Fifth GFOI Regional Technical Workshop: Accuracy Assessment

Santa Rosa, Philippines

BEEODA: a suite of open-source software and educational materials for processing Earth Observation data

P. Olofsson

BOSTON UNIVERSITY

Why are we doing this?

- Our own research needs!
- Built in support of our own research and education, and capacity building organizations/initiatives (SilvaCarbon, GFOI, GOFC/GOLD)
- Lesson learned from teaching workshops: **software to take home and that are in line with new advancements needed**

Remote sensing advancements: time series analysis, OBIA, statistical inference, etc.

ADVANCEMENTS

BEEODA: free open-source software and educational materials for EO analysis

SOFTWARE

Capacity building needs in tropical countries; SilvaCarbon, NASA, FAO, etc.

CAPACITY BUILDING

Vast Landsat archive, lots of new imagery recently added through LGAC; + high resolution data

EARTH OBS. DATA

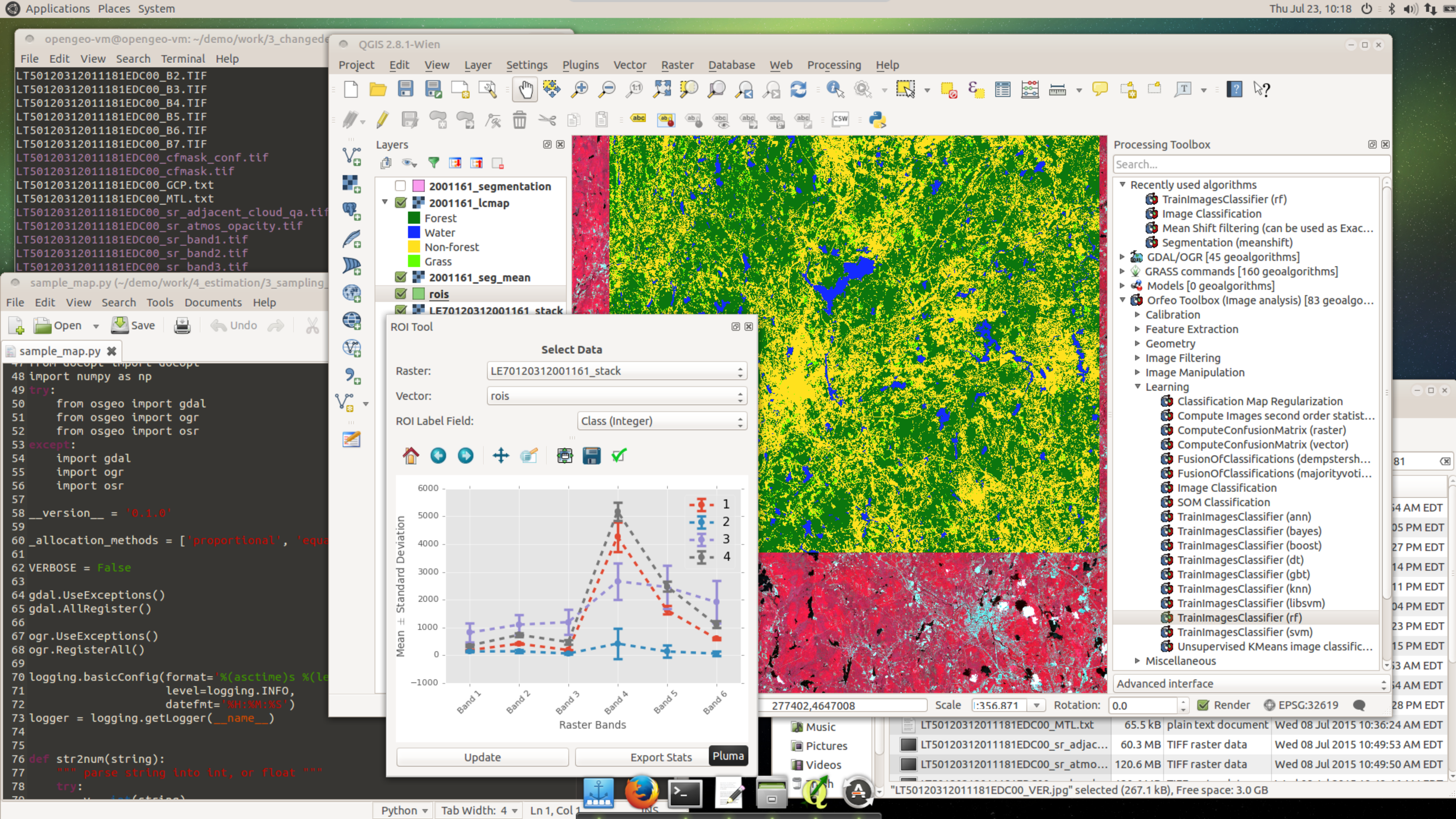
BOSTON UNIVERSITY

Overview

- BEEODA: *Boston Education in Earth Observation Analysis*
- Collection of: existing open-source tools +
  our own implementations +
  educational materials
- Philosophy: “*try to use as much existing tools as possible and only implement the missing pieces*”
- Educational material divided into modules in line with demand and advancements in remote sensing analyses such as time series analysis and accuracy/area estimation

Technical aspects

- Runs as a virtual machine in Windows, OS X, Solaris and Linux platforms
- *Oracle VM VirtualBox* (freeware) is the only software required
- Can also be run from a USB stick with no additional software needed
- In addition to homemade tools, VM includes *QGIS*, *GDAL*, *Orfeo*, *R*, *Python*, *Git* in a Linux Ubuntu operating system


Applications Places System
Thu Jul 23, 10:18
opengeo-vm@opengeo-vm: ~/demo/work/3_changede
File Edit View Search Terminal Help
LT50120312011181EDC00_B2.TIF
LT50120312011181EDC00_B3.TIF
LT50120312011181EDC00_B4.TIF
LT50120312011181EDC00_B5.TIF
LT50120312011181EDC00_B6.TIF
LT50120312011181EDC00_B7.TIF
LT50120312011181EDC00_cfmask_conf.tif
LT50120312011181EDC00_cfmask.tif
LT50120312011181EDC00_GCP.txt
LT50120312011181EDC00_MTL.txt
LT50120312011181EDC00_sr_adjacent_cloud_qa.tif
LT50120312011181EDC00_sr_atmos_opacity.tif
LT50120312011181EDC00_sr_band1.tif
LT50120312011181EDC00_sr_band2.tif
LT50120312011181EDC00_sr_band3.tif
sample_map.py (~/demo/work/4_estimation/3_sampling
File Edit View Search Tools Documents Help
Open Save Undo
sample_map.py
48 import numpy as np
49 try:
50 from osgeo import gdal
51 from osgeo import ogr
52 from osgeo import osr
53 except:
54 import gdal
55 import ogr
56 import osr
58 __version__ = '0.1.0'
60 _allocation_methods = ['proportional', 'equa
62 VERBOSE = False
64 gdal.UseExceptions()
65 gdal.AllRegister()
67 ogr.UseExceptions()
68 ogr.RegisterAll()
70 logging.basicConfig(format='%(asctime)s %(le
71 level=logging.INFO,
72 datefmt='%H:%M:%S')
73 logger = logging.getLogger(__name__)
76 def str2num(string):
77 """ parse string into int, or float """
78 try:
Python Tab Width: 4 Ln 1, Col 1 INS
QGIS 2.8.1-Wien
Project Edit View Layer Settings Plugins Vector Raster Database Web Processing Help
Layers
2001161_segmentation
2001161_lcmap
Forest
Water
Non-forest
Grass
2001161_seg_mean
rois
LE70120312001161_stack
ROI Tool
Select Data
Raster: LE70120312001161_stack
Vector: rois
ROI Label Field: Class (Integer)
Mean ± Standard Deviation
Raster Bands
Band 1
Band 2
Band 3
Band 4
Band 5
Band 6
6000
5000
4000
3000
2000
1000
0
−1000
1
2
3
4
Update
Export Stats
Pluma
277402,4647008
Scale 1:356,871
Rotation: 0.0
Render
EPSG:32619
Processing Toolbox
Search...
Recently used algorithms
TrainImagesClassifier (rf)
Image Classification
Mean Shift filtering (can be used as Exac...
Segmentation (meanshift)
GDAL/OGR [45 geoalgorithms]
GRASS commands [160 geoalgorithms]
Models [0 geoalgorithms]
Orfeo Toolbox (Image analysis) [83 geoalgo...
Calibration
Feature Extraction
Geometry
Image Filtering
Image Manipulation
Learning
Classification Map Regularization
Compute Images second order statist...
ComputeConfusionMatrix (raster)
ComputeConfusionMatrix (vector)
FusionOfClassifications (dempstersh...
FusionOfClassifications (majorityvoti...
Image Classification
SOM Classification
TrainImagesClassifier (ann)
TrainImagesClassifier (bayes)
TrainImagesClassifier (boost)
TrainImagesClassifier (dt)
TrainImagesClassifier (gbt)
TrainImagesClassifier (knn)
TrainImagesClassifier (libsvm)
TrainImagesClassifier (rf)
TrainImagesClassifier (svm)
Unsupervised KMeans image classific...
Miscellaneous
Advanced interface
Music
Pictures
Videos
LT50120312011181EDC00_MTL.txt 65.5 kB plain text document Wed 08 Jul 2015 10:36:24 AM EDT
LT50120312011181EDC00_sr_adjac... 60.3 MB TIFF raster data Wed 08 Jul 2015 10:49:53 AM EDT
LT50120312011181EDC00_sr_atmo... 120.6 MB TIFF raster data Wed 08 Jul 2015 10:49:50 AM EDT
"LT50120312011181EDC00_VER.jpg" selected (267.1 kB), Free space: 3.0 GB

Modules

- Modules consist of
  - step-by-step instructions
  - relevant literature
  - recorded demos and lectures

BEEODA

beeoda.org/

Search

Home About Services Modules Contact BEEODA Team

Open Source Modules

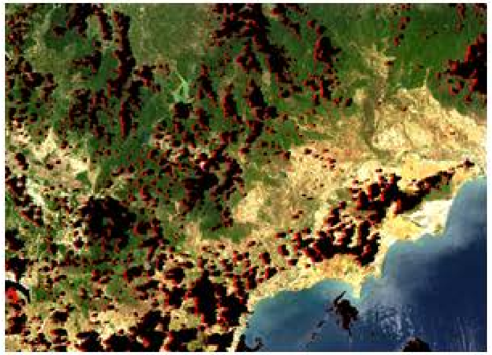
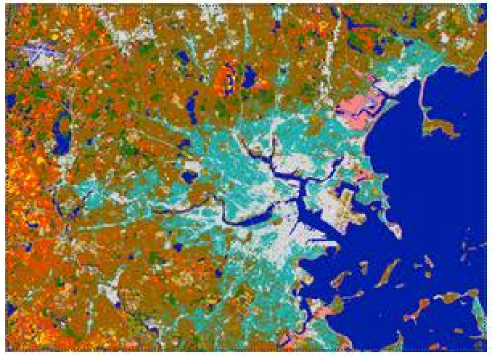
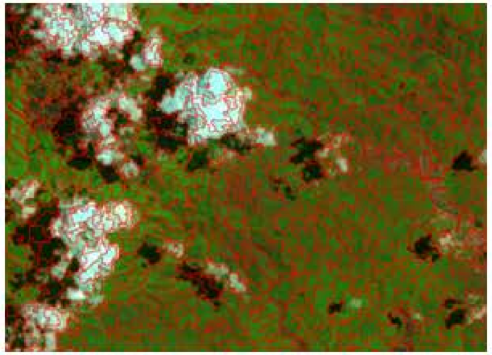


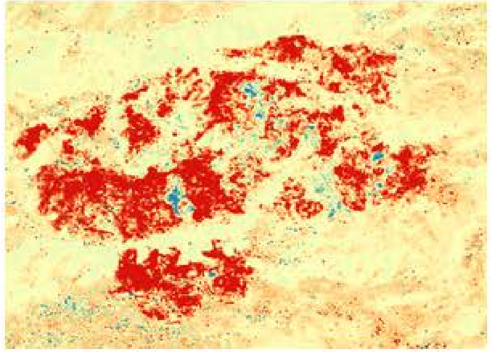
Image Pre-Processing
Downloading, atmospheric correction, etc



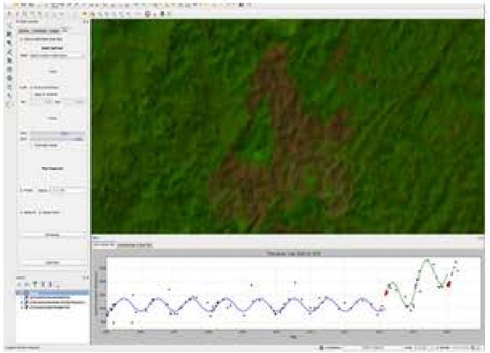
Classification
Random forest, etc



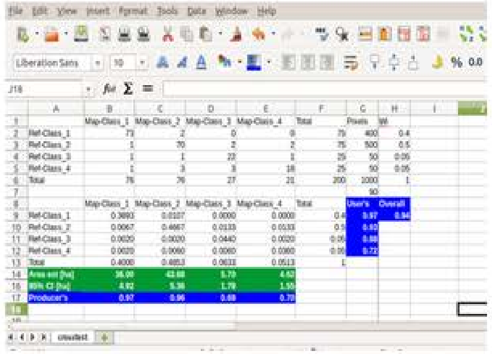
Object-Based Analysis
Segmentation and Classification



Change Detection
Direct Classification, PCA, MAD, etc



Time Series Analysis
CCDC, YATSM, etc



Area and Accuracy Estimation
Statistical inference

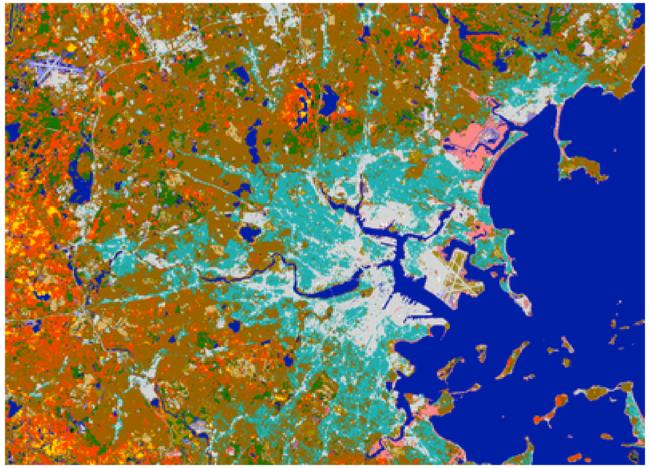
BOSTON UNIVERSITY

BEEODA

beeoda.org/

Search

Classification



Topics to do with classification include:

- Gathering training data appropriate for study
- Using ancillary data to train machine learning algorithms
- Creating maps with user-defined classes
- Interactively tweeking training data and comparing classification algorithms to perfect map
- Cleaning up map to remove isolated pixels

View the tutorial at: GitHub.

Date: July 2015

✖ Close Project

Pontus Olofsson, Christopher E. Holden, Eric L. Bullock
Boston Education in Earth Observation Data Analysis/
Department of Earth & Environment, Boston University

BOSTON UNIVERSITY

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

a **symbolic link** to the original file rather than copying it
open *MATE Terminal* and navigate to
nd type

```
ion/0_sourcedata/[imagename].tar.gz .
```

nt directory as indicated by the dot. If you list the content
in the terminal the name of the tarball should appear in
riginal file in red:

```
o/work/2_classification/0_sourcedata$ ls -l

ngeo-vm 74 Jul  9 10:44
094915.tar.gz ->
cedata/LE70120312001161-SC20150708094915.tar.gz
```

nd stacked files too. If you need assistance, look at the

```
urcedata$ cd ../1_extract/
tract$ ln -s
act/LE70120312001161/ .
tract$ cd ../2_stack/
ack$ ln -s ../../1_introduction/2_stack/* .
```

(with the linked files having a black arrow in their icons):

elp
50% List View
/home/opengeo-vm/demo/work/2_classification

	Size	Type	Date Modified
ata	1 item	Folder	Thu 09 Jul 2015 10:4
12001161-SC201507...	684.1 MB	Link to Tar archiv	Wed 08 Jul 2015 11:
	1 item	Folder	Thu 09 Jul 2015 10:5
12001161	36 items	Link to Folder	Wed 08 Jul 2015 04:
	2 items	Folder	Thu 09 Jul 2015 10:5
12001161_stack.tif	714.9 MB	Link to TIFF raste	Wed 08 Jul 2015 04:
12001161_stack.tif....	15.0 kB	Link to XML docu	Wed 08 Jul 2015 05:
	1 item	Folder	Fri 10 Jul 2015 10:14
	7 items	Folder	Fri 10 Jul 2015 10:34
	0 items	Folder	Fri 10 Jul 2015 10:35

ll be working on is not relevant in learning the concepts,

process is creating a subset of one of your images. This will
nd allow you to redo steps more efficiently. In order to
future to the same extent, it makes sense to create a
et in the form of a shapefile. This way, you can apply this
matching extents. You can also create a subset by drawing
ted in Module 1 *Introduction*.
find a location that contains a good mix of land covers.
nall, for we want to be able to classify diverse land covers
to make the subset around 1/10th the size of the Landsat
se your judgment. **Reference Subsection S2.1 in**
***hods* for detailed instructions on how to subset an**

upervised classification using regions of interest (ROIs)
fier. The goal of training the classifier is to provide
l signatures associated with each class in the map. There
fication algorithms that can be used to assign the pixels in
ses. The one you will be using today is called Random

methods require prior identification of "training" samples.
aining polygons, which are digitized on an image as
st thing you need to do is define the legend for the map, or
What classes do you want to map? Try to keep it simple,
est/non-forest – 3-5 might be a good choice. It is also
each of the classes. The lack of clear definitions of the
aps difficult for others to use.
hen collecting your training data. This is very important!
gest improvements in mapping the land surface in the
ive use (or combinations) of inputs as opposed to
s. You can try using multiple dates of data (instead of a
bands. The possibilities are many and should relate back to
e to map. **Reference subsection S2.2 in Supplementary**
l instructions on how to create your ROIs.

e classes might get confused, you might consider plotting
fferent land classes. Notice a few things in the signatures of
es are more distinct spectrally than others. For example,
NIR and MIR wavelengths, much darker than the other
to classify water correctly. Also, not all pixels in the same

will help you begin to understand the inherent variability

gly influence the results of your classification is the map
es with high variance will tend to include more places in
, if you use only very small and limited sites for a class,
nces and the class will be underrepresented in your map.
upplementary Module S2 *Methods* for detailed
your ROIs.

sed classification is to utilize a Machine Learning
orithms utilize training data to efficiently "learn" how to
algorithms can "train" a classifier, and then use this to
e map.
thm that is particularly popular in remote sensing is
sts algorithm creates numerous decision trees for each
"votes" on what the pixel should be classified as. The
st votes is then assigned as the map class for that pixel.
large data and highly accurate when compared to other

e going to use a Random Forest classifier utilizing the
ou must train the classifier using the regions of interest
bsection S2.4 in Supplementary Module S2 *Methods* for
train your classifier.

ved that contains all the information needed to classify
est classifier can used the information in this model and
the image. **Reference subsection S2.5 in**
***hods* for detailed instructions on how to classify your**

be displayed as a layer in QGIS. Each different class will
ve that class when creating the ROIs. Look around at the
with the results? If not (and you won't be the first time),
data. Some classes are bound to be "too big" and others
it will help to add a few more training sit r bet
ariability of the class. As a rule of thumb, if u h e lo

BOSTON UNIVERSITY

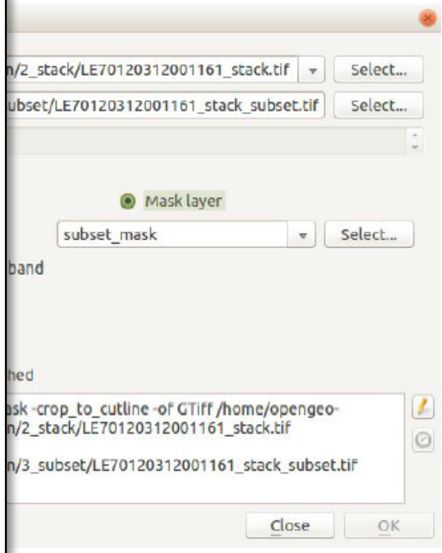
Pontus Olofsson, Christopher E. Holden, Eric L. Bullock
Boston Education in Earth Observation Data Analysis/
Department of Earth & Environment, Boston University

BOSTON UNIVERSITY

S2. Methods: Classification

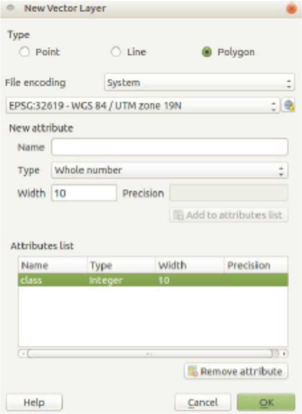
S2.1 Subset an image using *Mask layer*

1. Display the raster image that you want to subset in addition to the subset produced in the **Introduction.** If you have not created a single subset yet, create one by selecting *Extent* and *Clipping mode* and simply drawing a rectangle on the image canvas by clicking the mouse. This is detailed in **Tutorial 1 *Introduction* subsection 1.8.**
2. The subset raster will be used to create a 'clipper' vector. This vector can be used to clip all future images to the same extent. This is necessary for much of the analysis done in this tutorial.
3. Go to *Processing* > *Toolbox.* In the *Processing Toolbox* go to *Orfeo Toolbox* > *Geometry* > *Image Envelope.*
4. For the input image select the subset you made in the **Introduction.**
5. For *Output Vector Data* choose a filename and save it to your "/datasets" folder. This belongs here because you will likely use this file in multiple exercises.
6. Click *Run.*
7. To extract the subset using the vector layer, go to *Raster* > *Extraction* > *Clipper.* For *Input file (raster)* put the Landsat image you want to subset. For *Output file* click *Select…* and save your output as "[imagename]_stack_subset". For the **Introduction** exercise, it is recommended saving the subsets into the image subfolders in "5_composite".
8. Under *Clipping mode* select *Mask layer* and select the shapefile you just created. Click *OK.* (See screenshot below.)



to easily work
he previous
riate 3-band

eed to create a
esponding to
id before, go to
Shapefile Layer.
ecify the same
as the image
dd an attribute
me 'Class', and
can delete the
te List by
n the name 'id'



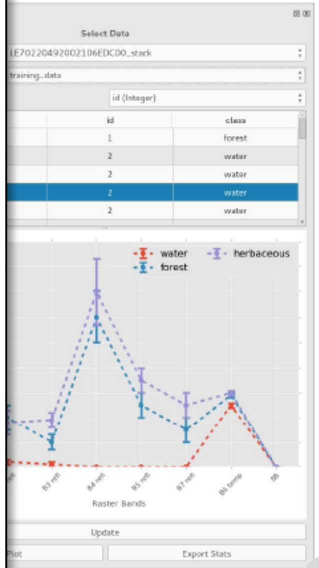
e. Your box should be filled out like the screenshot above.
ile. It will ask you where to save the ROI. Save it to
".
ight clicking the newly created shapefile in the *Layers*
the *Add Feature* button (or *Layers* > *Add Feature*); left click
on, and right click to complete the polygon.
ROI, it will ask you to fill in a
n your map will have a
u have 5 classes, then your
ake sure to keep track of
to which class. To the right is
ater class.
for the class, click *OK.* Repeat
lass. **When collecting ROIs,**
below.
ROIs, right click on your
and click *Toggle Editing* to
dits.

he polygon that represents the same class. In fact, you should
ch class! Many small ROIs are better than a few large ones (a
OI). This can help account for the spectral variations in each land
example of forest, it will be hard to recognize the variety of
that class. If you have multiple sub-classes under the same class
ing considered forest), your ROIs should reflect the full breath

need to adjust your training data a number of times (maybe
good quality classification. The actual classification part is just
accuracy of your results is the quality of your training data.
ollecting ROIs.

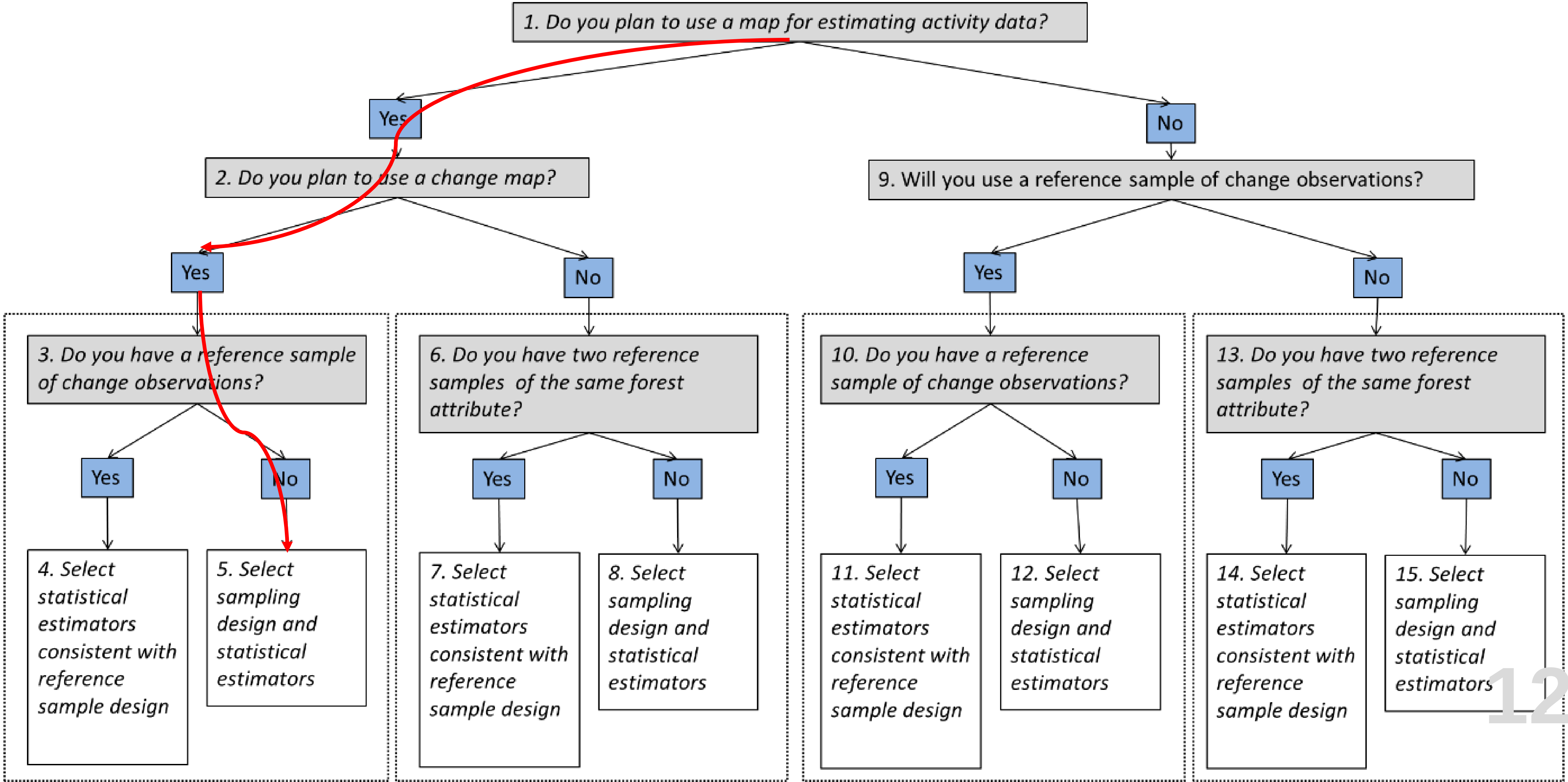
ch class should be proportional to the total area of that class in
than water, then you should have more ROIs for forest than you

atures of your training data ROIs, load the *ROI Explorer*
Plugins menu and click *Manage and Install Plugins.* Under
ROI Explorer plugin. If this plugin is not currently installed,
m the *Available* tab.
Explorer plugin should appear docked in the right side of
d not appear, right click on any QGIS toolbar and show the
ROI Explorer dropdown item.
ce tracks all raster and vector data opened within QGIS.
own box the raster image you want to use for your
r drop down box, select the vector layer containing your
est. Use the *ROI Label Field* drop down box to select the
aining data vector file that contains the training data labels.
e field that is used to aggregate features into class labels.

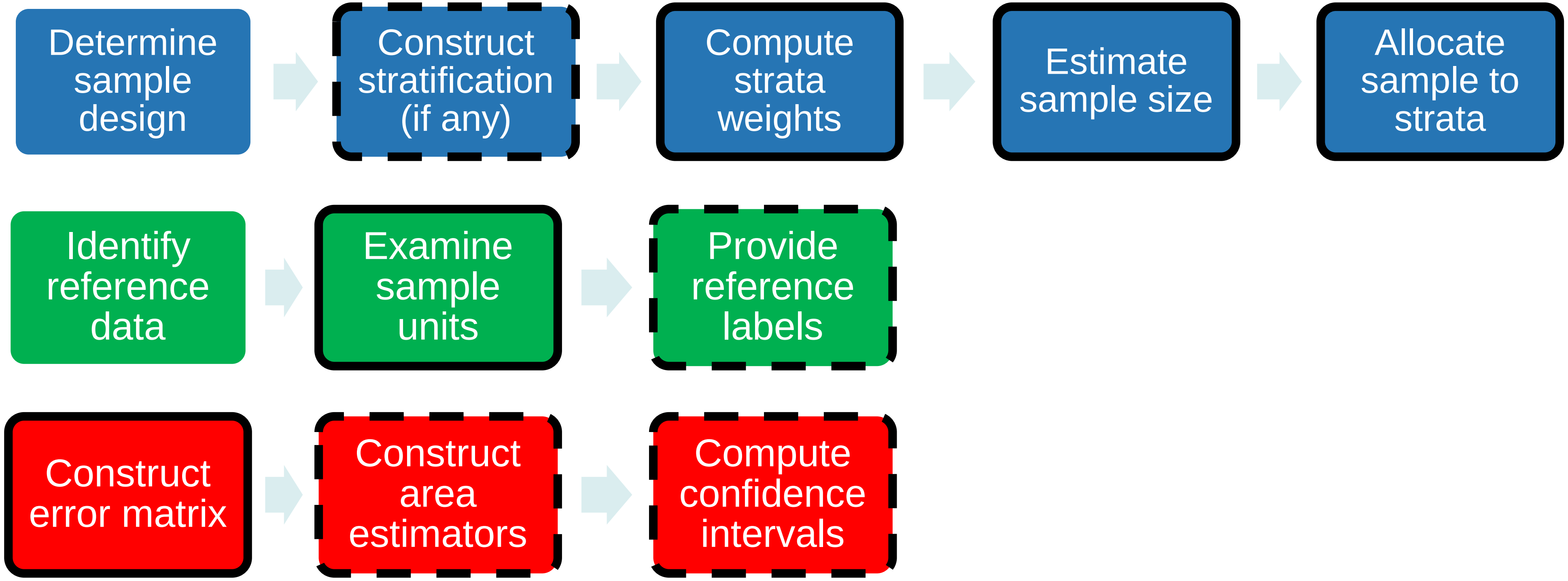


BOSTON UNIVERSITY

GFOI Methods & Guidance, inference of AD



Workflow, decision tree "red thread"



Own implementation

Existing tools (QGIS/GDAL)

BOSTON UNIVERSITY

Pontus Olofsson, Christopher E. Holden, Eric L. Bullock
Boston Education in Earth Observation Data Analysis/
Department of Earth & Environment, Boston University

4. Estimation

4.1 Introduction

In this Module we select sample of reference observations of the study area with the aim of estimating the area of forest change. We will also use the sample for estimating accuracy of
the map and map classes. If stratifying, a
instructions will refer to the land cover c
Module, *Change Detection*. The basic idea
land cover (or change in land cover) are
which are identified by comparing the m
estimates and accuracy are then inferred
three main steps: 1) design and selection
sample and decision of agreement of refe
sample.

4.2 Sampling design

The sampling design is the protocol for s
segments) that will form the basis of the
that the sampling design is a probability
randomization in the selection protocol a
such that the inclusion probability is kno
sample. A variety of probability sampling
used designs being simple random, strati
choosing a design, three main decisions a
and whether to use a systematic or simpl
cluster sampling is to reduce the cost of
high resolution data need to be collected
allow for collection only for the primary
(cluster designs as defined in this text in
is recommended only if cost savings or p
more complex analysis and because the p
(i.e., intracluster correlation) often redu
equal size. The use of strata is usually mo
small proportion of the total map and if t

might be required to implement the analysis. A stratified design is therefore usually a good choice, especially if the aim is to estimate land cover change.

That map that was created in Module 3 contained a certain number of classes (including *forest cover loss* and *gain*) but the theory and methodology is generic and could be applied to any thematic map regardless of how the map was made and regardless of the nature and number of map categories. As the aim is to estimate the area of forest change, it is recommended to use the map classes as strata. This will ensure that a sufficient sample size for estimation can be allocated to the change classes.

After settling on a sampling design -- stratified random in this case -- we need to determine the total sample size and allocation of the sample to strata. Please refer to **Subsection S4.1.1 in Supplementary Module S4, *Methods: Estimation*.** to complete this step. After sample size and allocation have been determined, the sample needs to be selected. This can be done in several ways but many software lack good support for a selecting sample. Therefore, we have written our own program: usage instructions are provided in **Subsection S4.1.2 in Supplementary Module S4, *Methods: Estimation*.**

4.3 Response design

Once we designed the sample and a stratified random sample is selected, it needs to be interpreted using a suitable source of reference data and we need to decide if the map and reference observations agree. This step is referred to as the response design.

First, we need to identify the reference data sources. Ideally, we would have plots revisited in the field but this is rarely attainable so we will need to collect reference observations by careful examination of the sample units in satellite data. The more data we have at our disposal the better. If you have no additional data you can use the Landsat data for collecting reference observations but the *process has to be more accurate than the process used to create the map being evaluated*. Careful manual examination can be regarded as being a more accurate process than automated classification. In addition to Landsat data, you can use whatever data available in Google Earth™. Please refer to **Subsection S4.2.1 in Supplementary Module S4, *Methods: Estimation*** for instructions for preparing the sample for interpretation including export of the sample to Google Earth™. As the estimates are based on the sample, **it is important that the labels are correct** and it is recommended that **three interpreters** examine each unit independently.

Once the sample has been interpreted, the agreement between map and reference labels needs to decided; this could potentially be a complicated task but in this case we are using the map classes as strata which makes the decision straightforward. The agreement is preferably expressed in the form of an **error matrix**, which is a simple cross-tabulation of the map labels against the reference labels for the sample units. The error matrix organizes the acquired sample data in a way that summarizes key results and aids the quantification of accuracy and area. The main diagonal of the error matrix highlights correct classifications while the off-diagonal elements show omission and commission errors. The cell entries and marginal values of the error matrix are fundamental to both accuracy assessment and area

Pontus Olofsson, Christopher E. Holden, Eric L. Bullock
Boston Education in Earth Observation Data Analysis/
Department of Earth & Environment, Boston University

S4. Methods: Estimation

S4.1 Sample design

S4.1.1 Determine sample size and allocation

1. Display your map in QGIS by clicking L
2. Color it if you haven't already: right-cl
   > *Style*; set *Render Type* to *Singleband*
   set values to 1-7, and give each catego
3. Determine the areas of each map categ
   and type: `gdalinfo -hist strat`
4. This gives the number of pixels of each
   gdalinfo gives the following areas in pi

	Non-forest	*Forest*	*Wat*
Area	47,996	228,551	13,7
W_i	16.3%	77.6%	4.6%

5. To determine the sample size for a str
   Cochran (1977): $n \approx \left(\frac{\sum W_i S_i}{S(\hat{P})}\right)^2$ where W
   error for stratum i; the latter is estima
   forest loss in stratum i. $S(\hat{P})$ is the targ
   assuming one error of omission of for
   user's accuracy of 0.8 and a target stan
   a confidence interval of 1%); we get fo
   size:

	Non-forest	*Forest*	*Wat*
p_i	0.01	0.01	0
S_i	0.099	0.099	0.00
$S(\hat{P})$			

This in turn gives: $n \approx \left(\frac{\sum W_i S_i}{S(\hat{P})}\right)^2 = \left(\frac{0.098}{0.005}\right)^2 = 387$ (note that this is just an example and users need to specify their own target errors and expected accuracy and omission errors).

6. The second step is to determine how to allocate these units to strata. Good practices stipulate that 50, 75 or 100 units are allocated to the smaller classes depending on the total sample size and that the rest is proportionally allocated to the larger strata. In this all strata are small relative forest and the sample is allocated to strata as (*Forest gain* and *Forest gain/loss* are so small fractions of the map that we will not attempt to estimate them; if they were larger they would be included):

	Non-forest	*Forest*	*Water*	*Forest loss*	*Forest gain*	*For. loss/gain*
n_i	55	230	50	50	0	0

S4.1.2 Select sample

1. QGIS does not have built-in tools for drawing samples (this hold true also for most proprietary software) so we need to use of Python script: copy the "sample_map.py" and "docopt.py" from *Desktop* to your working directory (or download from https://github.com/ceholden/accuracy_sampler/tree/master/script and https://github.com/docopt/docopt); make sure both files are stored in the same folder.
2. If not using the Virtual Machine but a Windows operating system and *OS4Geo*, click the Windows Start button > *QGIS* > *OSGeo4W Shell*; in the terminal, navigate to the working directory. In the Virtual Machine, open a terminal. Type `python sample_map.py -h` and read about the different options.
3. To select a stratified random sample, type: `python sample_map.py -v -mask 0 --size 385 --allocation "55 230 50 50 0 0" --vector sample.shp stratified stratification_massachusetts_utm.tif`
4. This will create a shapefile "sample.shp" that contains the sample. **Note:** if the script halts with the message "MemoryError", the memory allocation when starting the Virtual Machine needs to be increased (in *Oracle VirtualBox Manager: Settings > System >* increase *Base Memory* before launching the VM).

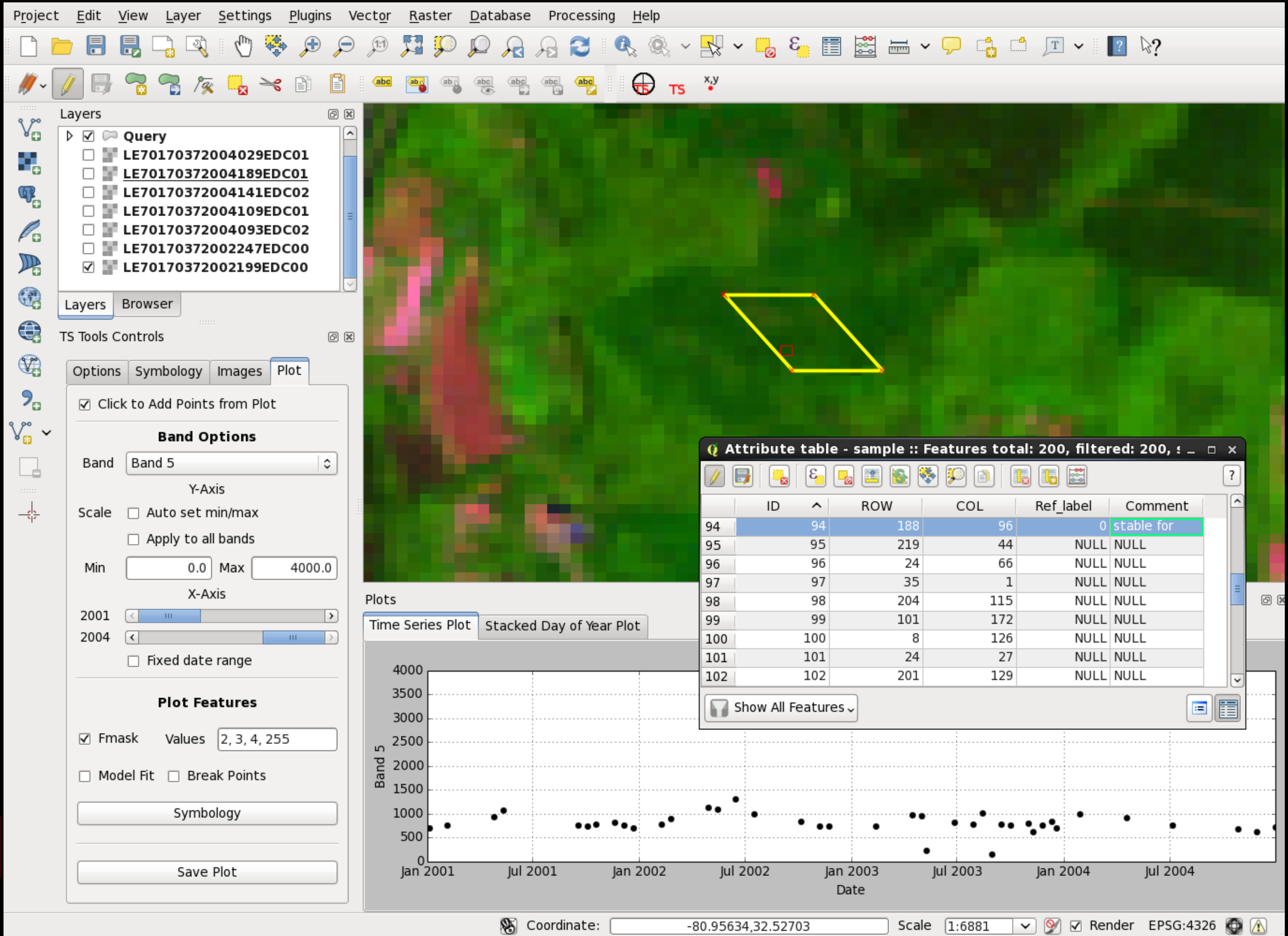
S4.2 Response Design

S4.2.1 Interpreting sample

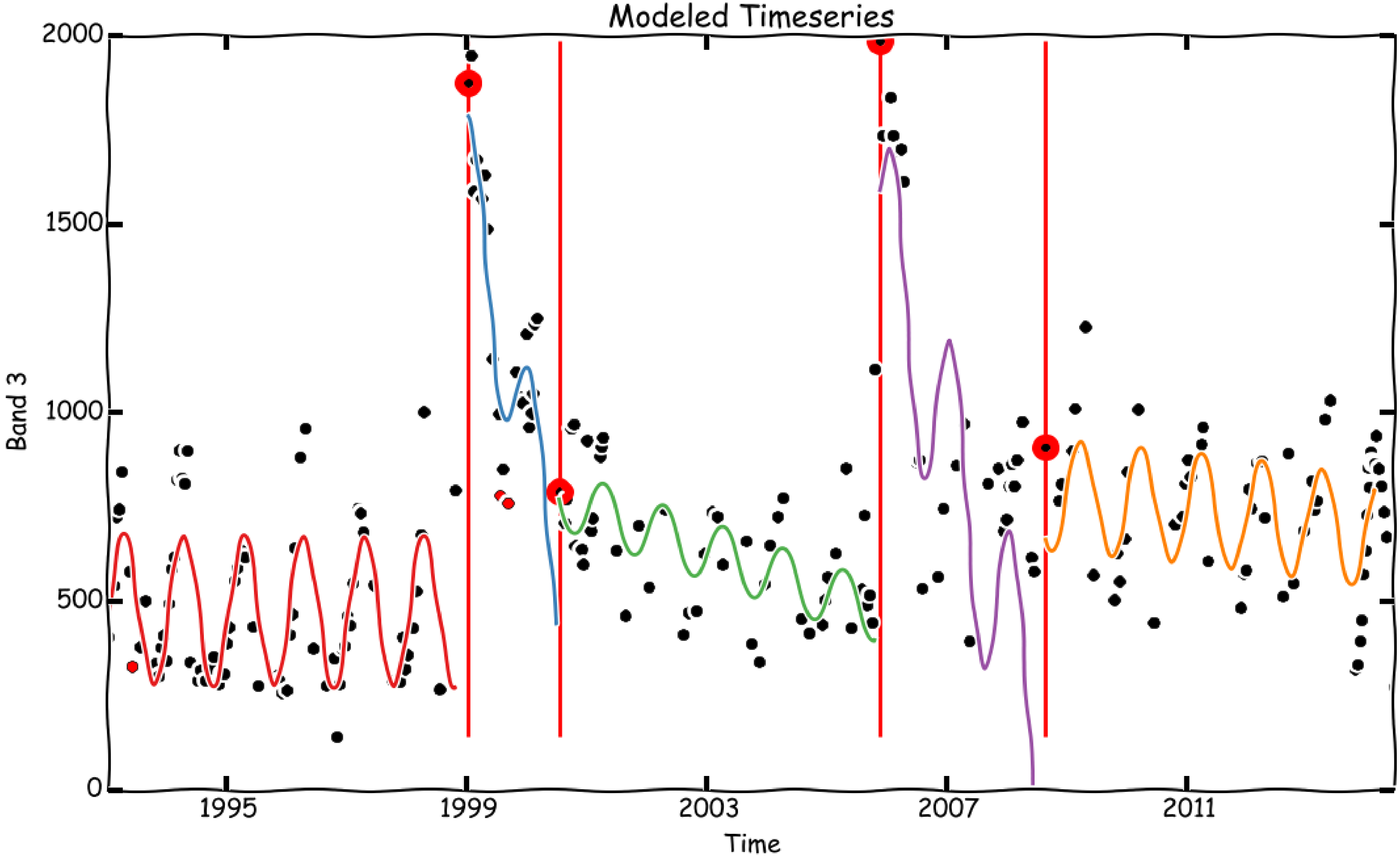
1. Display the reference data in QGIS, i.e., display the data you will use to interpret the sample you just created. This is likely a combination of different data sources, such as Landsat, RapidEye and Google Earth, acquired around the same times as the data used to create the map (in this case 2000 and 2012), and preferably also in-between.
2. Display the shapefile containing the sample, i.e. the file you created in Section 3.
3. Right-click shapefile in *Layer* pane; *Open Attribute Table*; then and then ;

Time series analysis

- CCDC implemented (on-the-fly and wall-to-wall)
- Teaming up with Wageningen University (Jan Verbesselt et al.) to implement BFAST



BOSTON UNIVERSITY


Modeled Timeseries
2000
1500
1000
500
0
Band 3
1995
1999
2003
2007
2011
Time

Pontus Olofsson, Christopher E. Holden, Eric L. Bullock
Boston Education in Earth Observation Data Analysis/
Department of Earth & Environment, Boston University

5. Additional Resources

Here we list additional resources we believe are useful. The focus is on open source tutorials, packages, and software for the topics focused on in this tutorial. In addition, we list papers we believe would be useful and relate to the material. This list will be updated as we find additional resources we find useful.

5.1 Tutorials

- Chris Holden's open source tutorial on Remote Sensing using R and Python. Including classification, calculations, and visualization:
  - https://github.com/ceholden/open-geo-tutorial
- Tutorials from Wageningen University on Remote Sensing analysis using R and Python:
  - https://github.com/GeoScripting-WUR
- Chris Holden's tutorial on searching for images, downloading, stacking, subsetting, and generating previews using Bash and Python:
  - https://github.com/ceholden/landsat_preprocess
- USGS page on Landsat data access:
  - http://landsat.usgs.gov/Landsat_Search_and_Download.php
- Downloading Landsat data via USGS GloVis:
  - http://landsat.gsfc.nasa.gov/wp-content/uploads/2013/05/Make-Your-Own-Landsat-Image-Tutorial.pdf
- Tutorial from Wageningen University using R to perform supervised (Random Forests) and unsupervised (k-means) classification:
  - http://geoscripting-wur.github.io/AdvancedRasterAnalysis/#unsupervised-classification-k-means
- Python course hosted online from Utah State University

ceholden/open-geo-tutor...
GitHub, Inc. (US) https://github.com/ceholden/open-geo-tutorial
Search

README.md

Open Source Geoprocessing Tutorial

Tutorial of basic remote sensing and GIS methodologies using open source software (GDAL in Python). Tutorial covers workflow to perform image classification using machine learning classifiers:

1. Introduction [Python] [R]
2. The GDAL datatypes and objects [Python] [R]
3. Your first vegetation index [Python] [R]
4. Visualizing data [Python] [R]
5. Vector data - the OGR library [Python] [R]
6. Land cover classification [Python] [R]
7. The spatial dimension - filters and segmentation [Python]

Access

Viewing IPython Notebook chapters

You can view these IPython notebooks as HTML pages by viewing them locally by opening the outputs in the `build` folder, or by accessing the IPython notebooks via a web viewer at here, or the chapter links shown above.

Running IPython Notebook chapters

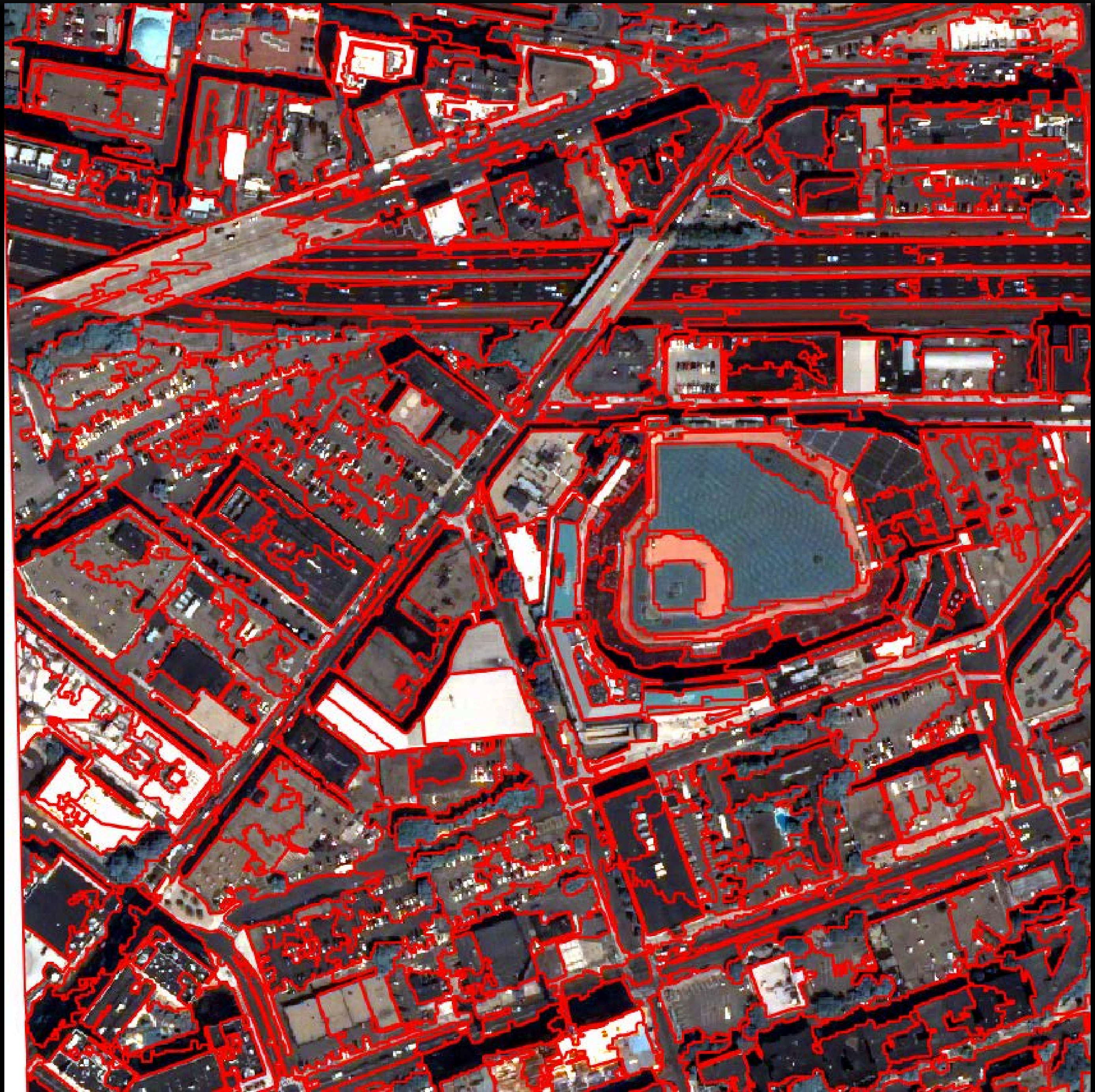
You can, of course, download this repository and run the IPython notebooks yourself. The added benefit to this is that you can interact with them.

For users who don't have the software environment required, the people behind the Wakari serivce - Continuum Analytics - have a free tier (limited CPU and space compared to paid plans) which will host IPython Notebooks over the web on their servers. These IPython Notebook environments contain all the third party libraries we will use, including GDAL, NumPy, and scikit-learn.

I've included a few special instructions for Wakari users (see Chapter 5), but the workflow for running the notebooks on Wakari is as follows:

Object based image analysis

- Popular mapping used methodology but proprietary software for OBIA often prohibitively **expensive**
- BEEODA contains open source solution
- Still in progress



BOSTON UNIVERSITY

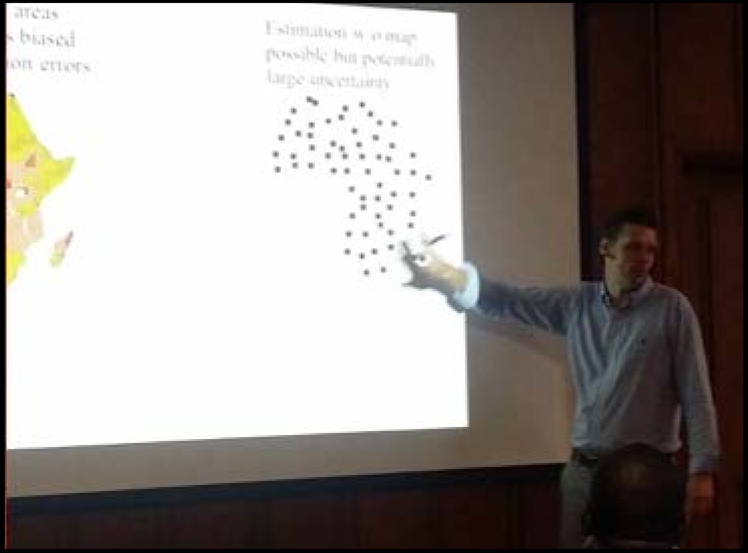
Advantages

- Fully open-source, available on GitHub
- Developed by researchers
- Integrated solution
- Customizable (add your own or other open-source tools)
- Dynamic (easily updated)
- Dependencies solved (major issue in Windows)
- One single software installation (or none if booting from USB)
- Runs offline on basically any computer

Current status

- Working on new modules, such as OBA
- Used in workshops and university courses and by researchers
- Unfunded*
- Unsupported
- Currently maintained by BU researchers and grad students...

Experiences



- Five three-week hands-on workshops at USGS/BU
- Used proprietary software until recently – *not ideal*
- USAID/FCMC workshop in Peru on accuracy/area estimation using open-source but no VM – *better but issues*
- START/GOFC-GOLD workshop at BU Jul/Aug 2015 using BEEODA – *most successful solution to date*
- Now, ready for hands-on workshops in countries
- BU undergraduate course in remote sensing, fall 2015

Comments from participants (Jul/Aug 2015)

- “*I learnt processing of data using open source software [...] which makes it easily transferable to my networks and other developing country researchers due to the free access [...]*” Mercy South Africa
- *“A crucial advantage of the training was that we performed all these advanced methods using open-source software [...] By this training, I not only acquired a lot of data and processing techniques and tools which will be shared at my home institutions, I can rather be considered as a candidate for any land cover change detection and estimation analysis. Furthermore, I can contribute to a good implementation of National REDD program and to the capacity building in my region.”* Maleki, Togo

Comments from participants (Jul/Aug 2015)

"*One important limitation […] I had experienced was that commercial software are often prohibitively expensive. Often there is a need of more than one software package for completing the analysis procedures. Further, the installation and operation of those systems are often operating system dependent and have many issues when running on even on machines with different versions of windows. Here, […] significant add-ons was the open source set of tools in the form of Virtual Machine provided to us. This has the potential to address most of the issues with conventional software systems. This can be installed as an independent operating systems and has a set of tools ranging from basic spreadsheets program to advanced image processing. Further, the tools are open source, thus we can use, adapt as per our requirement to solve diverse data analysis problems. And also all the tutorials are provided […] with free access to anybody.*

[In] my work, I have been volunteering a significant amount of time trying to help students and other fellow researchers with their data analysis issues. In doing this one important limitation was the software as for example I could demonstrate the data analysis process in a machine with a licensed software but they could not try those on their own machine. This was mainly due to the lack of licensed software and often the operation system itself or component requirements within the operating systems. Now, I am confident that access to as well as skills on using this tools will be very beneficial." Shiva, Nepal

Comments from participants (Jul/Aug 2015)

- *"[The] two week data analysis exercise at Boston University is really excellent. I really liked the neat planning and execution of training in terms of objectives like data collection, classification and accuracy assessment. [...] The use of open source tools such as QGIS, Orfeo toolbox, GDAL, Python and other in house developed tools by Boston University team provided effective and ever improving solution for satellite data analysis."* Suryakant , India
- We know for sure that many participants and their colleagues are currently using BEEODA and at their home institutions.
- In short: very well received. This is also true for the BU students who have are excited to have the whole system for analysis on their laptops – for completing labs but also for future use.

Future options

1. Continue unsupported and practitioners can use the software and materials if they want to
2. Or, if you see the benefit to build support there are option, e.g.:
   - support to continued development of BEEODA
   - support for workshops (not necessarily taught by us)
   - support for users (hire a designated developer)

Email: olofsson@bu.edu

Web: http://beeoda.org

GitHub: http://github.com/beeoda

BEEODA	UN-FAO OFGT
QGIS to call GDAL library; command-line optional	"GDAL command-line utilities"
GUI-based mainly but some functions from command-line	"Collection of prototype command-line utilities"
All tools run in a Virtual Machine	"Stand-alone programs and scripts"
Open-source (CC BY-SA 4.0, MIT and GNU GPL)	Open-source ("GNU GPLv3 license")
Runs in Windows, OS X, Linux	Runs in Windows (CygWin), OS X, Linux

BOSTON UNIVERSITY

Quotes from UN-FAO (2013). Open Foris Geospatial Toolkit User Manual, Version 1.25.4