APPENDIX G: GUIDE: RANDOM SAMPLING WITH QGIS

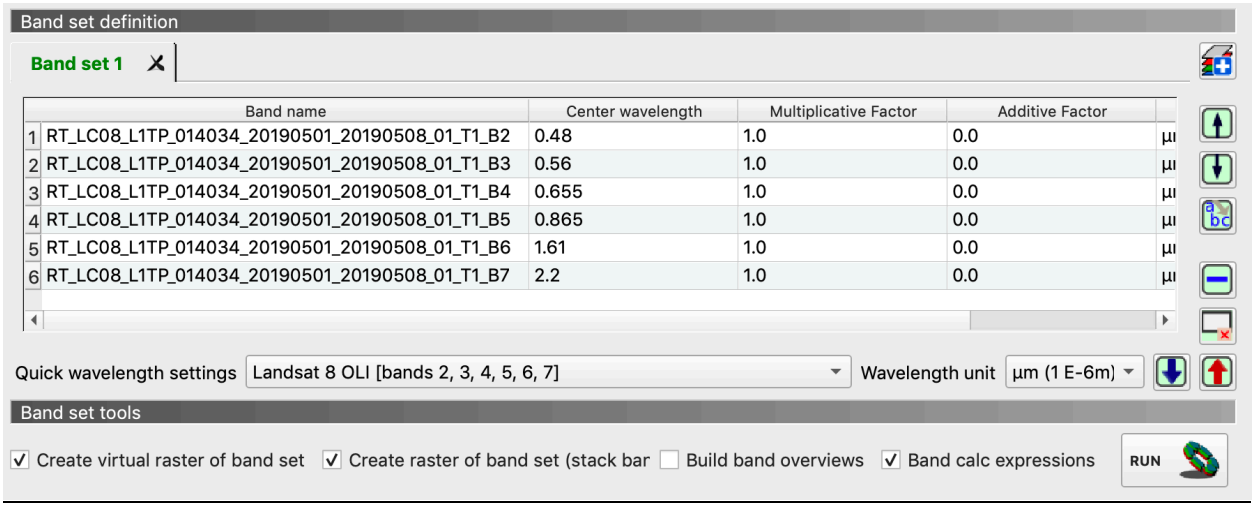
Using the semi-automatic classification plugin

* Make sure your project coordinate reference system is set before you begin and that all layers follow this convention. The VCR lies in WGS 84 / UTM zone 18N.

1. Preprocessing

a. Directory containing Landsat bands: Select entire L8 file folder
b. Select MTL file: Select mtl file from that folder
c. Options: Check the following:
    - Only to blue and green bands
    - Create band set
    - Add bands in new band set
d. Run

2. Band Set



a. Quick wavelength settings: Landsat 8 OLI (2,3,4,5,6,7)
b. Options: Check the following:
    - Create virtual raster

- Create raster
- Band calculation expressions

c. Run

d. Set RGB=3-2-1 on the toolbar

3. Classification

a. First, create a polygon around your area of interest

i. Layer > Create Layer > New Shapefile Layer

- Set coordinate reference system to Project CRS, Geometry type to polygon, and be sure to choose a location for the file manually using … next to the File name

ii. Click on  in the toolbar to edit the layer

iii. Click the  icon and digitize the polygon

a. Exit the SCP window and use the SCP dock

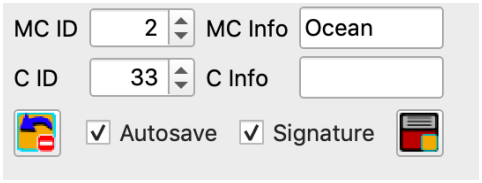
b. Training input tab:

i. Select  to create a shapefile for training samples

ii. Use  on the toolbar to digitize polygons. Left click to create vertices and right click to close the polygon. (I had trouble closing the polygons with my mousepad on my laptop and had to use an external mouse).

- Be sure to define the MC ID and C ID. I used MC (macroclass) ID = 1 for water and MC ID = 2 for water. The C ID did not matter because I did not use it for classification.

iii. Go to Macroclass list tab to make sure you macroclasses are listed.



c. Classification tab:

i. Use MC ID for classification

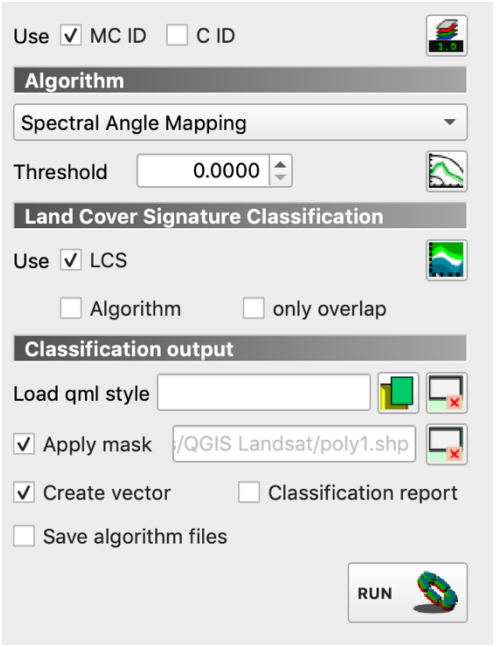
ii. Use Spectral Angle Mapping

iii. Land Cover Signature Classification

- Check LCS for

iv. Classification output

- Apply mask: Set as shapefile with the polygon around the area of interest



- Check Create vector

v. Run

d. Check to make sure your classification is accurate, and if not, digitize more polygon

Using Research Tools from QGIS

Vector > Research Tools

1. Create Grid
    a. Grid type: Choose one of the polygon options. I chose rectangle
    b. Grid extent > Create from layer: Set to polygon around your area of interest
    c. Set horizontal spacing and vertical spacing depending on how many points you want to sample. I recommend starting with 1000 m x 1000 m and then either increasing or decreasing from there until the grid has as many cells as you want.
    d. Grid CRS > Project CRS
2. Clip the grid vector layer: Vector > Geoprocessing tools > Clip
    a. Input layer > grid
    b. Overlay layer > polygon around area of interest
3. Vector > Research Tools > Random points in layer bounds
    a. Input layer > Grid
    b. Number of points: 1
    c. Minimum distance between points: 1000m

You will have one randomly sampled point in each cell of your grid.

4. Reproject point layer and classified vector with Vector > Data Management Tools > Reproject layer > Target CRS: Project CRS
    a. This step may be redundant, but it ensures that both your points and classified image are projected onto the same coordinate system.

Using the Point Sampling Tool 

1. Layer containing sampling points: Point layer, Layers with fields/bands to get values from: Classified vector layer. Save as .gpkg (default)
2. Right click on .gpkg layer and Open Attribute Table
3. Select/Filter Using Form 
4. Select C_ID to 1
5. Inverse selection
6. Toggle editing mode 
7. Delete Selected Features 
8. Save edits
9. Right click on .gpkg layer, Export>Save Vector Layer As…

i. Formary > CSV
ii. Set file location
iii. Geometry > AS_YX

Converting UTM to Decimal Degrees in R

```
library(rgdal)
setwd('/Users/sarahlang/Documents/QGIS Landsat/')
# .csv saved from QGIS
data<-read.csv('coords1.csv')
utmcoords<-SpatialPoints(cbind(data$X,data$Y), proj4string=CRS("+proj=utm
+zone=18N +datum=WGS84"))
ddcoords <- spTransform(utmcoords, CRS("+proj=longlat +datum=WGS84"))
write.csv(ddcoords@coords,"\\Converted.csv")
```

Sources:
https://semiautomaticclassificationmanual-v5.readthedocs.io/_/downloads/ar/latest/pdf/
https://pvanb.wordpress.com/2010/02/15/sampling-raster-values-at-point-locations-in-qgis/
https://stackoverflow.com/questions/30018098/how-to-convert-utm-coordinates-to-lat-and-long-in-r/30018607
https://CRAN.R-project.org/package=rgdal