



## 2. Supervised classification of land-cover

### 2.1 Introduction

For this assignment, the requirement is to make a single-date thematic map of some kind using image classification. Learning to do image classification well is extremely important and requires experience. So here is your chance to build some experience! My suggestion is to pick an image or place that you care about for this assignment and also one that you know something about the classes you hope to map. Google Earth may also prove a very useful source of independent information! So give it some thought.

### 2.2 Goals

- Generate a cloud-free image subset that is easier for processing
- Create an appropriate list of land cover classes that can be used to create your classification map
- Create regions of interest (ROIs) for your classes that can be used to train a machine learning algorithm
- Classify your image using a Random Forests classifier

### 2.3 Create file structure

Create the following structure in your work directory using either *Caja* or *MATE Terminal*:

```
2_classification
  0_sourcedata
  1_extract
  2_stack
  3_subset
  4_training
  5_map
```

If you **don't** want use the image you worked on in Module 1 *Introduction*, copy a suitable image from your data directory to "0\_sourcedata" that you just created, extract the bands to "1\_extract" and create a stack in "2\_stack". If you have forgotten how to complete these steps, refer to Module 1 *Introduction*. If you **do** want to use the image you already extracted



and stacked, it is better to create a **symbolic link** to the original file rather than copying it as this will save space. To do this, open *MATE Terminal* and navigate to `/2_classification/0_sourcedata` and type

```
ln -s ../../1_introduction/0_sourcedata/[imagename].tar.gz .
```

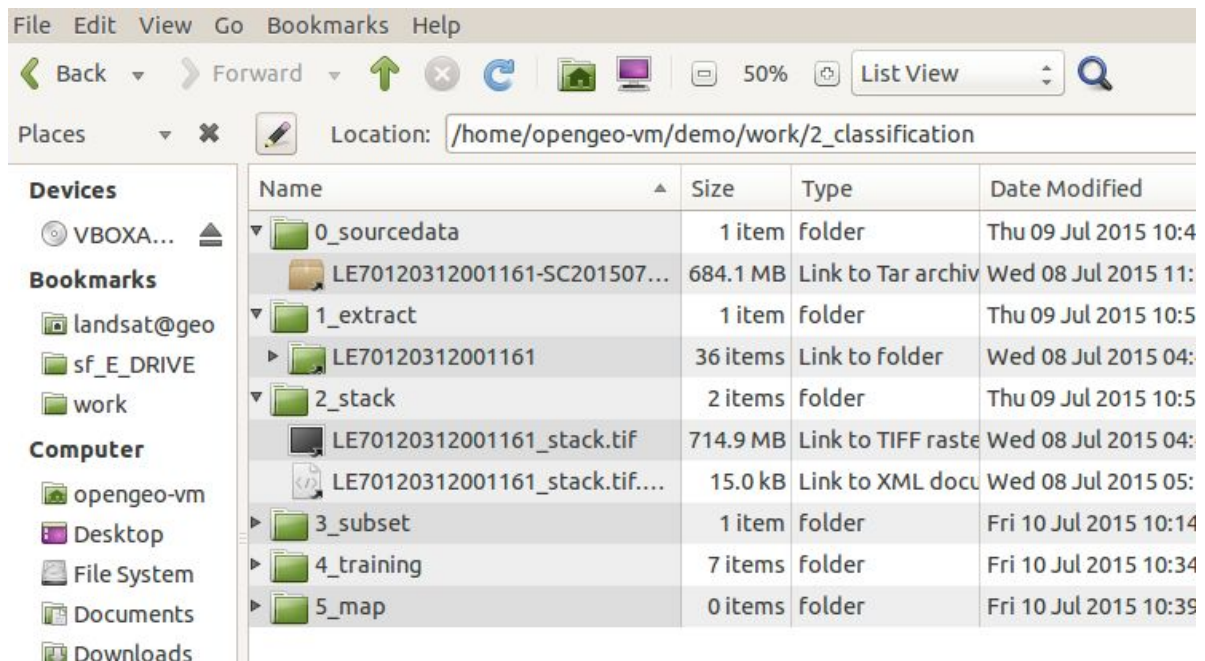
which link the tarball to the current directory as indicated by the dot. If you list the content of the new source data directory in the terminal the name of the tarball should appear in cyan color with an arrow to the original file in red:

```
opengeo-vm@opengeo-vm:~/demo/work/2_classification/0_sourcedata$ ls -l
total 4
lrwxrwxrwx 1 opengeo-vm opengeo-vm 74 Jul  9 10:44
LE70120312001161-SC20150708094915.tar.gz ->
../../1_introduction/0_sourcedata/LE70120312001161-SC20150708094915.tar.gz
```

Redo this step for the extracted and stacked files too. If you need assistance, look at the example below:

```
/work/2_classification/0_sourcedata$ cd ../1_extract/
/work/2_classification/1_extract$ ln -s
../../1_introduction/1_extract/LE70120312001161/ .
/work/2_classification/1_extract$ cd ../2_stack/
/work/2_classification/2_stack$ ln -s ../../1_introduction/2_stack/* .
```

In *Caja*, it should look like below (with the linked files having a black arrow in their icons):



## 2.4 Subset the image

Since the size of the image you will be working on is not relevant in learning the concepts,

the first step in the classification process is creating a subset of one of your images. This will greatly reduce processing time and allow you to redo steps more efficiently. In order to allow you to subset images in the future to the same extent, it makes sense to create a 'mask' of where you want to subset in the form of a shapefile. This way, you can apply this same mask to other images to get matching extents. You can also create a subset by drawing a square on the canvas as illustrated in Module 1 *Introduction*.

Navigate around the image to find a location that contains a good mix of land covers. You do not need to make it too small, for we want to be able to classify diverse land covers and find changes in them. Aiming to make the subset around 1/10<sup>th</sup> the size of the Landsat image might be a good size, but use your judgment. **Reference Subsection S2.1 in Supplementary Module S2 *Methods* for detailed instructions on how to subset an image.**

## 2.5 Generate your ROIs

In this assignment, you will do a supervised classification using regions of interest (ROIs) that you define to train the classifier. The goal of training the classifier is to provide examples of the variety of spectral signatures associated with each class in the map. There are a number of supervised classification algorithms that can be used to assign the pixels in the image to the various map classes. The one you will be using today is called Random Forests.

Any supervised classification methods require prior identification of "training" samples. This can be done in the form of training polygons, which are digitized on an image as Regions of Interest (ROIs). The first thing you need to do is define the legend for the map, or the list of classes to be included. What classes do you want to map? Try to keep it simple, but more interesting than just forest/non-forest – 3-5 might be a good choice. It is also important to have definitions for each of the classes. The lack of clear definitions of the classes can make the resulting maps difficult for others to use.

It is important to take time when collecting your training data. This is very important! My humble opinion is that the largest improvements in mapping the land surface in the future will come from more creative use (or combinations) of inputs as opposed to improved classification algorithms. You can try using multiple dates of data (instead of a single date), or try using texture bands. The possibilities are many and should relate back to the nature of the classes you hope to map. **Reference subsection S2.2 in Supplementary Module S2 *Methods* for detailed instructions on how to create your ROIs.**

## 2.6 Analyze your ROIs

To help you understand why some classes might get confused, you might consider plotting the spectral signatures of your different land classes. Notice a few things in the signatures of different classes. First, some classes are more distinct spectrally than others. For example, water is consistently dark in the NIR and MIR wavelengths, much darker than the other classes. So it shouldn't be difficult to classify water correctly. Also, not all pixels in the same

classes are the same! I hope this will help you begin to understand the inherent variability of the classes.

Another factor that will strongly influence the results of your classification is the map class variance. For example, classes with high variance will tend to include more places in the map than you'd like. Similarly, if you use only very small and limited sites for a class, you may get extremely low variances and the class will be underrepresented in your map.

**Reference subsection S2.3 in Supplementary Module S2 *Methods* for detailed instructions on how to analyze your ROIs.**

## 2.7 Train Your Classifier

One way of performing a supervised classification is to utilize a Machine Learning algorithm. Machine Learning algorithms utilize training data to efficiently “learn” how to classify pixels. Using ROIs, these algorithms can “train” a classifier, and then use this to classify the rest of the pixels in the map.

One Machine Learning algorithm that is particularly popular in remote sensing is Random Forests. A Random Forests algorithm creates numerous decision trees for each pixel. Each of these decision trees “votes” on what the pixel should be classified as. The classification that receives the most votes is then assigned as the map class for that pixel. Random Forests are efficient on large data and highly accurate when compared to other classification algorithms.

To do our classification we are going to use a Random Forest classifier utilizing the Orfeo toolbox in QGIS. But first, you must train the classifier using the regions of interest you just collected. **Reference subsection S2.4 in Supplementary Module S2 *Methods* for detailed instructions on how to train your classifier.**

## 2.8 Classify Your Image

You should now have a .txt file saved that contains all the information needed to classify your image. Now the Random Forest classifier can use the information in this model and apply it to the rest of the pixels in the image. **Reference subsection S2.5 in Supplementary Module S2 *Methods* for detailed instructions on how to classify your map.**

## 2.9 Examine Your Map

Now your classified image should be displayed as a layer in QGIS. Each different class will correspond to the number you gave that class when creating the ROIs. Look around at the classified image- are you happy with the results? If not (and you won't be the first time), you need to revisit your training data. Some classes are bound to be “too big” and others “too small”. If a class is too small, it will help to add a few more training sites for better characterization of the inherent variability of the class. As a rule of thumb, if you have lots

of pixels in a class, it will tend to have higher variances, and tend to be “too big” in your results. You will need to spend quite a bit of time looking over your results and modifying your sites to get a good result. One common problem is that there may be land covers in the image that you haven’t included in your legend, and those areas will be poorly classified. So you might need to add a class or two.

## 2.10 Smoothing The Classification Results

When examining your map, you may notice a “salt and pepper” effect in some of your classes. This refers to lone pixels of a class inside a large otherwise homogenous class that act to make the map look “noisy”. Chances are, these pixels should be classified as the majority class that surrounds it. If this is occurring in your map and you are confident these pixels are the result of classification errors then they can easily be removed. A technique often used in remote sensing is to ‘sieve’ isolated pixels and replace them with the classification of the majority class that surrounds it. **Reference subsection S2.6 in Supplementary Module S2 *Methods* for instruction on cleaning up the classification map.**

## 2.11 OPTIONAL: Object-Based Classification

One additional step you can do in the classification process is to classify on segments instead of individual pixels. This can be advantageous if you are finding your classification image to be overly “noisy”. Image segmentation algorithms look for clusters of similar pixels in an image and group them together based on pre-determined criteria. In doing so, you lose the spatial resolution of the image but can create a simpler image that’s easier to work with. In addition, the shape, size, and spectral variance of a segment can aid in the classification process. The clusters reveal changes at the landscape level, which is why segmentation is often used in both manual interpretation and automated algorithms for change detection. There are numerous different algorithms for image segmentation, but the one we are going to use (which is one of the most commonly used techniques) is called Mean-Shift Clustering.

Mean-shift analysis is a non-parametric clustering technique that is widely used in image processing. The algorithm assumes that the spectral values in the image are sampled from an underlying probability density function. The dense clusters of data are therefore assumed to correspond with the modes of the underlying density function. The algorithm makes no assumption about the number of modes or size of output clusters.

While the segmentation process can be computationally intensive, it can reveal additional information about the landscape that is not present at the pixel level. Primarily, it allows spectral variance to be utilized in the classification process. The output of the segmentation described in this tutorial is an image that is made up of segments determined by the Mean-Shift Algorithm. The segments contain both the mean and the variance of the pixels that they contain. This image can then be classified in the same way described in the

previous sections. **Reference subsection S2.7 in Supplementary Module S2 *Methods* for detailed instructions on how to perform Mean-Shift Clustering and classification on your image.**

## **2.11 OPTIONAL: Map Filtering Through Segmentation**

In step **2.10** you sieved out isolated groups of classified pixels and replaced them with the group majority of the pixels around it. This made the map appear less noisy while still retaining the geometry of the features in the map. An alternative method to cleaning up a map is to use image segmentation to assign classifications at the segment level instead of at the pixel level. In step **2.11** zonal statistics were used to gain information about the segments and perform object-based classification. This is not the only way segmentation can be utilized in classification, however.

An alternative to classifying objects based on the original Landsat data is to classify the map at the pixel level and use the segmentation results to clean up that classification. In this process, no means or variances are calculated. Instead, each pixel is classified normally. The mode, or most likely classification, for each segment is then calculated. These modes are then assigned to each segment and used at the output classification map. **Reference subsection S2.8 in Supplementary Module S2 *Methods* for detailed instructions on map filtering using Mean-Shift segmentation.**