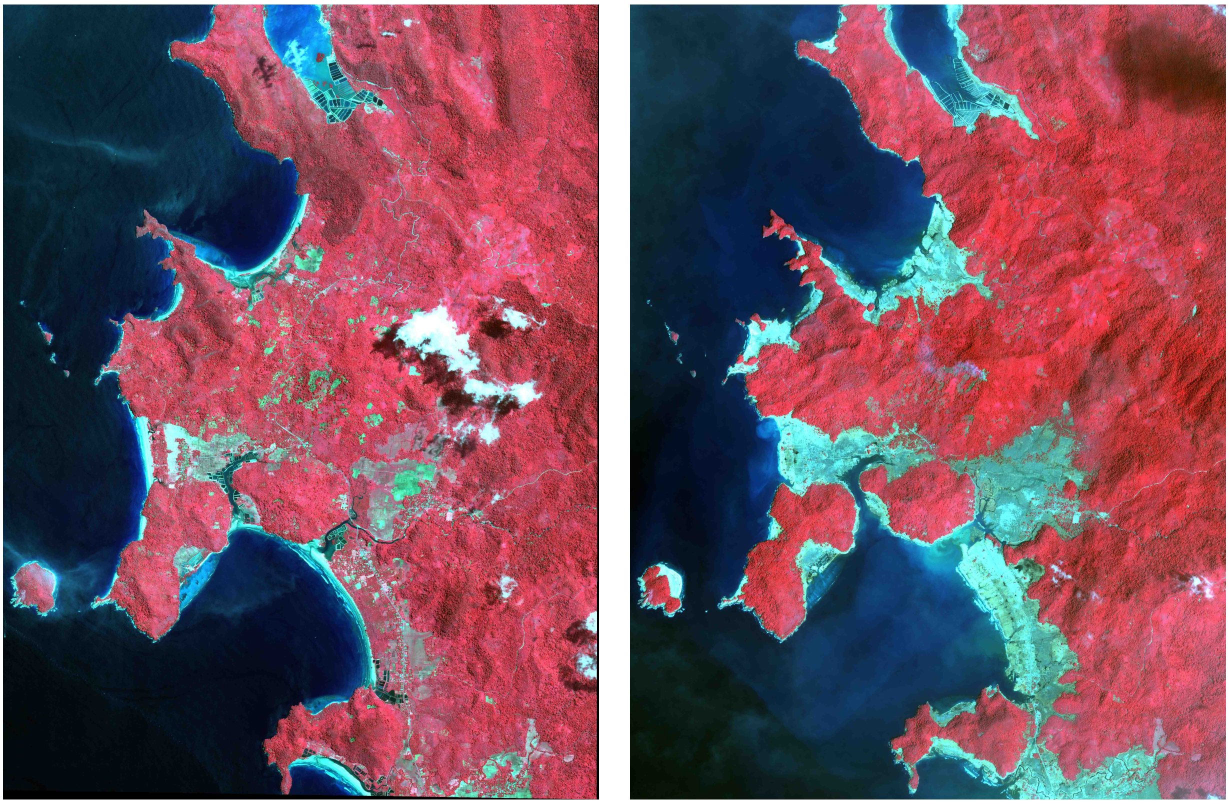

Figure 3: False colours images of Lhonga Leupung area (a) before and (b) after flooding.

7 Classification of remote sensing images

```
FORMATION: Presential
DURATION: 04:00
SEQUENCE: 3
```

7.1 Introduction

The aim of this labwork is to perform the classification of remote sensing images using supervised algorithms. The principle is the same than segmentation. But now the gray level intervals are not defined manually and the definition of a radiometric behavior is not limited to a rectangular area in the spectral domain. Furthermore, since all the computation are done by supervised algorithms, it is possible to use more information than one or two bands and the full multispectral image can be use. In fact, more than one image can be used. In this work, the two *Fabas* images will be classified: first separately and then conjointly.

The OTB proposes various classifiers, each one having different characteristics. In order to train (or learn) the classifier, some labeled pixels should be provided. It is possible to construct the ground-truth (set of labeled pixels) in different ways:

- Using GIS layer and extract the relevant information at the pixel level.
- Do field survey and use GPS to identify pixels.
- Do photo-interpretation when possible.

In this works, the ground-truth is provided as a vector file, see 5. Five classes are considered, they are given in Table 7.

Figure 4: Google view of the impacted area. The red square represents the area of Figure 3.

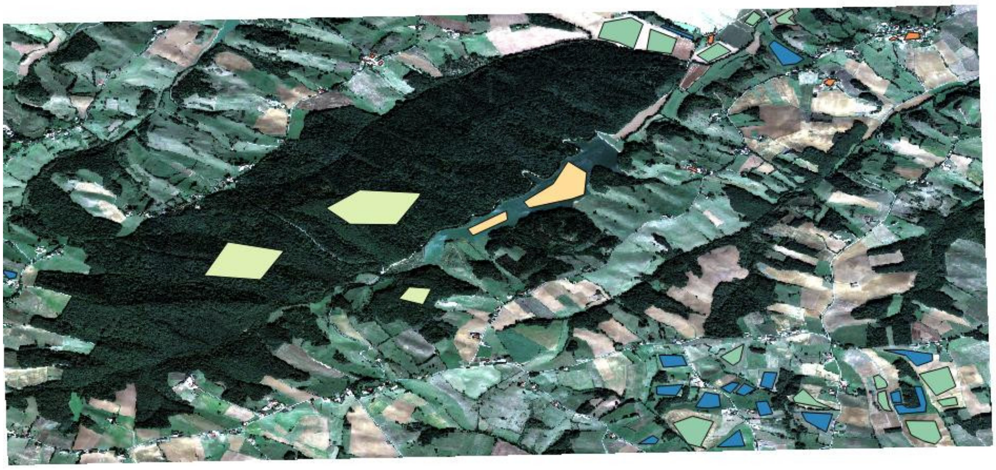

Figure 5: Ground truth for the *Fabas* image.

During this labwork, it is proposed to compare in terms of classification accuracy and processing time some of the classifiers proposed in OTB and all the combination of input data, *i.e.*:

- K-nn, Bayes, SVM and Random Forest.
- The ground-truth being composed of pixels from one date, and two *concatenated* dates.

7.2 Getting started with OTB

There are several steps to do a classification.

1. *Learn the classifier*: It is done with `TrainImagesClassifier`. It takes as inputs, the (set of) remote sensing image(s), the ground-truth (in vector format), and some parameters of the method. To learn the classifier, only the pixels inside the ground-truth are used. After this step, a *model* that contains the parameters is saved. If asked, a confusion matrix is computed.

Table 7: Classes of interest. Numbers corresponding to the attribute in the GIS file is also given.

Classes	Sparse vegetation	Bare soil	Woody vegetation	Water	Built up
Attribute	1	2	3	4	5

2. *Classify the image*: Once the classifier is learned, it is possible to apply the model to all the pixels of the image. It can be done with `ImageClassifier`.

3. Compute the accuracy of the thematic map according to some groundthruth. **This groundthruth should not be spatially correlated with the one used for training**. The confusion matrix can be computed using the function `ComputeConfusionMatrix`.

Work 7.1:

This should be done for one image and one classifier only.

1. Learn the model,
2. Apply the model to classify the entire image,
3. Compute the confusion matrix and save it in a *csv* file.
4. Open the CSV using a spreadsheet. From the confusion matrix, compute the following indices:
   - Global accuracy,
   - Producer accuracy,
   - User accuracy.

7.3 Automatize the process with scripts

It is possible to run directly the *OTB-Applications* from the the command line (on linux-based OS). This way, it is possible to run several operations on one data set or on several data sets automatically.

Works:

1. Write the script to learn the model for all the classification methods and with each date. Each time extract the confusion matrix and compute the global accuracy and the class average accuracy.

2. Report the results on the *collaborative spreadsheet*.

3. For the best method in terms of classification accuracy, discuss about the errors obtained with the confusion matrix.

4. Classify the whole image and compare by visual inspection the errors with what you have inferred from the confusion matrix.

7.4 Multi dates

From the same area, two dates are available. It is possible to use them conjointly in many ways. Two possible solutions are considered here. The first one consider the second date as additional data, *i.e.*, there are twice as many pixels in the training set. For each pixel, we have its reflectance the *[2013-10-12 Sat]* and the *[2013-12-10 Tue]*. The second one is to consider that we have the temporal evolution of the reference.

The first approach can be simply done by providing the two images as inputs to the training function. The classification of the whole images is then done independently (two classification maps). The second approach necessitates to *concatenate* the two dates before training. The concatenation can be done using the function `otbcli_ConcatenateImages`. The classification of the whole image is then done conjointly (only one classification map).

Works:

Table 8: Simulated pixels from two classes

Pixel	Date	Class	B	G	R	IR
$\mathbf{x}_1$	*[2013-10-12 Sat]*	Broadleave	0.30	0.40	0.20	0.80
$\mathbf{x}_1$	*[2013-12-10 Tue]*	Broadleave	0.40	0.45	0.43	0.40
$\mathbf{x}_2$	*[2013-10-12 Sat]*	Conifer	0.29	0.41	0.18	0.75
$\mathbf{x}_2$	*[2013-12-10 Tue]*	Conifer	0.27	0.36	0.30	0.70
$\mathbf{x}_3$	*[2013-10-12 Sat]*	Bare soil	0.39	0.37	0.38	0.39
$\mathbf{x}_3$	*[2013-12-10 Tue]*	Bare soil	0.42	0.44	0.43	0.40

1. Using pixels from Table 8, plot on spreadsheet all pixels according to both approaches. Discuss the advantages and drawbacks of each approach in terms of how it captures the specto-temporal behavior of the different classes.

2. Perform the classification using both approaches, for all the classifiers.

3. Report the results on the *collaborative spreadsheet*.

7.5 Influence of the spatial distribution of the learning samples

```
FORMATION: Non Presential
DURATION: 01:40
SEQUENCE: 4
DAYS: [2016-04-01 Fri 10:20-12:00]
```

Work:

In order to evaluate the influence of the validation samples, you will investigate several reference layers to compute the confusion matrix. Since OTB only select a few samples from all the available one (can be controlled with the options `samples.mt` and `samples.mv`), we need to repeat the experiment several times, to avoid bias.

Select one classifier for all the experiments. You are encouraged to define a shell script … Repeat 20 times the following test

1. Learn with *train_fabas* and compute the confusion matrix with *train_fabas*. Save the confusion matrix for each repetition.

2. Learn with *train_fabas* and compute the confusion matrix with *valid_fabas*. Save the confusion matrix for each repetition.

Compute the average global accuracy and the mean class accuracy and their standard deviation. Discuss about the results.

8 Appendix

8.1 Introduction to shell

This section provides an introduction to *shell* programming and *shell scripts*. A script is a set of commands, which allows to write a processing chain for a given image, or to apply one processing to a set of images. Of course, mixing these two situations is possible. You can find more information easily on the web, a good starting point can be the Wikibook.

Shell is a programming language that is available on all GNU/Linux distributions. It can be used directly from the prompt (interactive mode), or by writing a file with a set of commands to be run. This file should start with the line

```
#!/bin/bash
```

In the following, it is assumed that we are working on the file `script.sh`. To insert comment inside the script, the symbol `#` has to be used.

```
# This is a comment
```

With Linux, a file can be *writable*, *readable* and/or *executable*. To be run as a script, it should be at least *executable* by the OS. It can be by done by running the following command:

```
chmod +x script.sh
```

To run it, just do

```
./script.sh
```

8.1.1 Basic commands

- **cd**: Change repertory. To enter a repertory, do `cd Name_Of_Repertory`.
- **ls**: List all the file in the current repertory.
- **pwd**: Return the name of the current repertory.
- **cp**: Copy a file/repertory, for instance `cp A B`.
- **mv**: Move a file to another, for instance `mv A B`.
- **mkdir**: Create a repertory, `mkdir Name_Of_Repertory`.

For instance, to get all the `tif` files in the current folder:

```
ls *tif
fabas_10_12_2013.tif  fabas_12_10_2013.tif
```

8.1.2 Variables

In shell, a variable is a string (not a number). It can be defined as:

```
var1='Mathieu' # Store "Mathieu" in variables "var1"
var2='Fauvel'
var3='34'
```

Be careful to spaces: there are no spaces, otherwise an error is returned! A variable is displayed using the `echo` function and the variable is accessed with the command `$`.

```
echo $var1 $var2       # print "Mathieu Fauvel"
echo "$var3 ans"       # print "33 ans"
echo '$var3 ans'       # print "$var3 ans"
```

```
Mathieu Fauvel
34 ans
$var3 ans
```

Note the difference between the simple quote `'` and the double quote `"`. The simple quote does not evaluate the variable while the double quote does.

It is possible to pass parameters to the script, solely by adding them when the script is called. They are accessible using the command `$` following by the order number of appearance when the script is called. Let define the `script.sh` file.

```
# ./script.sh Name FamilyName Age
echo $1 $2
echo "J ai $3 ans !"
```

When we do this, we have the following output:

```
chmod +x script.sh
./script.sh Mathieu Fauvel 33
```

```
Mathieu Fauvel
J ai 33 ans !
```

8.1.3 Loop

As in any programming language, loop are very useful to apply a series of processing to several elements of a sequence. The example below applies a processing on all *tif* files of the current directory:

```
for i in *.tif # For all tif file
do
    cp $i ndvi_$i # create a new file and add ndvi_ at the beginning of the filename
done
```

8.1.4 Sequence

It is possible to define sequences of string like this:

```
for name in bayes libsvm knn rf
do
    echo $name
done
```

```
bayes
libsvm
knn
rf
```

Sequences of numbers can be defined like this:

```
for i in `seq 1 5`
do
echo $i
done
```

```
1
2
3
4
5
```