Combining continuous and categorical modeling: Digital soil mapping of soil horizons and their depths.

Soil Security Laboratory

2018

The motivation for this chapter is to gain some insights into a digital soil mapping approach that uses a combination of both continuous and categorical attribute modeling. Subsequently, we will build on the efforts of the material in the chapters that dealt with each of these type of modeling approaches separately. There are some situations where, a combinatorial approach might be suitable in a digital soil mapping work flow. An example of such a workflow is in Malone et al. (2015) in regards to the prediction of soil depth. The context behind that approach was that often lithic contact was not achieved during the soil survey activities, effectively indicating soil depth was greater than the soil probing depth (which was 1.5 m). Where lithic contact was found, the resulting depth was recorded. The difficulty in using this data in the raw form was that there were many sites where soil depth was greater than 1.5 m together with actual recorded soil depth measurements. The nature of this type of data is likened to a zero-inflated distribution, where many zero observations are recorded among actual measurements (Sileshi, 2008). In Malone et al. (2015) the zero observations were attributed to soil depth being greater than 1.5m. They therefore performed the modeling in two stages. First modeling involved a categorical or binomial model of soil depth being greater than 1.5 m or not. This was followed by continuous attribute modeling of soil depth using the observations where lithic contact was recorded. While the approach was a reasonable solution, it may be the case that the frequency of recorded measurements is low, meaning that the spatial modeling of the continuous attribute is made under considerable uncertainty, as was the case in Malone et al. (2015) with soil depth and other environmental variables spatially modeled in that study; for example, the frequency of winter frosts.

Another interesting example of a combinatorial DSM work flow was described in Gastaldi et al. (2012) for the mapping of occurrence and thickness of soil profiles. There they used a multinomial logistic model to predict the presence or absence of the given soil horizon class, followed by continuous attribute modeling of the horizon depths. For the purposes a demonstrating the work flow of this combinatorial or two-stage DSM, we will re-visit the approach that is described by Gastaldi et al. (2012) and work through the various steps needed to perform it within `R`.

The data we will use comes from 1342 soil profile and core descriptions from the Lower Hunter Valley, NSW Australia. These data have been collected on an annual basis since 2001 to present. These data are distributed across the $220km^2$ area as shown in Figure 1. The intention is to use these data first to predict the occurrence of given soil horizon classes (following the nomenclature of the Australian Soil Classification (Isbell, 2002)). Specifically we want to prediction the spatial distribution of the occurrence of A1, A2, AP, B1, B21, B22, B23, B24, BC, and C horizons, and then where those horizons occur, predict their depth.

First lets perform some data discovery both in terms of the soil profile data and spatial covariates to be used as predictor variables and to inform the spatial mapping. You will notice the soil profile data `dat` is arranged in a flat file where each row is a soil profile. There are many columns of information which include profile identifier and spatial coordinates. Then there are 11 further columns that are binary indicators of whether a horizon class is present or not (indicated as 1 and 0 respectively). The following 11 columns after the binary columns indicate the horizon depth for the given horizon class.

```
# library
library(ithir)
library(sp)
library(raster)

# data
load("HV_horizons.rda")
str(dat)
```

```
## 'data.frame': 1342 obs. of  25 variables:
##  $ FID : Factor w/ 1342 levels "1","10","100",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ e   : num  338014 338183 341609 341352 339736 ...
##  $ n   : num  6370646 6370550 6370437 6370447 6370439 ...
##  $ A1  : int  1 0 1 1 1 1 1 0 1 1 ...
##  $ A2  : int  1 0 0 0 0 1 1 0 0 0 ...
##  $ AP  : int  0 1 0 0 0 0 0 1 0 0 ...
##  $ B1  : int  0 0 1 1 1 0 0 0 1 0 ...
##  $ B21 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ B22 : int  1 0 1 0 1 1 1 1 1 1 ...
##  $ B23 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ B24 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ B3  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ BC  : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ C   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ A1d : num  21 NA 17 45 20 25 13 NA 10 44 ...
##  $ A2d : num  27 NA NA NA NA 15 13 NA NA NA ...
##  $ APd : num  NA 40 NA NA NA NA NA 35 NA NA ...
##  $ B1d : num  NA NA 25 25 30 NA NA NA 30 NA ...
##  $ B21d: num  26 60 26 30 30 25 58 40 20 38 ...
##  $ B22d: num  26 NA 32 NA 20 20 16 25 25 18 ...
##  $ B23d: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ B24d: int  NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ B3d : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ BCd : int  NA NA NA NA NA NA NA NA 15 NA ...
##  $ Cd  : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
# convert data to spatial object
coordinates(dat) <- ~e + n

# covariates
data(hunterCovariates)
```

At our disposal are a few covariates that have either been derived from a digital elevation model and airborne gamma radiometric survey data. These data are available from the `ithir` package by way of the `hunterCovariates` object. These covariates are all registered to the common spatial resolution of 25m and have been organized together into a `rasterStack`.

```
# covariates
names(hunterCovariates)
```

```
## [1] "AACN"              "Drainage.Index"    "Light.Insolation"
## [4] "TWI"               "Gamma.Total.Count"
```

```
# resolution
res(hunterCovariates)
```

```
## [1] 25 25
```

```
# raster properties
dim(hunterCovariates)
```

```
## [1] 860 675   5
```

For a quick check, lets overlay the soil profile points onto the DEM. You will notice on Figure 1 the area of concentrated soil survey (which represents locations of annual survey) within the extent of a regional scale soil survey across the whole study area.

```
plot(hunterCovariates[["AACN"]])
points(dat, pch = 20)
```

The last preparatory step we need to take is the covariate intersection of the soil profile data, and remove any sites that are outside the mapping extent.

```
# Covariate extract
ext <- extract(hunterCovariates, dat, df = T, method = "simple")
w.dat <- cbind(as.data.frame(dat), ext)

# remove sites with missing covariates
x.dat <- w.dat[complete.cases(w.dat[, 27:31]), ]
```

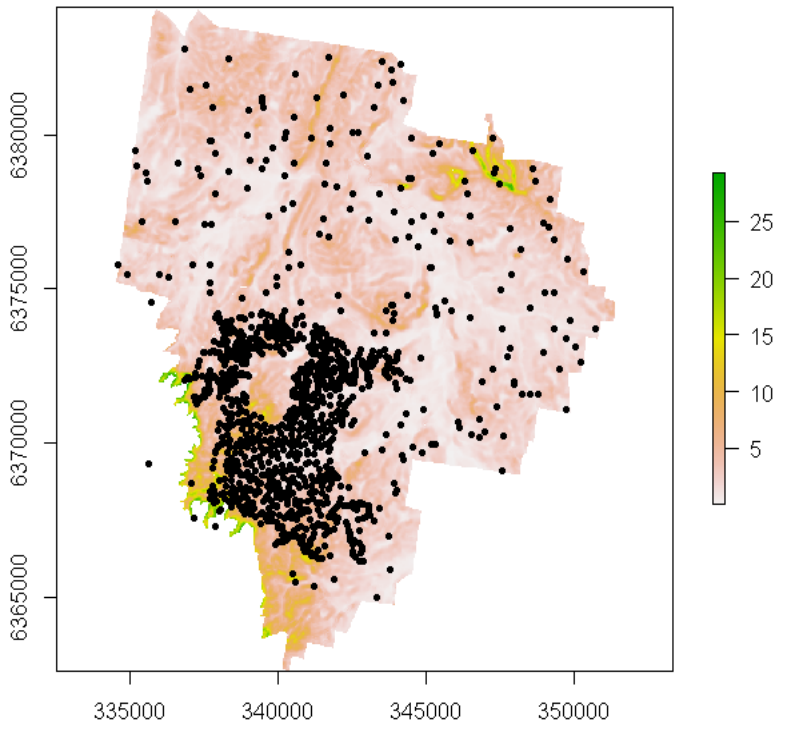

Figure 1: Hunter Valley soil profiles locations overlaying digital elevation model

1 Two-stage model fitting and validation.

A demonstration will be given of the two-stage modeling work flow for the A1 horizon, but given some indication of the results for the other horizons and their depths further on. First we want to subset 75% of the data for calibrating models, and keeping the rest aside for validation purposes.

```
# A1 Horizon
x.dat$A1 <- as.factor(x.dat$A1)
# random subset
set.seed(123)
training <- sample(nrow(x.dat), 0.75 * nrow(x.dat))
# calibration dataset
dat.C <- x.dat[training, ]
# validation dataset
dat.V <- x.dat[-training, ]
```

We first want to model the presence/absence in this case of the A1 horizon.

We will use a multinomial model, followed up with a stepwise regression procedure in order to remove non-significant predictor variables.

```
library(nnet)
library(MASS)

# A1 presence or absence model
mn1 <- multinom(formula = A1 ~ AACN + Drainage.Index + Light.Insolation + TWI +
    Gamma.Total.Count, data = dat.C)

# stepwise variable selection
mn2 <- stepAIC(mn1, direction = "both", trace = FALSE)

summary(mn2)
```

```
## Call:
## multinom(formula = A1 ~ TWI + Gamma.Total.Count, data = dat.C)
##
## Coefficients:
##                         Values   Std. Err.
## (Intercept)        0.498432035 0.510337728
## TWI                0.161779911 0.043292373
## Gamma.Total.Count -0.002239168 0.001316435
##
## Residual Deviance: 782.6183
## AIC: 788.6183
```

We use the `goofcat` function from `ithir` to assess the model quality both in terms of the calibration and validation data.

```
# calibration
mod.pred <- predict(mn2, newdata = dat.C, type = "class")
goofcat(observed = dat.C$A1, predicted = mod.pred)
```

```
## $confusion_matrix
##     0   1
## 0   0   0
## 1 137 861
##
## $overall_accuracy
## [1] 87
##
## $producers_accuracy
##   0   1
##   0 100
##
## $users_accuracy
##   0   1
## NaN  87
##
## $kappa
## [1] 0
```

```
# validation
val.pred <- predict(mn2, newdata = dat.V, type = "class")
goofcat(observed = dat.V$A1, predicted = val.pred)
```

```
## $confusion_matrix
##    0   1
## 0  0   0
## 1 45 288
##
## $overall_accuracy
## [1] 87
##
## $producers_accuracy
##   0   1
##   0 100
##
## $users_accuracy
##   0   1
## NaN  87
##
## $kappa
## [1] 0
```

It is clear that the `mn2` model is not too effective for predicting sites where the A1 horizon is absent.

What we want to do now is to model the A1 horizon depth. We will be using an alternative model to those that have been examined in this book so far. The is a quantile regression forest, which is a generalized implementation of the random forest model from Breiman (2001). The algorithm is available via the `quantregForest` package, and further details about the model can be found at Meinshausen (2006).The `caret` package also interfaces with this model too. Fundamentally, random forests are integral to the quantile regression algorithm. However, the useful feature and advancement from normal random forests is the ability to infer the full conditional distribution of a response variable. This facility is useful for building non-parametric prediction intervals for a any given level of confidence information and also the ability to detect outliers in the data easily. Quantile regression used via the `quanregForest` algorithm is implemented in the chapter simply to demonstrate the wide availability of prediction models and machine learning methods that can be used in digital soil mapping.

Getting the model initiated we first need to perform some preparatory tasks. Namely the removal of missing data from the available data set.

```
# Remove missing values calibration
mod.dat <- dat.C[!is.na(dat.C$A1d), ]

# validation
val.dat <- dat.V[!is.na(dat.V$A1d), ]
```

It is useful to check the inputs required for the quantile regression forests

(using the help file); however its parameterization is largely similar to other models that have been used already in this book, particularly those for the random forest models.

```
# Fit quantile regression forest
library(quantregForest)
qrf <- quantregForest(x = mod.dat[, 27:31], y = mod.dat$A1d, importance = TRUE)
```

Naturally, the best test of the model is to use an external data set. In addition to our normal validation procedure we can also derive the PICP for the validation data too.

```
## Calibration
quant.cal <- predict(qrf, newdata = mod.dat[, 27:31], all = T)
goof(observed = mod.dat$A1d, predicted = quant.cal[, 2])
```

```
##          R2 concordance      MSE    RMSE       bias
## 1 0.8302768   0.8384576 32.05473 5.66169 -0.9436702
```

```
# Validation
quant.val <- predict(qrf, newdata = val.dat[, 27:31], all = T)
goof(observed = val.dat$A1d, predicted = quant.val[, 2])
```

```
##            R2 concordance      MSE     RMSE       bias
## 1 0.008559143  0.06093924 114.5833 10.70436 -0.6666667
```

```
# PICP
sum(quant.val[, 1] <= val.dat$A1d & quant.val[, 2] >= val.dat$A1d)/nrow(val.dat)
```

```
## [1] 0.4826389
```

Based on the outputs above, the calibration model seems a reasonable outcome for the model, but is proven to be largely un-predictive for the validation data set. We should also be expecting a PICP close to 90%, but this is clearly not the case above.

What has been covered above for the two-stage modeling is repeated for all the other soil horizons, with the main results displayed in Table 1. These statistics are reported based on the validation data. It clear that there is a considerable amount of uncertainty overall in the various soil horizon models. For some horizons the results are a little encouraging; for example the model to predict the presence of a BC horizon is quite good. It is clear however that distinguishing between different B2 horizons is challenging. However predicting the presence or absence of a B22 horizons seems acceptable.

Another way to assess the quality of the two-stage modeling is to assess first the number of soil profile that have matching sequences of soil horizon types. We can do this using:

```
vv.dat <- read.table(file = "validation_outs.txt",
    sep = ",", header = T)
dat.V <- read.table(file = "validation_obs.txt",
    sep = ",", header = T)
```

Table 1: Selected model validation diagnostics returned for each horizon class and associated depth model.

	Presence/Absence of Horizon			Depth of Horizon		
Horizon	**Overall Accuracy**	**User's Accuracy**	**Kappa Statistic**	**Concordance**	**RMSE**	**PICP**
A1	87%	Pres = 89% Abs = 54%	0.19	0.05	10	46%
A2	87%	Pres = 100% Abs = 87%	0.04	0.10	12	42%
AP	86%	Pres = 50% Abs = 88%	0.15	0.00	12	53%
B1	91%	Pres = 0% Abs = 91%	0	0.16	12	45%
B21	97%	Pres = 97% Abs = 0%	0	0.05	17	41%
B22	73%	Pres = 73% Abs = 34%	0	0.10	14	41%
B23	78%	Pres = 0% Abs = 78%	0	0.04	12	45%
B24	97%	Pres = 0% Abs = 97%	0	0.00	22	46%
BC	74%	Pres = 68% Abs = 75%	0.20	0.06	18	29%
C	95%	Pres = 0% Abs = 95%	0	0	NA	68%

```
# Validation data horizon observations (1st 3 rows)
dat.V[1:3, c(1, 4:14)]
```

```
##    FID A1 A2 AP B1 B21 B22 B23 B24 B3 BC C
## 1  101  1  0  0  1   1   0   0   0  0  0 0
## 2 1022  0  0  1  0   1   1   0   0  0  0 0
## 3 1026  1  0  0  0   1   1   0   0  0  0 0
```

```
# Associated model predictions (1st 3 rows)
vv.dat[1:3, 1:12]
```

```
##   dat.V.FID a1 a2 ap b1 b21 b22 b23 b24 b3 bc c
## 1       101  1  0  0  0   1   1   0   0  0  0 0
## 2      1022  1  0  0  0   1   1   0   0  0  0 0
## 3      1026  1  0  0  0   1   1   0   0  0  0 0
```

```
# matched soil profiles
sum(dat.V$A1 == vv.dat$a1 & dat.V$A2 == vv.dat$a2 & dat.V$AP == vv.dat$ap &
    dat.V$B1 == vv.dat$b1 & dat.V$B21 == vv.dat$b21 & dat.V$B22 == vv.dat$b22 &
    dat.V$B23 == vv.dat$b23 & dat.V$B24 == vv.dat$b24 & dat.V$B3 == vv.dat$b3 &
    dat.V$BC == vv.dat$bc & dat.V$C == vv.dat$c)/nrow(dat.V)
```

```
## [1] 0.2222222
```

The result above indicates that just over 20% of validation soil profiles have matched sequences of horizons. We can examine visually a few of these matched profiles to examine whether there is much coherence in terms of observed and associated predicted horizon depths. We will select out two soil profiles: One with an AP horizon, and the other with an A1 horizon. We can demonstrate this using the `aqp` package, which is a dedicated `R` package for handling soil profile data collections.

```
# Subset of matching data (observations)
match.dat <- dat.V[which(dat.V$A1 == vv.dat$a1 & dat.V$A2 == vv.dat$a2 & dat.V$AP ==
    vv.dat$ap & dat.V$B1 == vv.dat$b1 & dat.V$B21 == vv.dat$b21 & dat.V$B22 ==
    vv.dat$b22 & dat.V$B23 == vv.dat$b23 & dat.V$B24 == vv.dat$b24 & dat.V$B3 ==
    vv.dat$b3 & dat.V$BC == vv.dat$bc & dat.V$C == vv.dat$c), ]

# Subset of matching data (predictions)
match.dat.P <- vv.dat[which(dat.V$A1 == vv.dat$a1 & dat.V$A2 == vv.dat$a2 &
    dat.V$AP == vv.dat$ap & dat.V$B1 == vv.dat$b1 & dat.V$B21 == vv.dat$b21 &
    dat.V$B22 == vv.dat$b22 & dat.V$B23 == vv.dat$b23 & dat.V$B24 == vv.dat$b24 &
    dat.V$B3 == vv.dat$b3 & dat.V$BC == vv.dat$bc & dat.V$C == vv.dat$c), ]
```

Now we just select any row where we know there is and AP horizon

```
match.dat[49, ]  #observation
```

```
##     FID      e       n A1 A2 AP B1 B21 B22 B23 B24 B3 BC C A1d A2d APd B1d
## 195 642 338096 6372259  0  0  1  0   1   1   0   0  0  1 0  NA  NA  10  NA
##     B21d B22d B23d B24d B3d BCd Cd  ID totalCount    thppm
## 195   30   15   NA   NA  NA  45 NA 735   446.7597 7.192239
##     Terrain_Ruggedness_Index    slope SAGA_wetness_index      r57      r37
## 195                0.846727 0.697118           13.34301 1.955882 0.794118
##          r32      PC2       PC1      ndvi    MRVBF    MRRTF
## 195 1.542857 -1.89351 -2.239939 -0.076923 0.111123 3.746326
##     Mid_Slope_Positon light_insolation     kperc Filled_DEM drainage_2011
## 195          0.130692         1716.388 0.5863795   142.8293      3.909594
##     Altitude_Above_Channel_Network
## 195                       25.52147
```

```
match.dat.P[49, ]  #prediction
```

```
##     dat.V.FID a1 a2 ap b1 b21 b22 b23 b24 b3 bc c a1d a2d      apd b1.1
## 195       642  0  0  1  0   1   1   0   0  0  1 0  18  16 21.92308   18
##     b21d b22d b23d     b24d b3d bcd cd
## 195   31   27   20 15.41176  NA  32 NA
```

We can see in these two profiles, the sequence of horizons is AP, B21, B22, BC. Now we just need to upgrade the data to a soil profile collection. Using the horizon classes together with the associated depths, we want to plot both soil profiles for comparison. First we need to create a data frame of the relevant data then upgrade to a `soilProfileCollection`, then finally plot. The script below demonstrates this for the soil profile with the AP horizon. The same can be done with the associated soil profile with the A1 horizon. The plot of the is shown on Figure 2.

```
# Horizon classes
H1 <- c("AP", "B21", "B22", "BC")

# Extract horizon depths then combine to create soil profiles
p1 <- c(22, 31, 27, 32)
p2 <- c(10, 30, 15, 45)
p1u <- c(0, (0 + p1[1]), (0 + p1[1] + p1[2]), (0 + p1[1] + p1[2] + p1[3]))
p1l <- c(p1[1], (p1[1] + p1[2]), (p1[1] + p1[2] + p1[3]), (p1[1] + p1[2] + p1[3] +
    p1[4]))
p2u <- c(0, (0 + p2[1]), (+p2[1] + p2[2]), (+p2[1] + p2[2] + p2[3]))
p2l <- c(p2[1], (p2[1] + p2[2]), (p2[1] + p2[2] + p2[3]), (p2[1] + p2[2] + p2[3] +
    p2[4]))

# Upper depths
U1 <- c(p1u, p2u)
# Lower depths
L1 <- c(p1l, p2l)

# Soil profile names
S1 <- c("predicted profile", "predicted profile", "predicted profile",
 "predicted profile",
    "observed profile", "observed profile", "observed profile", "observed profile")

# Random soil colors selected to distinguish between horizons
hue <- c("10YR", "10R", "10R", "10R", "10YR", "10R", "10R", "10R")
val <- c(4, 5, 7, 6, 4, 5, 7, 6)
chr <- c(3, 8, 8, 1, 3, 8, 8, 1)

# Combine all the data
TT1 <- data.frame(S1, U1, L1, H1, hue, val, chr)

# Convert munsell colors to rgb
TT1$soil_color <- with(TT1, munsell2rgb(hue, val, chr))

# Upgrade to soil profile collection
depths(TT1) <- S1 ~ U1 + L1

# Plot
plot(TT1, name = "H1", colors = "soil_color")
title("Selected soil with AP horizon", cex.main = 0.75)
```

Soil is very complex, and while there is a general agreement between observed and associated predicted soil profiles, the power of the models used in this two-stage example has certainly been challenged. Recreating the arrangement of soil horizons together with maintenance of their depth properties is an interesting problem for pedometric studies and one that is likely to be pursued with vigor as better methods become available. The next section will briefly demonstrate a work flow for creating maps that are resultant of this type of modeling framework.

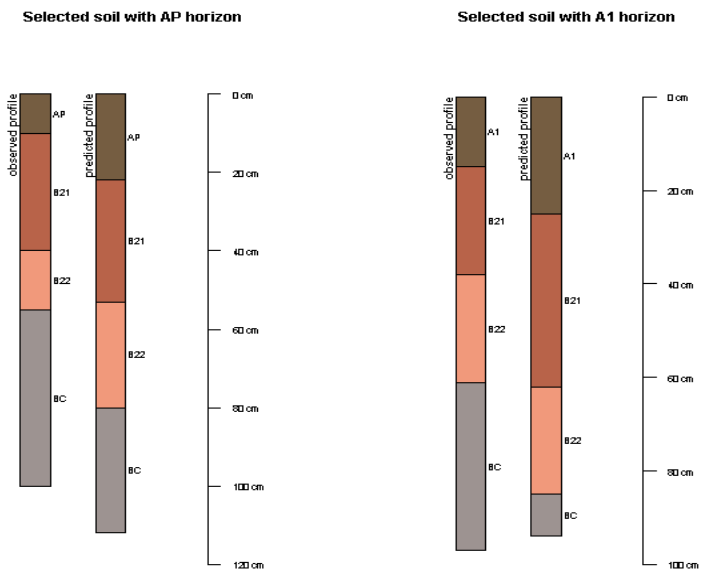

Figure 2: Examples of observed soil profiles with associated predicted profiles from the two-stage horizon class and horizon depth model.

2 Spatial application of the two-stage soil horizon occurrence and depth model

We will recall from previous chapters the process for applying prediction models across a mapping extent. In the case of the two-stage model the mapping work flow if first creating the map of horizon presence/occurrence. Then we apply the horizon depth model. In order to ensure that the depth model is not applied to the areas where a particular soil horizon is predicted as being absent, those areas are masked out. Maps for the presence of the A1 and AP horizons and their respective depths are displayed in Figure 3.The following scripts show the process of applying the two-stage model for the A1 horizon.

```
# Apply A1 horizon presence/absence model spatially Using the raster
# multi-core facility
beginCluster(4)
A1.class <- clusterR(hunterCovariates, predict, args = list(mn2, type = "class"),
    filename = "class_A1.tif", format = "GTiff", progress = "text", overwrite = T)

# Apply A1 horizon depth model spatially Using the raster multi-core
# facility
A1.depth <- clusterR(hunterCovariates, predict, args = list(qrf, index = 2),
```

```
    filename = "depth_A1.tif", format = "GTiff", progress = "text", overwrite = T)
endCluster()

# Mask out areas where horizon is absent
A1.class[A1.class == 0] <- NA
mr <- mask(A1.depth, A1.class)
writeRaster(mr, filename = "depth_A1_mask.tif", format = "GTiff", overwrite = T)
```

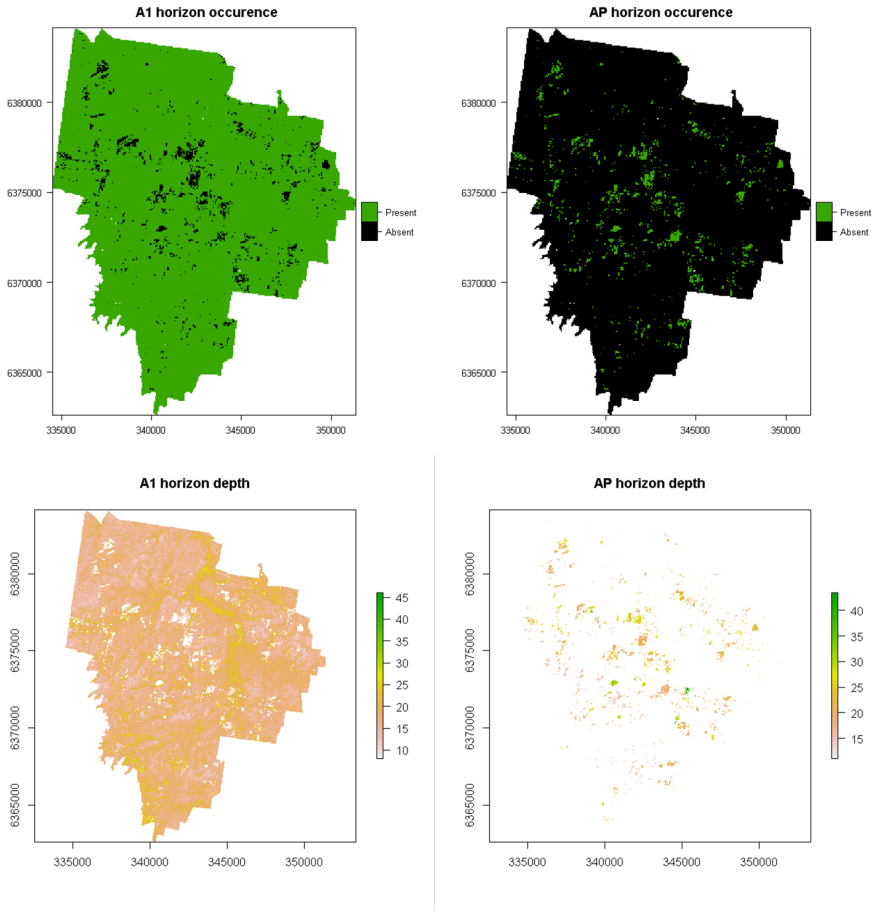

Figure 3: Predicted occurrence of AP and A1 horizons, and their respective depths in the Lower Hunter Valley, NSW.

Figure 3 displays an interesting pattern whereby AP horizons occur where A1 horizons do not. This makes reasonable sense. The spatial pattern of the AP horizon coincides generally with the distribution of vineyards across the study area, where soils are often cultivated and consequently removing the presence of the A1 horizon. Increased depth of A1 horizons also appears to be the case too in lower lying and stream channel catchments of the study area.

References

Breiman, L.
2001. Random forests. *Machine Learning*, 41:5–32.

Gastaldi, G., B. Minasny, and A. B. McBratney
2012. Mapping the occurrence and thickness of soil horizons. In *Digital Soil Assessments and Beyond*, B. Minasny, B. P. Malone, and A. B. McBratney, eds., Pp. 145–148, London. Taylor & Francis.

Isbell, R. F.
2002. *The Australian Soil Classification: Revised Edition*. Collingwood, VIC: CSIRO Publishing.

Malone, B. P., D. B. Kidd, B. Minasny, and A. B. McBratney
2015. Taking account of uncertainties in digital land suitability assessment. *PeerJ*, P. 3:e1366.

Meinshausen, N.
2006. Quantile regression forests. *Journal of Machine Learning Research*, 7:983–999.

Sileshi, G.
2008. The excess-zero problem in soil animal count data and choice of appropriate models for statistical inference. *Pedobiologia*, 52(1):1 – 17.