# Appendices

**Appendix A:**

Code developed for obtaining raw satellite imagery in Google Earth

Engine

////Code developed by Daniëlle Seymour\\\\

Sentinel-2 MultiSpectral Instrument: Level-2A raw imagery in Google Earth Engine:

///Insert Sentinel 2-A Image Collection and filter by area using an imported shapefile\\\

var image = ee.ImageCollection('COPERNICUS/S2\_SR')

.filterBounds(WetlandExtent2);

///Function to mask clouds S2\\\

var mask = function(image) {

var QA60 = image.select(['QA60']);

var clouds = QA60.bitwiseAnd(1<<10).or(QA60.bitwiseAnd(1<<11)); // this gives us cloudy

pixels

return image.updateMask(clouds.not()); // remove the clouds from image

};

///Filter imagery for Summer 2020/21 date ranges \\\

///Create joint filter and apply it to Image Collection.\\\

var Summer2020 = ee.Filter.date('2020-12-01','2021-02-28');

var SummerFilter = ee.Filter.or(Summer2020);

var allsum = image.filter(SummerFilter);

///Make a Composite: Apply the cloud mask function, use the median reducer, and clip the

composite to area of interest\\\

///create visualization parameters for composite\\\

var rgbVis = {

bands: ['B4', 'B3','B2'],

min: 0,

max: 3000,

gamma: [0.95, 1.1, 1]};

var composite = allsum

.map(mask)

.median()

.clip(WetlandExtent2);

///Specify a projection\\\

var proj = ee.Projection('EPSG:32734');

print(proj);

///Display the Composite\\\

Map.addLayer(composite, rgbVis,'Du Toits Wetland\_Summer', 0);

Map.setCenter(19.162248, -33.974126, 13);

///Export the image, specifying scale and region.\\\

Export.image.toDrive({

image: composite,

description: 'S2JanFebComp\_2021',

folder: 'Raw/Imagery/Reprojected',

scale: 10,

region: WetlandExtent2

//fileFormat: 'GeoTIFF'

});

130

////Filter imagery for Winter 2020 date ranges\\\

///Create joint filter and apply it to Image Collection.\\\

var Winter2020 = ee.Filter.date('2020-06-01','2020-08-30');

var WinterFilter = ee.Filter.or(Winter2020);

var allsum = image.filter(WinterFilter);

///Make a Composite: Apply the cloud mask function, use the median reducer, and clip the

composite to our area of interest\\\

///create visualization parameters for composite\\\

var true\_colour = {

bands: ['B4', 'B3','B2'],

min: 0,

max: 3000,

gamma: [0.95, 1.1, 1]};

var composite = allsum

.median()

.clip(WetlandExtent2);

///Specify a projection\\\

var proj = ee.Projection('EPSG:32734');

print(proj);

///Display the Composite\\\

Map.addLayer(composite, true\_colour,'Du Toits Wetland\_Winter', 0);

Map.setCenter(19.162248, -33.974126, 13);

///Export the image, specifying scale and region.\\\

Export.image.toDrive({

image: composite,

description: 'S2WinterComp\_2020',

folder: 'Raw/Imagery/Reprojected2',

scale: 10,

region: WetlandExtent2

//fileFormat: 'GeoTIFF'

});

//////////////////////////////////////////////////////End of script\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

**Appendix B:**

**Description of landcover classes**

Transects were laid out from east to west with 1-meteruadrats at each 15m interval, which is equivalent to a pan-sharpened Landsat 8 pixel (see Figure 1.4). The transects were placed to sample vegetation in each quadrat to identify the frequency/proportion of plant species that indicate terrestrial and wetland habitats (Table 1). *Prionium Serratum* and *Psoralea pinnata* within Peatland wetland vegetation (i.e., wetland subtype-1)*,* a condensed group of *Pteridium aquilinum, Restio paniculatus,* and *Merxmuellera cincta* within Sclerophyllous Wetland Vegetation (i.e., wetland subtype-2), Fynbos, Bare soil/sandstone, Water and Degraded. Seven classes (i.e., splitting wetland vegetation into *Prionium serratum* and *Psoralea pinnata*) were decided on based on field observations and understanding the species composition of the wetland in its context (Table 2). Terrestrial vegetation was aggregated into a single class, as this vegetation group was primarily based on single vegetation occurrences of non-wetland vegetation such as *Brabejum stellatifolium* and *Acacia mearnsii* (invasive alien tree). These species do not occur in dense clusters that are spectrally visible from RS imagery and were therefore not included as a single class in the refined landcover classes. Hence, for the purpose of this study, Sclerophyllous Wetland Vegetation (henceforth referred to as SWV) is considered a mixture of facultative wetland species i.e., grasses, ferns and restios, intermittently mixed with upland Fynbos species, and occasionally small single patches of non-wetland vegetation may be present. Moreover, field observations have shown that soils in the SWV quadrats were sandier and drier, rather than the deeper, organic (peat conditions) found within Wetland Vegetation quadrats. Hence Wetland Vegetation and Sclerophyllous Wetland Vegetation are split as belonging to two subtypes of overall wetland vegetation based on the National Wetland Vegetation Database (Sieben, Mtshali & Janks 2014). Bare soil and the Degraded classes were additionally added to reduce the misclassification of unassigned pixels. Species identification was done using the SANParks Scientific Services library and herbarium. Where plants were not found in books or samples in the herbarium, digital photos were uploaded and searched for online using an application called Plantsnap, as well as the SANBI online website named PlantZAfrica (http://pza.sanbi.org/). Each individual species was recorded in a quadrant and given a percentage cover or frequency of occurrence in the quadrat. Where an individual species covered a percentage higher than 50%, the species would then be used to determine whether this quadrat was terrestrial habitat or wetland habitat.

A second field visit 7-9 June 2021 (winter) collected 26 observations of dominant species and subsequent class assignment per GPS point. A simple random sampling design (Wegman et al. 2016) across the full extent of the wetland was generated in ArcGIS 10.7.1. and field samples were collected as close as possible to these randomly generated points, given access constraints ().

Table 1. Species and percentage cover per quadrant and subsequent Landcover Class assignment during field vegetation sampling, October 2020

\*Note: Soil property measurements were recorded in the second round of data collection due to equipment delays.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Time started:** | **Date:** | **Co-ordinates X:** | **Co-ordinates Y:** | **Elevation:** | **Quadrat:** | **Plant id:** | **Common names** | **Quadrat Percentage:** | **Landcover Class** | **Soil: 50cm** | **Soil: 100cm** | **Munsell Chart Reading** | **Comments** |
| 09h45 | 21/10/2020 | 19.189399 | -34.004305 | 320.528229 | Q1 | *Merxmuelera cincta* |  | 10% | swv |  |  |  |  |
|  |  | 19.189399 | -34.004305 | 320.528229 | Q1 | *Wachendorfia thyrsiflora* |  | 1% | terrestrial |  |  |  |  |
|  |  | 19.189399 | -34.004305 | 320.528229 | Q1 | *Merxmuelera cincta* |  | 1% | swv |  |  |  |  |
|  |  | 19.189399 | -34.004305 | 320.528229 | Q1 | *Chordifex fastigiatus* |  | 8% | wetland |  |  |  | Australian wetland plant |
|  |  | 19.189399 | -34.004305 | 320.528229 | Q1 | *Eligia felicae* |  | 50% | wetland | sandy, loam |  |  | Restio-native, fynbos |
|  |  | 19.189399 | -34.004305 | 320.528229 | Q1 | *Restio paniculatus* |  | 5% | swv |  |  |  |  |
|  |  | 19.189399 | -34.004305 | 320.528229 | Q1 | *Helichrysum milfordiae* |  | 15% | terrestrial |  |  |  | Indigenous |
|  |  | 19.189399 | -34.004305 | 320.528229 | Q1 | *Osteospermum Polygaloides* |  | 10% | terrestrial |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10h05 | 21/10/2020 | 19.1895 | -34.004194 | 321.516785 | Q2 | *Isolepis prolifera* |  | 1% | swv |  |  |  |  |
|  |  | 19.1895 | -34.004194 | 321.516785 | Q2 | unidentified |  | 1% |  |  |  |  |  |
|  |  | 19.1895 | -34.004194 | 321.516785 | Q2 | unidentified |  | 1% |  |  |  |  |  |
|  |  | 19.1895 | -34.004194 | 321.516785 | Q2 | *Cliffortia strobilifera* |  | 6% | swv |  |  |  |  |
|  |  | 19.1895 | -34.004194 | 321.516785 | Q2 | *Eligia felicae* |  | 60% | wetland |  |  |  | fynbos restio |
|  |  | 19.1895 | -34.004194 | 321.516785 | Q2 | *Epischoenis gracilis* |  | 15% | wetland |  |  |  | sedge |
|  |  | 19.1895 | -34.004194 | 321.516785 | Q2 | *Psoralea pinnata* | fountainbush | 8% | wetland |  |  |  |  |
|  |  | 19.1895 | -34.004194 | 321.516785 | Q2 | unidentified |  | 5% |  |  |  |  |  |
|  |  | 19.1895 | -34.004194 | 321.516785 | Q2 | unidentified |  | 3% |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10h35 | 21/10/2020 | 19.189428 | -34.004275 | 318.754395 | Q3 | *Leucadendron conicum* |  | 98% | fynbos |  |  |  |  |
|  |  | 19.189428 | -34.004275 | 318.754395 | Q3 | unidentified |  | 1% |  |  |  |  |  |
|  |  | 19.189428 | -34.004275 | 318.754395 | Q3 | unidentified |  | 1% |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11h10 | 21/10/2020 | 19.189327 | -34.004341 | 318.820557 | Q4 | *Elegia filicae* |  | 60% | wetland |  |  |  | Rocky, loose white soil |
|  |  | 19.189327 | -34.004341 | 318.820557 | Q4 | *Pteridium aquilinium* | bracken fern | 10% | wetland |  |  |  |  |
|  |  | 19.189327 | -34.004341 | 318.820557 | Q4 | *Drosera trinervia* |  | 5% | wetland |  |  |  | damp, peaty exposed areas |
|  |  | 19.189327 | -34.004341 | 318.820557 | Q4 | unidentified |  | 1% | terrestrial |  |  |  |  |
|  |  | 19.189327 | -34.004341 | 318.820557 | Q4 | *Psoralea pinnata L.* | fountain bush (Eng.); fonteinbos, bloukeur, penwortel (Afr.); umHlonishwa (Zulu) | 5% | wetland |  |  |  |  |
|  |  | 19.189327 | -34.004341 | 318.820557 | Q4 | *Leucadendron conicum* |  | 10% | wetland |  |  |  | peat |
|  |  | 19.189327 | -34.004341 | 318.820557 | Q4 | unidentified |  | 5% |  |  |  |  |  |
|  |  | 19.189327 | -34.004341 | 318.820557 | Q4 | unidentified |  | 3% |  |  |  |  |  |
|  |  | 19.189327 | -34.004341 | 318.820557 | Q4 | unidentified |  | 1% |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11h55 | 21/10/2020 | 19.18921 | -34.004425 | 317.355652 | Q5 | *Laurembergia repens* |  | 10% | swv |  |  |  |  |
|  |  | 19.18921 | -34.004425 | 317.355652 | Q5 | unidentified |  | 5% |  |  |  |  |  |
|  |  | 19.18921 | -34.004425 | 317.355652 | Q5 | *Cliffortia strobilifera* |  | 15% | swv |  |  |  |  |
|  |  | 19.18921 | -34.004425 | 317.355652 | Q5 | unidentified |  | 3% |  |  |  |  |  |
|  |  | 19.18921 | -34.004425 | 317.355652 | Q5 | unidentified |  | 10% |  |  |  |  |  |
|  |  | 19.18921 | -34.004425 | 317.355652 | Q5 | unidentified |  | 5% |  |  |  |  |  |
|  |  | 19.18921 | -34.004425 | 317.355652 | Q5 | *Thelypteris confluens* |  | 40% | swv |  |  |  |  |
|  |  | 19.18921 | -34.004425 | 317.355652 | Q5 | unidentified |  | 10% |  |  |  |  |  |
|  |  | 19.18921 | -34.004425 | 317.355652 | Q5 | unidentified |  | 2% |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12h30 | 21/10/2020 | 19.188946 | -34.005041 | 311.366852 | Q6 | *Thelypteris confluens* |  | 50% | swv |  |  |  |  |
|  |  | 19.188946 | -34.005041 | 311.366852 | Q6 | small swv plants |  | 50% | swv |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12h45 | 21/10/2020 | 19.188871 | -34.005221 | 310.508057 | Q7 | *Isolepis prolifera* |  | 40% | swv |  |  |  |  |
|  |  | 19.188871 | -34.005221 | 310.508057 | Q7 | *Merxmuelera cincta* |  | 5% | swv |  |  |  |  |
|  |  | 19.188871 | -34.005221 | 310.508057 | Q7 | *Merxmuelera cincta* |  | 5% | swv |  |  |  |  |
|  |  | 19.188871 | -34.005221 | 310.508057 | Q7 | *Merxmuelera cincta* |  | 50% | swv |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13h12 | 21/10/2020 | 19.181979 | -33.986955 | 332.724548 | Q8 | *Brabejum stellatifolium* |  | 100% | terrestrial | loose dry white sand |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13h30 | 21/10/2020 | 19.181713 | -33.986891 | 332.4888 | Q9 | *Searsia augustifolia* |  | 10% | terrestrial |  |  |  |  |
|  |  | 19.181713 | -33.986891 | 332.4888 | Q9 | unidentified |  | 5% |  |  |  |  |  |
|  |  | 19.181713 | -33.986891 | 332.4888 | Q9 | unidentified |  | 20% |  |  |  |  |  |
|  |  | 19.181713 | -33.986891 | 332.4888 | Q9 | unidentified |  | 10% |  |  |  |  |  |
|  |  | 19.181713 | -33.986891 | 332.4888 | Q9 | *Dicerothamnus rhinocerotis* |  | 50% | swv |  |  |  |  |
|  |  | 19.181713 | -33.986891 | 332.4888 | Q9 | unidentified |  | 15% |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14h12 | 21/10/2020 | 19.181349 | -33.987252 | 330.188507 | Q10 | *Thelypteris confluens* |  | 60% | wetland |  |  |  |  |
|  |  | 19.181349 | -33.987252 | 330.188507 | Q10 | *Cliffortia strobilifera* |  | 20% | swv |  |  |  |  |
|  |  | 19.181349 | -33.987252 | 330.188507 | Q10 | *Carpobrotus edulis* |  | 5% | terrestrial |  |  |  |  |
|  |  | 19.181349 | -33.987252 | 330.188507 | Q10 | *Epischoenis gracilis* |  | 5% | wetland |  |  |  |  |
|  |  | 19.181349 | -33.987252 | 330.188507 | Q10 | *Pteridium aquilinum* |  | 5% | wetland |  |  |  |  |
|  |  | 19.181349 | -33.987252 | 330.188507 | Q10 | *Cyperus thunbergii* |  | 5% | wetland |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14h35 | 21/10/2020 | 19.181158 | -33.98734 | 330.125275 | Q11 | *Merxmuelera cincta* |  | 90% | wetland |  |  |  | grass |
|  |  | 19.181158 | -33.98734 | 330.125275 | Q11 | *Elegia filicae* |  | 5% | wetland |  |  |  |
|  |  | 19.181158 | -33.98734 | 330.125275 | Q11 | *Elegia capensis* |  | 5% | fynbos |  |  |  | Fynbos |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14h50 | 21/10/2020 | 19.178657 | -33.982652 | 332.728149 | Q12 | *Merxmuelera cincta* |  | 80% | wetland |  |  |  | grass |
|  |  | 19.178657 | -33.982652 | 332.728149 | Q12 | *Elegia filicae* |  | 10% | wetland |  |  |  |  |
|  |  | 19.178657 | -33.982652 | 332.728149 | Q12 | *Juncus spp.* |  | 10% | wetland |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 08h45 | 22/10/2020 | 19.173443 | -33.958172 | 348.007935 | Q13 | *Prionium serratum* |  | 30% | wetland |  |  |  |  |
|  |  | 19.173541 | -33.958126 | 348.007935 | Q13 | *Pteridium aquilinum* |  | 30% | swv |  |  |  |  |
|  |  | 19.17337 | -33.958029 | 337.292877 | Q13 | *Psoralea pinnata L.* |  | 20% | wetland |  |  |  |  |
|  |  | 19.173289 | -33.958003 | 336.508545 | Q13 | *Psoralea pinnata L.* |  | 20% | wetland |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 09h20 | 22/10/2020 | 19.17197 | -33.951853 | 346.822266 | Q14 | *Pteridium aquilinum* |  | 80% | swv |  |  |  |  |
|  |  | 19.17197 | -33.951853 | 346.822266 | Q14 | *Carpha glomerata* |  | 10% | wetland |  |  |  |  |
|  |  | 19.17197 | -33.951853 | 346.822266 | Q14 | *Psoralea pinnata L.* |  | 5% | wetland |  |  |  |  |
|  |  | 19.17197 | -33.951853 | 346.822266 | Q14 | *Restio paniculatus* |  | 5% | swv |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Table Seven refined distinct landcover classes for Random Forest and Support Vector Machine classification.

Sclerophyllous Wetland Vegetation condensed with sporadically occurring upland Fynbos vegetation

(Wetland subtype-2)

Peatland Vegetation

(Wetland subtype-1)

|  |  |
| --- | --- |
| **Landcover class** | **Description:** |
| *Prionium serratum* | Commonly known as palmiet, and are considered ecosystem engineers in wetlands, creating deep peat conditions (Job 2014; Rebelo et al. 2019; Rebelo, Somers, et al. 2018; Sieben, Mtshali & Janks 2014). |
| *Psoralea pinnata* | An erect shrub, or small tree, commonly known as fountain bush, that can reach an estimated height of 4 m and typically grows along streams and saturated environments (Palmer & Pitman 1973). |
| Sclerophyllous Wetland Vegetation | This cluster comprises a grouping of *Pteridium aquilinum* (Bracken fern), *Restio paniculatus, Elegia capensis* and wetland grasses such as *Merxmuellera cincta* (Sieben, Mtshali & Janks 2014). These three vegetation communities were grouped as one class in the classification as they often co-occurred in the wetland. |
| Fynbos | Fynbos species belonging to the CFR such as *Protea neriifolia*, *Berzelia abrotanoides*, *Leucadendron conicum, Leucadendron coniferum* and *Metalasia muricata* (Rebelo et al. 2006)*.* |
| Bare soil/sandstone | All visible sandy deposits around active, exposed channels and eroded channels, or exposed and degraded areas of land. |
| Water | All openly visible water and channels/tributaries. |
| Degraded | This class is vegetation that is not ground-truthed or sampled but visually appears as degraded vegetation (possibly burnt) from previous farming practices and water extraction. |

Table . Dominant species and subsequent class assignment per GPS point () during field vegetation sampling, 7-9 June 2021 (winter). A simple random sampling design (Wegman et al. 2016) across the full extent of the wetland was generated in ArcGIS 10.7.1. and field samples were collected as close as possible to these randomly generated points, given access constraints.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Co-ordinates X:** | **Co-ordinates Y:** | **Landcover class** | **Elevation** | **Dominant plant ID** | **FID** | **Date:** | **soil: 50cm** | **soil: 100cm** | **Munsell Chart Reading** | **Comments** |
| 19.18631 | -34.0042 | Wetland | 311.0545 | *Prionium serratum* | Q1 | 07-Jun-21 | Sandy loam, damp | Sandy loam, very very wet | Hue: 5YR, Value: 2.5, Chroma: 1 | A lot of organic matter in 50cm, very coarse, very wet. At 100cm completely saturated.Patches of agglomerated palmiet close to narrow tributries. |
| 19.18631 | -34.0044 | SWV | 310.2428 | *Pteridium aquilinum* | Q2 | 07-Jun-21 | Sandy loam, low wetness, fairly dry | Sandy loam, slightly damp | Hue: 5YR, Value: 2.5, Chroma: 1 | Very little organic matter, could be seasonally wet zone, Light grey. |
| 19.18593 | -34.0046 | SWV | 311.2105 | *Merxmuellera cincta* | Q3 | 07-Jun-21 | Sandy loam, very wet | Sandy loam, very wet | Hue: 5YR, Value: 2.5, Chroma: 1 | Higher & finer organic material, sandy granules closer together and finer |
| 19.18371 | -34.0035 | SWV | 310.9797 | *Brabejum stellatifolium* | Q4 | 07-Jun-21 | Sand, nutrient poor | Sand, damp | Hue: 5YR, Value: 2.5, Chroma: 1 | Not fine granules but gritty, dry and not very course |
| 19.18407 | -34.0035 | Fynbos | 311.996 | *Leucadendron coniferum* | Q5 | 07-Jun-21 | Sandy loam, very wet | Sandy loam, leached | Hue: 5YR, Value: 2.5, Chroma: 1 | Gritty, fine granules |
| 19.17867 | -33.9919 | SWV | 321.138 | *Pteridium aquilinum* | Q6 | 07-Jun-21 | Sand | Sand | Hue: 10YR, Value 4, Chroma: 4 | More sand than loam fairly organic but <30% organic carbon |
| 19.17846 | -33.9918 | SWV | 319.4586 | *Restio paniculatus Rottb.* | Q7 | 07-Jun-21 | Sandy loam | Sandy loam | Hue: 10YR, Value 4, Chroma: 4 | Fine, reddish sand, little organic matter |
| 19.17787 | -33.9914 | SWV | 320.4131 | *Merxmuellera cincta* | Q8 | 07-Jun-21 | Loam | Loam | Hue: 7.5YR, Value: 3, Chroma: 3 | Fine, brown sand, not very wet & compact |
| 19.17917 | -33.9904 | Fynbos | 327.0022 | *Leucadendron coniferum* | Q9 | 07-Jun-21 | Sandy loam | Sandy loam | Hue: 7.5YR, Value: 3, Chroma: 3 | Very low wetness |
| 19.17845 | -33.9827 | Fynbos | 334.1852 | *Berzelia abrotanoides* | Q10 | 08-Jun-21 | Loamy sand | Clay | Hue: 2.5 , Value: 3, Chroma: 1 | Course, and gritty chalk white colour at about 70cm |
| 19.17819 | -33.9828 | Fynbos | 330.3489 | *Metalasia muricata* | Q11 | 08-Jun-21 | Sand | Sand | N7 Value: 6, Gleyed soil, sand | Little organic matter |
| 19.178 | -33.9831 | SWV | 332.025 | *Brabejum stellatifolium* | Q12 | 08-Jun-21 | Loamy sand | Loamy sand | Hue: 10YR, Value: 6, Chroma: 2 | High organic matter, very dark, fine sand that’s almost silty |
| 19.17791 | -33.9834 | Fynbos | 332.1571 | *Protea nerifolia* | Q13 | 08-Jun-21 | Loamy sand | Loamy sand | Hue: 2.5Y, Value: 4, Chroma: 2 | Damp, slightly wet and at 100 cm it gets slightly light brown |
| 19.17733 | -33.9835 | SWV | 329.3565 | *Merxmuellera cincta/Elegia capensis* | Q14 | 08-Jun-21 | Sandy loam, very saturated | Sandy loam | Hue: N 2.5, Value: 2.5 | Surface water visible, gleyed |
| 19.16606 | -33.9688 | Wetland | 321.7184 | *Psoralea pinnata L.* | Q15 | 08-Jun-21 | Sandy loam | Sandy loam, clay | Hue: 10YR, Value: 6, Chroma: 2 | Thick organic top 50cm layer, can dominantly mixed with dense clogs of palmiet, ferns and restios as understory |
| 19.1664 | -33.9689 | SWV | 324.8384 | *Pteridium aquilinum* | Q16 | 08-Jun-21 | Loamy sand | Loamy sand | Hue: 2.5Y, Value: 3, Chroma: 2 | Grey & gleyed towards 100cm mark with high organic matter |
| 19.16666 | -33.9689 | SWV | 324.2585 | *Restio paniculatus Rottb.* | Q17 | 08-Jun-21 | Sand, reddish | Sand, darker brown to black | Hue: 10YR, Value: 6, Chroma: 2 | First 15cm is sand, then becomes reddish, then at 50cm its much darker and black |
| 19.17082 | -33.9635 | Wetland | 335.7029 | *Psoralea pinnata L.* | Q18 | 08-Jun-21 | Clay | Clay | Hue: 10YR, Value: 6, Chroma: 2 | Black, then becomes greyish sand further down into 100cm |
| 19.17063 | -33.9634 | Wetland | 343 | *Psoralea pinnata L./Restio paniculatus Rottb./Zantedeshichia aethiopica* | Q19 | 09-Jun-21 | Loamy clay | Loamy clay | Gleyed N 2.5 | Smooth velvety, very dark, very dense , organic clay |
| 19.17052 | -33.9633 | Wetland | 334 | *Prionium serratum* | Q20 | 09-Jun-21 | Clayey silt | Sandy loam | Gleyed N 2.5 | Smooth velvety, very dark, very dense , organic clay |
| 19.1736 | -33.9571 | SWV | 344 | *Restio paniculatus Rottb.* | Q21 | 09-Jun-21 | Sandy loam | Loamy, sandy, clay | Gleyed 5 G 5.1 | Very wet, surface water |
| 19.17374 | -33.9571 | Fynbos | 343 | *Berzelia abrotanoide/Elegia capensis/Restio paniculatus Rottb./Leucadendron/Pteridium aquilinum* | Q22 | 09-Jun-21 | Sand | Loamy sand | Hue: 2.5Y, Value: 4, Chroma: 2 | Light brown, then darker to the core |
| 19.17395 | -33.9569 | Fynbos | 343 | *Berzelia abrotanoides* | Q23 | 09-Jun-21 | Sand | Sand | Gleyed sand | fine, lighter, gritty sand |
| 19.17345 | -33.9546 | Wetland | 343 | *Psoralea pinnata L.* | Q24 | 09-Jun-21 | Loamy, fine sand | Sand | null | high organic matter , fine |
| 19.17399 | -33.9555 | SWV | 347 | *Pteridium aquilinum* | Q25 | 09-Jun-21 | hard rock | hard rock | null | null |
| 19.17301 | -33.9555 | Wetland | 347 | *Prionium serratum* | Q26 | 09-Jun-21 | Google earth point | null | null | null |

**APPENDIX C:**

Code developed in R for the Random Forest classifier:

###Based on code developed by Blessing Khavu, provided by Blessing Khavu as part of postgrad research group peer learning###

Winter RF:

###Random Forest Classification- to spectrally discriminate various landcover/vegetation of Du

Toits Wetland###

###Spectral signature graph code developed by Daniëlle Seymour###

###This script uses the 7 distinct vegetation classification scheme###

###This is RF script for Winter 2020###Sentinel-2A, MSI Level-2A###

###Raw imagery processed in GEE: composite spans '2020-06-01','2020-08-31'###

###Load packages###

library(rgdal)

library(raster)

library(rasterVis)

library(caret)

library(randomForest)

library(e1071)

############################Load Image, create list and brick raster###################

###List band and then brick the raster:

###Raster stack vs brick: how they store each band is different. The bands in a RasterStack are

stored as links to raster data that is located somewhere on our computer.

###A RasterBrick contains all of the objects stored within the actual R object. In most cases, we can

work with a RasterBrick in the same way we might work with a RasterStack.

###However a RasterBrick is often more efficient and faster to process - which is important when

working with larger files.

rm(list=ls())

setwd("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/RF\_Winter/Imagery\_Winter")

#First import all files in a single folder as a list

rastlist1 <- list.files(path =

"C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/RF\_Winter/Imagery\_Winter",

pattern='.tif', all.files=TRUE, full.names=FALSE)

img<brick("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/RF\_Winter/Imagery\_Winter/S2WinterCompRepro.tif")

names(img) <- c('B1', 'B2', 'B3', 'B4', 'B5', 'B6','B7', 'B8','B8A','B9','B10','B11', 'B12','AOT','WVP',

'SCL','TCI\_R', 'TCI\_G','TCI\_B','MSK\_CLDPRB','MSK SNWPRB', 'QA60', 'QA60 Bitmask')

names(img) <- paste0("B", c(1:23))

#########################Plotting RGB image##################################

plotRGB(img \* (img >= 0), r = 4, g = 3, b = 2, scale = 10000)

###########Load shapefile with class coordinates and class name i.e. Prionium serratum, Psoralea pinnata, Pteridium\_Restio\_Merx, Fynbos, Bare soil/sandstone, Water and

Degraded################

#######You can add it along with values of each band or do it in R###

trainData <-

shapefile("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/TrainingData\_shp5/DistinctTraining5.shp")

head(trainData)

###################Extract values to the shapefile that you just loaded###########

#beginCluster() #to optimize all cores

roi\_data <- extract(img, trainData, df=TRUE)

roi\_data$Class <- as.factor(trainData$Class[roi\_data$ID])

roi\_data <- roi\_data[roi\_data$Class!="0",]

head(roi\_data)

###Second option of extraction for signature plot###

roi\_data <- extract(img, trainData, df=TRUE)

head(roi\_data)

summary(roi\_data)

###Create signature plot###

specs <- aggregate(roi\_data, list(trainData$Class), mean, na.rm=TRUE)

specs

#instead of the first column, use row names

rownames(specs) <- specs[,1]

specs <- specs[,-1]

specs

setwd("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest")

write.csv(specs, "SpecsDistinct.csv")

#Create a vector of colour for the land cover classes for use in plotting

mycolor <- c('grey','red','yellow','dark green','light green','orange','blue')

#transform ms from a data.frame to a matrix

specs <- as.matrix(specs)

# First create an empty plot

plot(0, ylim=c(0,4000), xlim = c(1,12), type='n', xlab="Bands", ylab = "Reflectance")

##add the different classes

for (i in 1:nrow(specs)){

lines(specs[i,], type = "l", lwd = 3, lty = 1, col = mycolor[i])

}

# Title

title(main="Spectral Profile of Distinct Classes-S2A", font.main = 2)

# Legend

legend("topleft", rownames(specs),

cex=0.8, col=mycolor, lty = 1, lwd =3, bty = "n")

###########Set seed to make sure the same random sample is selected next time########

set.seed(200)

###Note: seed---Random number seed to use. If a value is provided, it will be used to initialize R's

random number generator before the model is fitted. ##

135

###If a value is not provided (the default), the random number generator will be initialized from the current time##########################################

#################Split the data set into test and training data set###################

splitIndex <- createDataPartition(roi\_data$Class,

p = .80,

list = FALSE,

times = 1)

trainDF <- roi\_data[ splitIndex,]

testDF <- roi\_data[-splitIndex,]

trainDF <- na.omit(trainDF)

#########################Load the Column names################

trainDF <- trainDF[, c('Class', "B1", "B2", "B3", "B4", "B5", "B6","B7", "B8","B9","B10","B11",

"B12")]

testDF <- testDF[, c('Class',"B1", "B2", "B3", "B4", "B5", "B6","B7", "B8","B9","B10","B11",

"B12")]

#########################Build the Random Forest Model with Training Data############

Fitcontrol <- trainControl("repeatedcv",

number=10,

repeats=1)

rf <- train(as.factor(Class) ~.,

data = trainDF,

method= "rf",

trControl = Fitcontrol,

preProcess = c("center", "scale"),

importance = TRUE)

###Print the model summary from the Random Forest model###

print(rf)

###Check for the Variable importance####

plot(varImp(rf,scale=FALSE))

### Predict to a new Dataframe for a Map Output#######

pred <- predict(rf , newdata = testDF, type= "raw")

confusionMatrix(pred, as.factor(testDF$Class))

img\_pred <- predict(img, model=rf, na.rm=T)

###Plot the image in R###

levels(img\_pred)

levelplot(img\_pred,col.regions = c("white","red","yellow","dark green","light

green","orange","blue"),main = "Supervised Classification")

###########################Write Output Grid of the classified image#################

setwd("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/RF\_Winter/Outputs2")

writeRaster(r3, filename="Distinct\_RF\_Winter2.tif", format="GTiff", overwrite=TRUE)

##########################END OF SCRIPT###########################

Summer RF:

###Random Forest Classification- to spectrally discriminate various landcover/vegetation of Du

Toits Wetland###

###Based on code developed by Blessing Khavu, provided by Blessing Khavu as part of postgrad

research group peer learning####

###Spectral signature graph code developed by Daniëlle Seymour###

###This script uses the 7 distinct veg classification scheme###

###This is RF script for Summer 2020/2021###Sentinel-2A, MSI Level2-A###

###Imagery processed in GEE: composite spans '2020-12-01','2021-02-28'###

###Load packages###

library(rgdal)

library(raster)

library(rasterVis)

library(caret)

library(randomForest)

library(e1071)

############################Load Image, create list and brick raster###################

####List band and then brick the raster: ###Raster stack vs brick: how they store each band is

different. The bands in a RasterStack are stored as links to raster data that is located somewhere on

our computer. ###A RasterBrick contains all of the objects stored within the actual R object. In

most cases, we can work with a RasterBrick in the same way we might work with a RasterStack.

###However a RasterBrick is often more efficient and faster to process - which is important when

working with larger files.

rm(list=ls())

setwd("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/RF\_Summer/Imagery\_Summer")

#First import all files in a single folder as a list

rastlist1 <- list.files(path =

"C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/RF\_Summer/Imagery\_Summer",

pattern='.tif', all.files=TRUE, full.names=FALSE)

img<-

brick("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/RF\_Summer/Imagery\_Summer/S2

SummerCompRepro.tif")

names(img) <- c('B1', 'B2', 'B3', 'B4', 'B5', 'B6','B7', 'B8','B8A','B9','B10','B11', 'B12','AOT','WVP',

'SCL','TCI\_R', 'TCI\_G','TCI\_B','MSK\_CLDPRB','MSK SNWPRB', 'QA60', 'QA60 Bitmask')

names(img) <- paste0("B", c(1:23))

########################Plotting RGB image#####################################

plotRGB(img \* (img >= 0), r = 4, g = 3, b = 2, scale = 10000)

#######################Load shapefile with class coordinates and class name i.e. WV, SWV,

Bare soil/sandstone, Water and Unknown####################

#######################You can add it along with values of each band or do it in R###

trainData <-

shapefile("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/TrainingData\_shp5/DistinctTra

ining5.shp")

################Extract values to the shapefile that you just loaded###########

#beginCluster() #to optimize all cores

roi\_data <- extract(img, trainData, df=TRUE)

137

roi\_data$Class <- as.factor(trainData$Class[roi\_data$ID])

roi\_data <- roi\_data[roi\_data$Class!="0",]

head(roi\_data)

###Second option of extraction for signature plot###

roi\_data <- extract(img, trainData, df=TRUE)

head(roi\_data)

summary(roi\_data)

###Create signature plot###

specs <- aggregate(roi\_data, list(trainData$Class), mean, na.rm=TRUE)

specs

# instead of the first column, use row names

rownames(specs) <- specs[,1]

specs <- specs[,-1]

specs

setwd("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest")

write.csv(specs, "SpecsDistinct2.csv")

#Create a vector of color for the land cover classes for use in plotting

mycolor <- c('grey','red','yellow','dark green','light green','orange','blue')

#transform ms from a data.frame to a matrix

specs <- as.matrix(specs)

# First create an empty plot

plot(0, ylim=c(0,4000), xlim = c(1,12), type='n', xlab="Bands", ylab = "Reflectance")

##add the different classes

for (i in 1:nrow(specs)){

lines(specs[i,], type = "l", lwd = 3, lty = 1, col = mycolor[i])

}

#Title

title(main="Spectral Profile of Distinct Classes-S2A", font.main = 2)

#Legend

legend("topleft", rownames(specs),

cex=0.8, col=mycolor, lty = 1, lwd =3, bty = "n")

###########Set seed to make sure the same random sample is selected next time########

set.seed(200)

###Note: seed---Random number seed to use. If a value is provided, it will be used to initialize R's

random number generator before the model is fitted. ###If a value is not provided (the default), the

random number generator will be initialized from the current time.

#################Split the data set into test and training data set###################

splitIndex <- createDataPartition(roi\_data$Class,

p = .80,

list = FALSE,

times = 1)

trainDF <- roi\_data[ splitIndex,]

testDF <- roi\_data[-splitIndex,]

trainDF <- na.omit(trainDF)

#########################Load the Column names. Edit them if you wish################

trainDF <- trainDF[, c('Class', "B1", "B2", "B3", "B4", "B5", "B6","B7", "B8","B9","B10","B11",

"B12")]

138

testDF <- testDF[, c('Class', "B1", "B2", "B3", "B4", "B5", "B6","B7", "B8","B9","B10","B11",

"B12")]

###############Build the Random Forest Model with Training Data############

Fitcontrol <- trainControl("repeatedcv",

number=10,

repeats=1)

rf <- train(as.factor(Class) ~.,

data = trainDF,

method= "rf",

trControl = Fitcontrol,

preProcess = c("center", "scale"),

importance = TRUE)

predict(rf)

########Print the model summary from the Random Forest model##########

print(rf)

#######Check for the Variable importance####################

plot(varImp(rf,scale=FALSE))

#########Predict to a new Dataframe for a Map Output##################

pred <- predict(rf , newdata = testDF, type= "raw")

confusionMatrix(pred, as.factor(testDF$Class))

img\_pred <- predict(img, model=rf, na.rm=T)

####################Plot the image in R###########################################

levels(img\_pred)

levelplot(img\_pred,col.regions = c("white","red","yellow","dark green","light

green","orange","blue"),main = "Supervised Classification")

#3x3 mean filter

r3 <- focal(img\_pred, w=matrix(1/9,nrow=3,ncol=3), median)

###########################Write Output Grid of the classified image#################

setwd("C:/Thesis\_2021/Results\_Mapwork/R/R\_Random\_Forest/RF\_Summer/Outputs4")

writeRaster(r3, filename="Distinct\_RF\_Summer2.tif", format="GTiff", overwrite=TRUE)

############################END OF SCRIPT###############################

####Script for SVM- comparison to RF script of Du Toits River Wetland##

###Code developed by Daniëlle Seymour####

###SVM script adapted form Random Forest Classification- to spectrally discriminate various landcover/vegetation of Du Toits Wetland###

####This script uses the 7 distinct veg classification scheme as in GEE script####

####This is RF script for Winter 2020###Sentinel-2A,MSI Level2-A####

###Imagery processed in GEE: composite spans '2020-06-01','2020-08-31'####

#####Load packages####

library(rgdal)

library(raster)

library(rasterVis)

library(caret)

library(kernlab)

library(e1071)

############################Load Image, create list and brick raster###################

####List band and then brick the raster:

###Raster stack vs brick: how they store each band is different. The bands in a RasterStack are stored as links to raster data that is located somewhere on our computer.

###A RasterBrick contains all of the objects stored within the actual R object. In most cases, we can work with a RasterBrick in the same way we might work with a RasterStack.

###However a RasterBrick is often more efficient and faster to process - which is important when working with larger files.

rm(list=ls())

setwd("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/R\_Random\_Forest/RF\_Winter/Imagery\_Winter")

#first import all files in a single folder as a list

rastlist1 <- list.files(path="//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/R\_Random\_Forest/RF\_Winter/Imagery\_Winter", pattern='.tif', all.files=TRUE, full.names=FALSE)

img<- brick("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/R\_Random\_Forest/RF\_Winter/Imagery\_Winter/S2WinterCompRepro.tif")

names(img) <- c('B1', 'B2', 'B3', 'B4', 'B5', 'B6','B7', 'B8','B8A','B9','B10','B11', 'B12','AOT','WVP', 'SCL','TCI\_R', 'TCI\_G','TCI\_B','MSK\_CLDPRB','MSK SNWPRB', 'QA60', 'QA60 Bitmask')

names(img) <- paste0("B", c(1:23))

####Add NDVI to imagery###

#ndvi = ((img$B8-img$B4)/(img$B8+img$B4))

#names(ndvi)= c('NDVI')

#img = addLayer(img, ndvi)

#plot(ndvi)

#setwd("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/H")

#writeRaster(ndvi, filename="///146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/H/Sentinel\_NDVI\_Summer.tif", format="GTiff", overwrite=TRUE)

##############################Plotting RGB image#####################################

##plotRGB(img \* (img >= 0), r = 4, g = 3, b = 2, scale = 10000)

################Load shapefile with class coordinates and class name i.e.WV,SWV,Bare soil/sandstone, Water and Unknown####################

################You can add it along with values of each band or do it in R###

trainData <- shapefile("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/R\_Random\_Forest/TrainingData\_shp5/DistinctTraining5.shp")

#######################Extract values to the shapefile that you just loaded###########

#beginCluster() #to optimize all cores

roi\_data <- extract(img, trainData, df=TRUE)

roi\_data$Class <- as.factor(trainData$Class[roi\_data$ID])

roi\_data <- roi\_data[roi\_data$Class!="0",]

head(roi\_data)

###################

###Second option of extraction for signature plot###

#roi\_data <- extract(img, trainData, df=TRUE)

#head(roi\_data)

#summary(roi\_data)

#################Split the data set into test and training data set###################

#####################################################################################

splitIndex <- createDataPartition(roi\_data$Class,

p = .80,

list = FALSE,

times = 1)

trainDF <- roi\_data[ splitIndex,]

testDF <- roi\_data[-splitIndex,]

trainDF <- na.omit(trainDF)

#####################################################################################

#########################Load the Column names. Edit them if you wish################

#####################################################################################

trainDF <- trainDF[, c('Class', "B1", "B2", "B3", "B4", "B5", "B6","B7", "B8","B9","B10","B11", "B12")]

testDF <- testDF[, c('Class', "B1", "B2", "B3", "B4", "B5", "B6","B7", "B8","B9","B10","B11", "B12")]

#########################Build the SVM Model with Training Data############

##SVM Linear piece of code:

trctrl <- trainControl(method = "repeatedcv",

number = 10,

repeats = 3)

svm\_Linear <- train(Class~ ., data = trainDF,

method = "svmLinear",

trControl=trctrl,

preProcess = c("center", "scale"),

tuneLength = 10)

predict(svm\_Linear)

print(svm\_Linear)

pred <- predict(svm\_Linear , newdata = testDF, type= "raw")

img\_pred <- predict(img, model=svm\_Linear, na.rm=T)

#######################Plot the image in R###########################################

levels(img\_pred)

levelplot(img\_pred,col.regions = c("white","red","yellow","dark green","light green","orange","blue"),main = "Supervised Classification\_SVM\_Linear")

# 3x3 mean filter

r3 <- focal(img\_pred, w=matrix(1/9,nrow=3,ncol=3), median)

setwd("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/H/SVM")

writeRaster(r3, filename="Distinct\_SVMLinear\_Winter.tif", format="GTiff", overwrite=TRUE)

##############

confusionMatrix(pred, as.factor(testDF$Class))##Everything working up until here

#######################End of script#########################################

####Script for SVM- comparison to RF script of Du Toits River Wetland##

##Code developed by Daniëlle Seymour####

###SVM script adapted from Random Forest Classification- to spectrally discriminate various landcover/vegetation of Du Toits Wetland###

####This script uses the 7 distinct veg classification scheme as in GEE script####

####This is SVM script for Summer 2020/2021###Sentinel-2A,MSI Level2-A####

###Imagery processed in GEE: composite spans '2020-12-01','2021-02-28'####

#####Load packages####

library(rgdal)

library(raster)

library(rasterVis)

library(caret)

library(kernlab)

library(e1071)

############################Load Image, create list and brick raster###################

####List band and then brick the raster:

###Raster stack vs brick: how they store each band is different. The bands in a RasterStack are stored as links to raster data that is located somewhere on our computer.

###A RasterBrick contains all of the objects stored within the actual R object. In most cases, we can work with a RasterBrick in the same way we might work with a RasterStack.

###However a RasterBrick is often more efficient and faster to process - which is important when working with larger files.

rm(list=ls())

setwd("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/R\_Random\_Forest/RF\_Summer")

#first import all files in a single folder as a list

rastlist1 <- list.files(path="//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/R\_Random\_Forest/RF\_Summer/Imagery\_Summer", pattern='.tif', all.files=TRUE, full.names=FALSE)

img<- brick("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/R\_Random\_Forest/RF\_Summer/Imagery\_Summer/S2SummerCompRepro.tif")

names(img) <- c('B1', 'B2', 'B3', 'B4', 'B5', 'B6','B7', 'B8','B8A','B9','B10','B11', 'B12','AOT','WVP', 'SCL','TCI\_R', 'TCI\_G','TCI\_B','MSK\_CLDPRB','MSK SNWPRB', 'QA60', 'QA60 Bitmask')

names(img) <- paste0("B", c(1:23))

####Add NDVI to imagery###

#ndvi = ((img$B8-img$B4)/(img$B8+img$B4))

#names(ndvi)= c('NDVI')

#img = addLayer(img, ndvi)

#plot(ndvi)

#setwd("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/H")

#writeRaster(ndvi, filename="///146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/H/Sentinel\_NDVI\_Summer.tif", format="GTiff", overwrite=TRUE)

##############################Plotting RGB image#####################################

##plotRGB(img \* (img >= 0), r = 4, g = 3, b = 2, scale = 10000)

################Load shapefile with class coordinates and class name i.e.WV,SWV,Bare soil/sandstone, Water and Unknown####################

################You can add it along with values of each band or do it in R###

trainData <- shapefile("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/R\_Random\_Forest/TrainingData\_shp5/DistinctTraining5.shp")

#######################Extract values to the shapefile that you just loaded###########

#beginCluster() #to optimize all cores

roi\_data <- extract(img, trainData, df=TRUE)

roi\_data$Class <- as.factor(trainData$Class[roi\_data$ID])

roi\_data <- roi\_data[roi\_data$Class!="0",]

head(roi\_data)

###################

###Second option of extraction for signature plot###

#roi\_data <- extract(img, trainData, df=TRUE)

#head(roi\_data)

#summary(roi\_data)

#################Split the data set into test and training data set###################

#####################################################################################

splitIndex <- createDataPartition(roi\_data$Class,

p = .80,

list = FALSE,

times = 1)

trainDF <- roi\_data[ splitIndex,]

testDF <- roi\_data[-splitIndex,]

trainDF <- na.omit(trainDF)

#####################################################################################

#########################Load the Column names. Edit them if you wish################

#####################################################################################

trainDF <- trainDF[, c('Class', "B1", "B2", "B3", "B4", "B5", "B6","B7", "B8","B9","B10","B11", "B12")]

testDF <- testDF[, c('Class', "B1", "B2", "B3", "B4", "B5", "B6","B7", "B8","B9","B10","B11", "B12")]

#########################Build the SVM Model with Training Data############

##SVM Linear piece of code:

trctrl <- trainControl(method = "repeatedcv",

number = 10,

repeats = 3)

svm\_Linear <- train(Class~ ., data = trainDF,

method = "svmLinear",

trControl=trctrl,

preProcess = c("center", "scale"),

tuneLength = 10)

predict(svm\_Linear)

print(svm\_Linear)

pred <- predict(svm\_Linear , newdata = testDF, type= "raw")

img\_pred <- predict(img, model=svm\_Linear, na.rm=T)

#######################Plot the image in R###########################################

levels(img\_pred)

levelplot(img\_pred,col.regions = c("white","red","yellow","dark green","light green","orange","blue"),main = "Supervised Classification\_SVM\_Linear")

# 3x3 mean filter

r3 <- focal(img\_pred, w=matrix(1/9,nrow=3,ncol=3), median)

setwd("//146.232.23.24/geography/Students/Users/Postgrad/Danielle/Results\_Mapwork/R/H/SVM")

writeRaster(r3, filename="Distinct\_SVMLinear\_Summer.tif", format="GTiff", overwrite=TRUE)

##############

confusionMatrix(pred, as.factor(testDF$Class))##

#################End of script#######################

**Appendix D : Soil profiles**

Soil conditions were observed and recorded using a Munsell Colour Chart to visually identify soils that were indicative of wetland conditions vs non-wetland conditions. Relevant literature guides such as the South African Wetland Classification System (Ollis et al. 2013) and the National Wetland Vegetation Database (Sieben, Mtshali & Janks 2014) were consulted to assist in grouping plants into the different classes of vegetation. Where plant identifications were not found in books or literature, photographs were taken in field and searched online using both Google search (websites such as SANBI’s PlantZAfrica <http://pza.sanbi.org/> ), and a plant identifier mobile application called PlantSnap. Photos below were taken in the field of soil auger samples collected within quadrats, demarcated at 50 cm and 100 cm from the surface. Soil colour patterns can provide an indication of the water regime; where soils are well drained (typically mineral soils), there is enough oxygen to oxidize irons in the soil resulting in brown, red, or yellow soil (Richards 2001). Where soils are saturated and anaerobic (devoid of oxygen), iron is leached from the soil and soils become grey, sometimes gleyed depending on the period of saturation (Richards 2001). Lastly, where soils are wetter (especially for longer periods of time), the presence of water reduces the rate of decomposition of organic matter resulting in darker, blacker, and higher organic matter (Job 2014; Richards 2001). If anaerobic soils in wetlands dry up or are drained, one often finds the presence of mottles which are iron oxides that form red or orange spots in the soil. Mottles are useful indicators of drained wetlands after long periods of saturation, and evidence of wetland loss in an area (Job 2014; Richards 2001). Field observations have shown that where peatland vegetation i.e., *Prionium serratum*, *Psoralea pinnata* (fountain bush), and also *Zantedeschia aethiopica* (arum lily) were dominantly present; soils were deeper, wetter (permanently saturated) and darker in colour with higher organic matter -sometimes clay (photos a and b)-as commonly found in peatlands (Job 2014; Sieben 2012). This class is thus referred to as wetland vegetation subtype-1 in this study which is considered to be ‘pure’ wetland habitat and/or peatland conditions. Where soils were damp to dry; sandy to sandy loam; brown, red, and grey in colour (photos c-f), species such *Pteridium aquilinum* (i.e., Bracken fern), *Merxmuellera cincta* (grass),this class was thus considered the Sclerophyllous Wetland Vegetation group i.e., subtype-2 of the overall wetland vegetation in this study as it was characterized by properties belonging partly to wetland and/or upland drier fynbos habitat conditions. The dominant upland Fynbos areas showed much drier, sandier, and coarser soils than in the peat wetland, and sclerophyllous wetland vegetation communities. There were no signs of mottling in any of the soil samples taken in the field.

A picture containing person, outdoor, ground, holding

Description automatically generatedA close-up of a piece of mud

Description automatically generated with medium confidenceA picture containing outdoor, footwear, person, grass

Description automatically generatedA close-up of a shovel

Description automatically generated with medium confidence

d)

c)

b)

a)

A picture containing outdoor, person, plant, ground

Description automatically generatedA close-up of a piece of dirt

Description automatically generated with low confidence

f)

e)

Soil auger profile photos taken in field at quadrats with dominant presence of a) *Prionium serratum* (Palmiet) & *Zantedeschia aethiopic*a (Arum lily); b) *Psoralea pinnata* (fountain bush); c) *Pteridium aquilinum* (i.e., Bracken fern), d) *Merxmuellera cincta* (grass); Fynbos species such as e) *Berzelia abrotanoides* *& Metalasia muricata* and f) *Leucadendron coniferum* with very small fragments of dry palmiet wetland vegetation.

**Appendix E:**

**Codes for ecotone mapping and assessment as applied in R:**

####Accuracy assessment code for the class probability map generated in ArcMap with four vegetation layers###

####Based on code developed by Vernon Visser-provided by Helen de Klerk####

###Load packages:

library(raster)

library(RStoolbox)

library(ggplot2)

library(rfUtilities)

library(sampling)

library(caret)

#Set working directory:

setwd("C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping")

#ArcMap Bayesian classification raster:

classProb = stack("C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping/Class\_Probability2/ClassProb\_Output/ClassifiedImage/4ClassVeg.tif") #PCRS

classProb[classProb>100] = NA #Change all values above 100 to NA

#Function to create binary classification raster based on classProb raster (1= P.serratum ; 2= P.pinnata ; 3= Pteridium\_Restio\_Merx; 4= Fynbos)

classProbPerc = function(perc){

cpRast = classProb[[1]]

cpRast[] = NA

cpRast[ classProb[[1]]<=perc | classProb[[4]]<=perc ] = NA

cpRast[ classProb[[1]]>perc & classProb[[2]]<=perc ] = 1

cpRast[ classProb[[2]]>perc & classProb[[1]]<=perc ] = 2

cpRast[ classProb[[3]]>perc & classProb[[4]]<=perc ] = 3

cpRast[ classProb[[4]]>perc & classProb[[3]]<=perc ] = 4

return(cpRast)

}

#90% threshold

classProb90 = classProbPerc(perc=90)

#plot(classProb90)

#Choose classified raster for accuracy testing:

rastClass = classProb90

#Read in testing data:

accShp = shapefile("C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping/Class\_Probability2/TestSamples2/Test2.shp")

head(accShp)

plot(accShp)

#Select random points:

sampleCells = cellFromPolygon(rastClass, accShp) #Get all possible raster cells overlapped by test polyons

sampleCellClasses = lapply(sampleCells, function(x) rastClass[x]) #Get class values from rastClass raster

sampleCellClasses

#Create list to store observed veg classes:

sampleClasses = list() #Create empty list that will be of same dimensions as "sampleCells" to store veg classes

for(l in 1:length(sampleCells)){

sampleClasses[[l]] = rep(accShp@data$class[l], length(sampleCells[[l]])) #Assign sample classes to empty list

}

#Many of the cells have NA values, either because the classification is uncertain or the training polygons do not overlap the

#classification raster. Below we remove these cells from our possible sampling cells:

sampleCellClassesNA = lapply(sampleCellClasses, function(x) which(is.na(x))) #Find all NA value cells

for(l in 1:length(sampleCells)){

if(length(sampleCellClassesNA[[l]])>0){

sampleCells[[l]] = sampleCells[[l]][-sampleCellClassesNA[[l]]] #Remove all NA value cells

sampleClasses[[l]] = sampleClasses[[l]][-sampleCellClassesNA[[l]]] #Remove all NA value cells

}

}

#whichRemove = which(unlist(lapply(sampleCells, function(x) length(x)))==0)

#sampleCells = sampleCells[-whichRemove] #Remove from list empty elements

#sampleClasses = sampleClasses[-whichRemove] #Remove from list empty elements

#See how many cells available in each veg class:

table(unlist(sampleClasses))

#Put sample data into a dataframe:

sampleDat = data.frame(cellIDs=unlist(sampleCells), class=unlist(sampleClasses))

sampleDat

sum(is.na(sampleDat$class))

#Sample 100 records from each veg class

subSampleCells = strata(sampleDat, stratanames='class', size = c(100,100,100,100), method='srswor')

subSampleCells = getdata(sampleDat,subSampleCells)

#Get predicted veg classes for sample cells (from rastClass raster):

pred = rastClass[subSampleCells$cellIDs]

pred[pred==1] = 'Prionium serratum'

pred[pred==2] = 'Psoralea pinnata'

pred[pred==3] = 'Pteridium\_Restio\_Merx'

pred[pred==4] = 'Fynbos'

#Get observed veg classes for sample cells:

obs = subSampleCells$class

#Confusion matrix:

table(obs, pred)

#Get accuracy:

accuracy(pred, obs)

##############################End of script#####################################

**Code developed for mapping ecotones and their associated fuzzy graphs in R:**

###Ecotone mapping in Du Toits River Wetland###

###Class probability map is based on Random Forest classified images as done in Chapter 3###

###Based on code developed by Vernon Visser-provided by Helen

de Klerk####

###Aim of this script: to map internal wetland ecotones i.e. changes/transition in vegetation composition from 'pure wetland'-Prionium serratum & Psoralea pinnata to sclerophyllous wetland, mixture of wetland grasses, ferns, restios and dryer Fynbos conditions###

###Load packages###

library(rgdal)

library(raster)

library(rasterVis)

library(caret)

library(e1071)

#Set working directory:

setwd("C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping")

#Read in Class Probability classified raster##Classified on 15 Sept 2021 based on RF classified outputs in Chap3, so no additional accuracy measures done##

vegetation = stack("C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping/Class\_Probability2/ClassProb\_Output/ClassifiedImage/4ClassVeg.tif") #PCRS

names(vegetation) <- c(X4ClassVeg.1 = "Prionium\_serratum", X4ClassVeg.2 = "Psoralea\_pinnata",X4ClassVeg.3 = "Pteridium\_Restio\_Merx", X4ClassVeg.4 = "Fynbos")

plot(vegetation)

#Get individual layers:

vegLayer1 = vegetation[[1]]

vegLayer2 = vegetation[[2]]

vegLayer3 = vegetation[[3]]

vegLayer4 = vegetation[[4]]

##Check each layer/class##

#vegLayer1

#vegLayer2

#vegLayer3

#vegLayer4

#Get shapefile of transect coordinates:

transShp = shapefile("C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping/Class\_Probability2/Large transects2/6\_trans\_1\_6kmx200m.shp")

transShp

#Add ID column:

transShp$TRANSECT= 1:nrow(transShp@data)

#Extract values

trans1Dat = extract(vegLayer1, transShp)

transShp$prob1 = trans1Dat #Add values to shapefile

head(trans1Dat)

trans2Dat = extract(vegLayer2, transShp)

transShp$prob2 = trans2Dat

head(trans2Dat)

trans3Dat = extract(vegetation[[3]], transShp)

transShp$prob3 = trans3Dat

head(trans3Dat)

trans4Dat = extract(vegetation[[4]], transShp)

transShp$prob4 = trans4Dat

head(trans4Dat)

#View shapefile data:

transShp@data

#Get individual transects:

for(t in 1:length(transShp$TRANSECT)){

trans = transShp[transShp$TRANSECT==t,]

assign(paste0('trans',t), trans)

rm(trans)

}

plot(trans1) #Plot one of these transects

plot(trans2)

plot(trans3)

plot(trans4)

plot(trans5)

plot(trans6)

plot(transShp)

#Function that will get mean probabilities for each layer (1 to 4) in 50 polygon bins (at approximately every 50 m across the transect)

library(maptools)

library(rgeos)

library(geosphere)

getProbsBins = function(trans){ #trans = transect for which you want to get data, e.g. trans1

coords = trans@polygons[[1]]@Polygons[[1]]@coords #Get all polygon coordinates

#Eastern-most point:

minX = data.frame(matrix(coords[which(coords[,1]==min(coords[,1])),], ncol=2))

minX = minX[1,]

#Western-most point:

maxX = data.frame(matrix(coords[which(coords[,1]==max(coords[,1])),], ncol=2))

maxX = maxX[1,]

#Northern-most point:

maxY = data.frame(matrix(coords[which(coords[,2]==max(coords[,2])),], ncol=2))

maxY = maxY[1,]

#Southern-most point:

minY = data.frame(matrix(coords[which(coords[,2]==min(coords[,2])),], ncol=2))

minY = minY[1,] #Added this in case we get two coordinates that are the max, so we select only one of them

#Assign corners of polygons based on angle from E to W:

if(minY[,1]<maxY[,1]){

UL = minX

LL = minY

UR = maxY

LR = maxX

} else if(minY[,1]>maxY[,1]){

UL = maxY

LL = minX

UR = maxX

LR = minY

}

#Get corner coordinates in UTM projection:

coordinates(UL) = ~X1+X2 #Transform to spatialPoints object

proj4string(UL) <- CRS("+proj=utm +south +zone=34 ellps=WGS84") #Assign projection

UL.latlon = spTransform(UL, CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs')) #Reproject to UTM

coordinates(UR) = ~X1+X2

proj4string(UR) <- CRS("+proj=utm +south +zone=34 ellps=WGS84")

UR.latlon = spTransform(UR, CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs'))

coordinates(LR) = ~X1+X2

proj4string(LR) <- CRS("+proj=utm +south +zone=34 ellps=WGS84")

LR.latlon = spTransform(LR, CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs'))

coordinates(LL) = ~X1+X2

proj4string(LL) <- CRS("+proj=utm +south +zone=34 ellps=WGS84")

LL.latlon = spTransform(LL, CRS('+proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs'))

#Find coordinates of midpoint between UL and LL points:

bearingUL.LL = gzAzimuth(UL.latlon@coords, LL.latlon@coords) #Get bearing between UL and LL

distUL.LL = distGeo(UL.latlon, LL.latlon) #Get distance between UL and LL

midUL.LL.latlon = data.frame(destPoint(p=UL.latlon, b=bearingUL.LL, d=distUL.LL/2)) #Get coordinates of midpoint between UL and LL points

coordinates(midUL.LL.latlon) = ~lon+lat #Transform to spatialPoints object

proj4string(midUL.LL.latlon) <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs") #Assign projection

midUL.LL = spTransform(midUL.LL.latlon, CRS('+proj=utm +south +zone=34 ellps=WGS84')) #Reproject to UTM

#Find coordinates of midpoint between UR and LR points:

bearingUR.LR = gzAzimuth(UR.latlon@coords, LR.latlon@coords)

distUR.LR = distGeo(UR.latlon, LR.latlon)

midUR.LR.latlon = data.frame(destPoint(p=UR.latlon, b=bearingUR.LR, d=distUR.LR/2))

coordinates(midUR.LR.latlon) = ~lon+lat

proj4string(midUR.LR.latlon) <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs")

midUR.LR = spTransform(midUR.LR.latlon, CRS('+proj=utm +south +zone=34 ellps=WGS84'))

#Create 50 polygon bins across the length of transect (polygons are about 200 m wide and 1.6 km long)

distMidUL.LL.midUR.LR = pointDistance(midUL.LL.latlon, midUR.LR.latlon, longlat=T) #Distance between midpoints

distBreaks = distMidUL.LL.midUR.LR/50 #Get break distances (above distance divided by 50)

bearingMidPts = gzAzimuth(midUL.LL.latlon@coords, midUR.LR.latlon@coords) #Get bearing between midpoints

bPolyList = {} #Create empty list for storing polygon bins

for(b in 1:50){ #Loop through all bins while advancing the starting point by "break distance" along the line between the midpoints each time

if(b==1){

bStart = midUL.LL.latlon@coords #For first break, use the midpoint between UL and LL corners

}

bEnd = data.frame(destPoint(p=bStart, b=bearingMidPts, d=distBreaks)) #Calculate coordinates "break distance" along the line between the midpoints

bUL = data.frame(destPoint(p=bStart, b=c(bearingMidPts-90), d=50)) #Calculate UL coordinates of polygon bin

bUR = data.frame(destPoint(p=bEnd, b=c(bearingMidPts-90), d=50)) #Calculate UR coordinates of polygon bin

bLL = data.frame(destPoint(p=bStart, b=c(bearingMidPts+90), d=50)) #Calculate LL coordinates of polygon bin

bLR = data.frame(destPoint(p=bEnd, b=c(bearingMidPts+90), d=50)) #Calculate LR coordinates of polygon bin

bPoly = Polygon(rbind(bUL, bUR, bLR, bLL, bUL)) #Create polygon from above coordinates

bPoly = Polygons(list(bPoly), ID=b) #Create polygon from above coordinates

bPolyList[[b]] = list(bPoly) #Add polygon to polygons list

bPolys = SpatialPolygons(unlist(bPolyList)) #Create multiple-polygon polygon

bStart = bEnd #Reset starting coordinate to be end of last bin

}

proj4string(bPolys) <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs") #Assign projection

bPolysUTM = spTransform(bPolys, CRS('+proj=utm +south +zone=34 ellps=WGS84')) #Reproject to UTM

#Crop the vegetation raster to the extent of the transect in question, and mask:

vegetationT = crop(vegetation, trans)

vegetationT = mask(vegetationT, trans)

#Get probability data:

breakTransVals = extract(vegetationT, bPolysUTM) #Extract probability data from each layer

breakTransMeans = lapply(breakTransVals, function(x) apply(x, MARGIN=2, mean, na.rm=T)) #Calculate mean value for each polygon bin

breakTransSE = lapply(breakTransVals, function(x) apply(x, MARGIN=2, function(x) sd(x, na.rm=T)/sqrt(length(x[!is.na(x)])))) #Calculate standard error value for each polygon bin

#Add data to a data.frame:

breakTransMeansGG = data.frame(brk=rep(c(1:50),4),

layer=rep(c(1:4),each=50),

prob = c(unlist(lapply(breakTransMeans, '[', 1)),

unlist(lapply(breakTransMeans, '[', 2)),

unlist(lapply(breakTransMeans, '[', 3)),

unlist(lapply(breakTransMeans, '[', 4))),

se = c(unlist(lapply(breakTransSE, '[', 1)),

unlist(lapply(breakTransSE, '[', 2)),

unlist(lapply(breakTransSE, '[', 3)),

unlist(lapply(breakTransSE, '[', 4))))

return(breakTransMeansGG)

}

transL = list(trans1, trans2, trans3, trans4, trans5, trans6) #Create list of all transect pointShape objects

breakTransMeansGG = lapply(transL, getProbsBins) #Apply the function above to all of these transect objects

ggDat = do.call(rbind, breakTransMeansGG) #Change format of the results above to get into one dataframe

#test breakTransMeansGG step worked:

breakTransMeansGG

ggDat

#Add a column for the transect number. This also puts the transects in the correct order now:

ggDat$transect = factor(rep(paste0('trans',c(1:length(transShp$TRANSECT))), each=200),

levels=paste0('trans',c(1:length(transShp$TRANSECT))))

##check that the 'order is the same as the original ORIG\_FID

head(ggDat$transect)

#Change percentage to probability:

ggDat$prob = ggDat$prob/100

#Get standard errors:

ggDat$SEupper = ggDat$prob + ggDat$se/100

ggDat$SElower = ggDat$prob - ggDat$se/100

#Plot the results for all transects:

library(ggplot2)

labels = c(trans1 = "Transect 1", trans2 = "Transect 2", trans3 = "Transect 3", trans4 = "Transect 4", trans5 = "Transect 5",

trans6 = "Transect 6")

#New titles for each transect

ggTransects = ggplot(ggDat) + #Specifies the dataset to use (ggDat) and the variables (x=brk, y=prob) and the variables that determine the line colours

geom\_line(aes(brk, prob, colour=factor(layer), group=factor(layer))) + #Specifies it must be a line plot

geom\_line(aes(brk, SEupper, colour=factor(layer), group=factor(layer))) + #Specifies it must be a line plot

geom\_line(aes(brk, SElower, colour=factor(layer), group=factor(layer))) + #Specifies it must be a line plot

labs(x='Distance along transect (m)', y="Probability") + #x- and y-axis labels

facet\_wrap( ~ transect, ncol=1, labeller=labeller(transect = labels)) + #Creates the multiple plot layout, facetting by transect number. You can change the number of plots in each row and column here too.

scale\_color\_manual(values=c("#ff0000","#00FF00","#0000FF","#F28C28" ), name = "", labels = c("Prionium serratum", "Psoralea pinnata","Pter\_Restio\_Merx","Fynbos")) + #Specifies line colours and used for legend editing. I sometimes use http://colorbrewer2.org to choose colours. Here (name = "") specifies there must be no legend header

scale\_y\_continuous(breaks=seq(0,1,0.2)) + #Change breaks along the y-axis

scale\_x\_continuous(breaks=seq(0,100,20), labels=seq(0,3000,600)) + #Change breaks along x-axis and their labels

theme\_bw() + #Changes overall plot colour to black and white theme

theme(strip.background =element\_rect(fill=NA), #Change other elements of the 'theme'. This removes the facet label background colour

axis.text.x = element\_text(angle = 90, hjust = 1)) #This makes the x-axis labels vertical aligned

ggTransects

jpeg('C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping/R\_Outputs/figures/Transects\_graphs2021\_50m.jpg', width=19, units='cm', res=600, height=50) #By changing the width and height you can manipulate how the plot looks (e.g. if labels don't all fit, you can increase the size)

ggTransects

dev.off()

pdf("C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping/R\_Outputs/figures/Transects\_graphs2021\_50m.pdf", width=11.69, height=8.27)

ggTransects

dev.off()

#Plot transect raster:

plot(vegetation)

####Plot individual transects together with their associated maps##

#Loop through each transect and plot:

for(t in 6:length(transShp$TRANSECT)){

ggDatSub = ggDat[ggDat$transect==paste0('trans',t),] #Select only data for transect in question

#Read in map jpeg:

#library(jpeg)

#transImage = readJPEG(paste0('C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping/R\_Outputs/transectsmaps2/Trans1.jpg',t,'.jpg'))

##JPEG not working for me###gives unable to open error…use TIFF instead

library(tiff)

transImage<- readTIFF("C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping/R\_Outputs/transectsmaps2/Trans6.tif", native=TRUE)

#Transform jpeg to raster image for plotting purposes:

library(grid)

g = rasterGrob(transImage, interpolate=FALSE)

#Get probability figure plot:

ggTransectSub = ggplot(ggDatSub) +

geom\_line(aes(brk, prob, colour=factor(layer), group=factor(layer))) + #Mean probability line for each veg type

geom\_ribbon(aes(brk, ymin=SElower, ymax=SEupper, group=factor(layer)), alpha=0.1) + #Standard error shading

labs(title=paste0('Transect ',t), x='Distance along transect (m)', y="Probability") + #Titles

scale\_color\_manual(values=c("#ff0000","#00FF00","#0000FF","#F28C28"), name = "", labels = c("Prionium serratum", "Psoralea pinnata","Pter\_Restio\_Merx","Fynbos")) + #Manual colour selection

scale\_y\_continuous(breaks=seq(0,1,0.2)) + #Manual y-axis tick breaks

scale\_x\_continuous(breaks=seq(0,100,20), labels=seq(0,3000,600)) + #Manual x-axis tick breaks

theme\_bw() + #Black and white plot

theme(strip.background =element\_rect(fill=NA), #Remove background colour of plot title

axis.text.x = element\_text(angle = 90, hjust = 1), #Change angle and position of x-axis labels

axis.text = element\_text(size=5), #Change axis label font size

axis.title = element\_text(size=5), #Change axis title font size

plot.title = element\_text(size=6, face='bold', hjust = 0.5), #Change plot title font size

legend.position="none", #Remove legend

panel.grid.minor = element\_blank(), #Don't show minor grid lines

panel.grid.major = element\_line(size=0.1)) #Change width of major grid lines

#Get map plot:

ggTransImage = qplot(1:10, 1:10, geom="blank") +

annotation\_custom(g, xmin=-Inf, xmax=Inf, ymin=-Inf, ymax=Inf) +

theme(line = element\_blank(),

text = element\_blank(),

title = element\_blank(),

panel.background = element\_blank())

#Arrange the two plots side by side:

library(gridExtra)

grid.arrange(ggTransectSub, ggTransImage, nrow=1)

#Create jpeg image with plots

jpeg(paste0('C:/Thesis\_2021/Results\_Mapwork/R/EcotoneMapping/R\_Outputs/figures2/Transects\_plot',t,'.jpg'), width=11.69, height=4, units='cm', res=600)

grid.arrange(ggTransectSub, ggTransImage, nrow=1, widths=c(1.8,1))

dev.off()

}

####################################End of script############################