ACLEP DSM Exchange Training REPORT

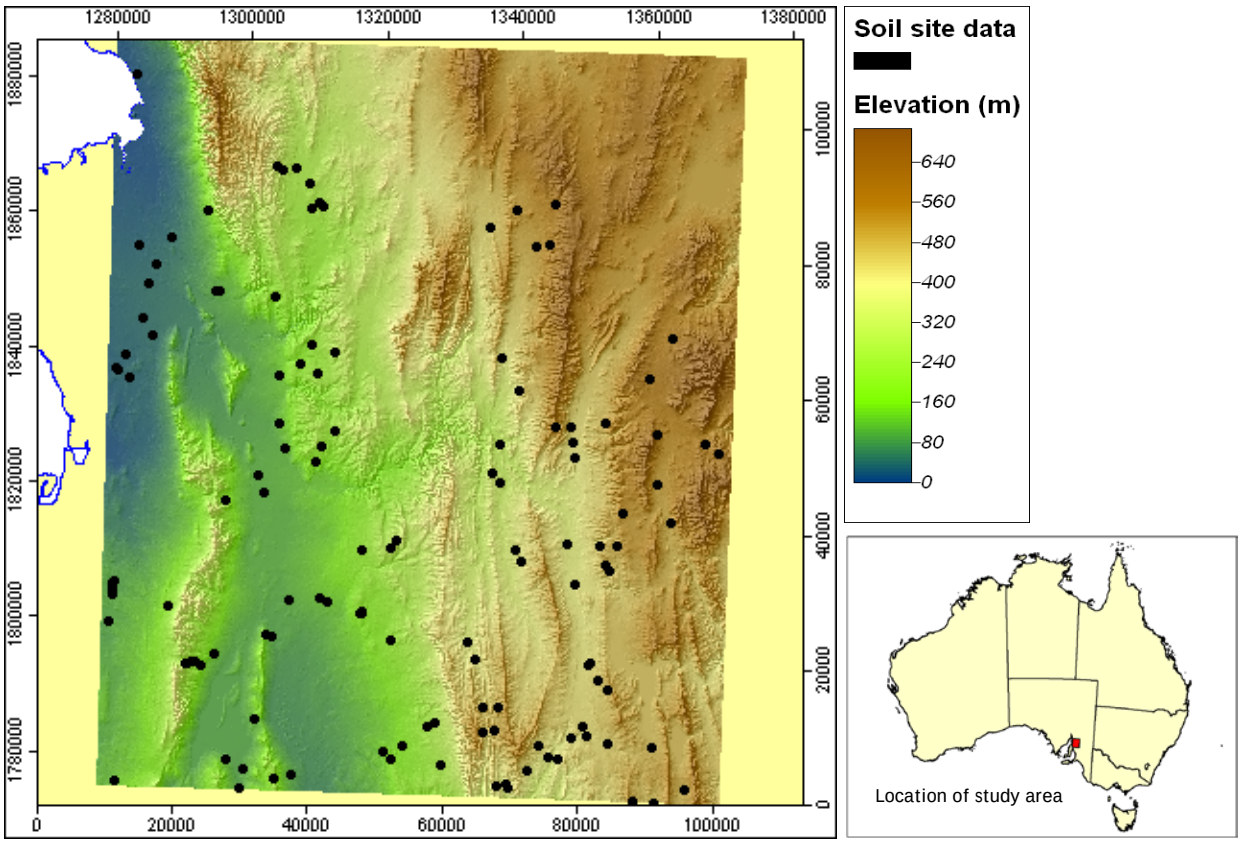
Craig Liddicoat (DEWNR SA), May 2013

Acknowledgements: special thanks to ACLEP and Darren Kidd (DPIPWE Tasmania) respectively for sponsoring and hosting this digital soil mapping (DSM) exchange training, 8-18 Apr 2013 in Launceston Tasmania.

Objectives: were to learn and apply some soil spatial prediction functions (and associated statistical analyses), using R + open-source GIS software, to selected SA soil site datasets.

Approach: Sydney Uni 2012 DSM Workshop exercises were used as a template for developing a number of soil spatial prediction models (e.g. multiple linear regression, regression trees, and cubist models) based on SA soil site data and covariates (raster layers of explanatory environmental variables). Darren Kidd assisted with understanding various parts of the modelling process and statistical analyses. This included showing examples of R code from the Tasmanian DSM work and additional examples developed by Brendon Malone, Sydney Uni.

For this training exercise, legacy site data* for surface soil organic carbon percentage (SOC%) in the fine earth fraction (<2mm) was chosen as the target variable for modelling (*from SA 'Soil Characterisation Sites'). A 1 deg x 1 deg tile within SA's Mid North was chosen as the study area. With 1 arc-second (approx. 30m) cell spacing, each covariate raster layer comprised 10,983,050 cells in total. 131 Soil Characterisation Sites with available SOC% observations were contained in the study area.



Map showing elevation and hillshading across the 1 degree x 1 degree study area.
Black dots indicate the location of 'Soil Characterisation Sites' with available SOC% data.

As part of the training exercise, where possible, Sydney Uni DSM Workshop R scripts were adapted to make more use of the recently evolving R-package 'raster' – as this offered promise to (i) handle more of the modelling steps within R thereby reducing the need to change between software platforms, and (ii) overcome some inherent memory limitations of R and my lower-spec computer through greater utilisation

of hard drive memory and the ability to read/write/process data in blocks. Examples of R code developed using features of the 'raster' package are shown in Appendix B.

In summary, the steps undertaken are summarised below:

1) Collate (and/or generate) and tidy up covariate layers. This includes projecting datasets (to Lamberts Conformal Conic in this case), filling gaps were necessary, clipping to the study area, and resampling to a consistent grid system. This preliminary work was performed in both SAGA-GIS and in R. {Although not done in this training example, other work may include smoothing of 'coarse' datasets}
2) Assemble a 'covariate stack'
3) Sample the covariate stack at point locations (ie. Soil Characterisation Sites)
4) Join soil observations to covariates (potential explanatory variables)
5) Perform preliminary data analysis - e.g. understand basic statistical properties of the data, identify potential erroneous data/ outliers that may need to be excluded, examine the need for data transformations (where modelling assumptions require normally distributed data), examine the efficacy of principal components (where many covariates are correlated), etc
6) Construct predictive model(s) aiming to explain observations in terms of covariates (or principal components of the covariates) with the greatest explanatory value.
7) Consider *metrics of model performance, ideally with separate analyses for 'training' and 'validation' datasets. *Metrics considered in this exercise included R-squared values, Lin's Concordance Coefficient (how well observations and predictions match, ie. correspond to 1:1 line), root mean square error (RMSE), and ratio of performance to deviation (or RPD, which simplifies to standard deviation/RMSE).
8) Apply the model (developed at point locations) across the whole study area (using appropriate layers in the covariate stack)
9) #Uncertainty analysis to report confidence in the model, and hence upper and lower 95th percentile confidence {/prediction?} interval values {# Note: this step was not achieved within the time available, but a possible method may involve determining fuzzy clusters within the covariate feature space, and analysing how residuals relate to the 'distance from cluster centroids'?}

Outcomes: Four models for surface SOC% within the study area were developed:

- a multiple linear regression (stepwise optimised) model
- a simple regression tree model (using 'rpart' package in R)
- a conditional inference regression tree model (using 'ctree' package in R)
- a cubist model

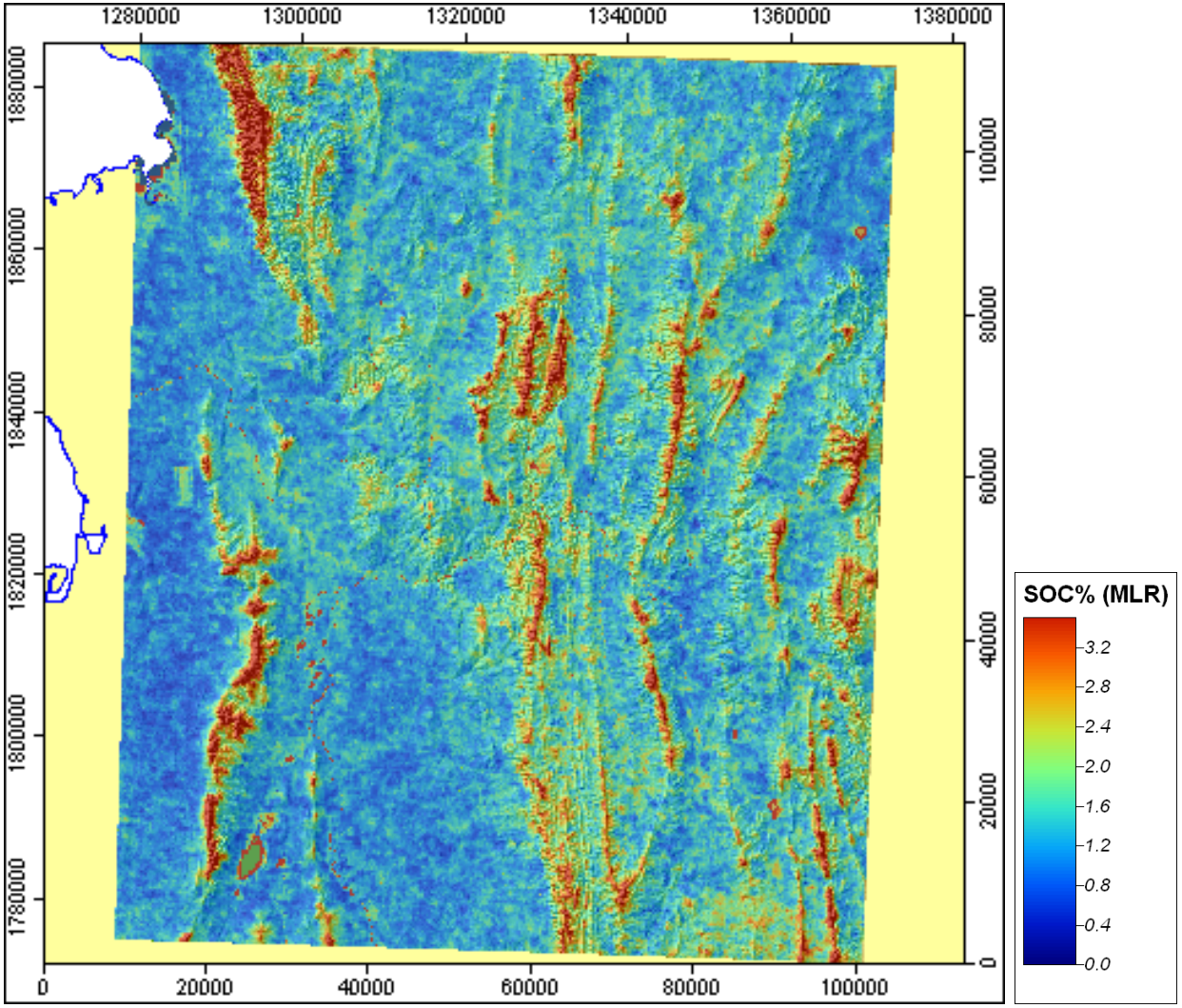
An additional 'ensemble mean' model is also shown.

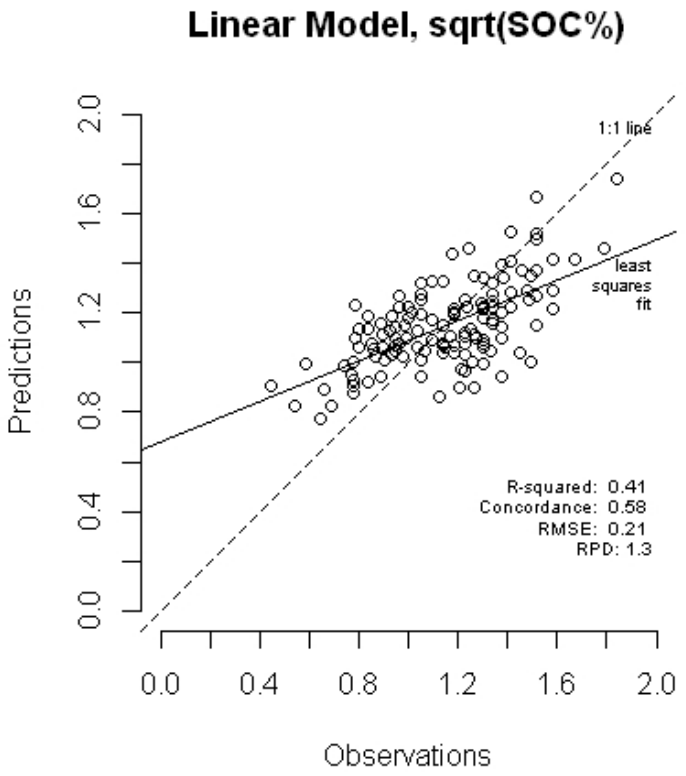
Model prediction maps for surface SOC% are shown on the following pages, together with plots of site observations and model predictions. Plots also display metrics of model performance (r-squared values, Lin's Concordance Correlation Coefficient, RMSE, RPD). In the training examples shown, all sites (n=131) were used to form a 'training dataset' (with no validation dataset). {As per the Sydney Uni workshop exercises, 30% random hold-back and 'leave-one-out' cross-validation techniques were trialled, however for simplicity and for the purpose of generating example model maps and performance metrics, separate 'validation' datasets are not presented in this document.}

Example preliminary data analyses, a listing of covariates used and model summary outputs generated during the training exercises/ model building process are shown in Appendix A.

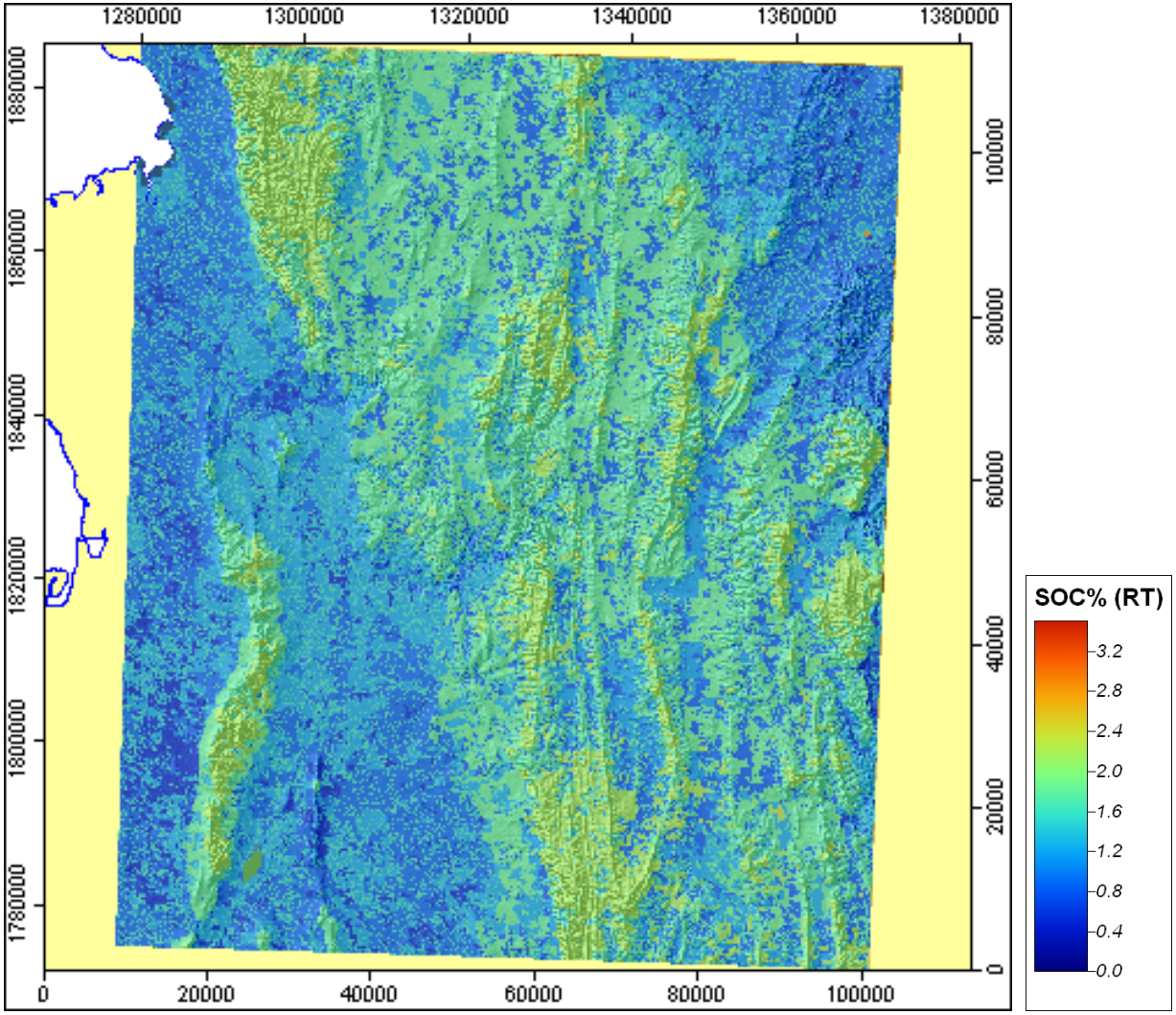
Future work: subsequent future work using soil site data will likely involve:

- Estimating soil carbon stocks (t/ha), based on SOC% (of fine earth fraction), gravel content and bulk density
- Applying depth splines to model various depth intervals
- Building and applying predictive models over larger areas
- Explore new covariate layers, with the aim of improving model performance (e.g. if consistent, finer-scale geology layers can be incorporated)
- Explore the value of transforming skewed covariate datasets (e.g. for legacy site data with likely bias in sampling locations)
- Performing uncertainty analyses (to develop upper and lower estimates)

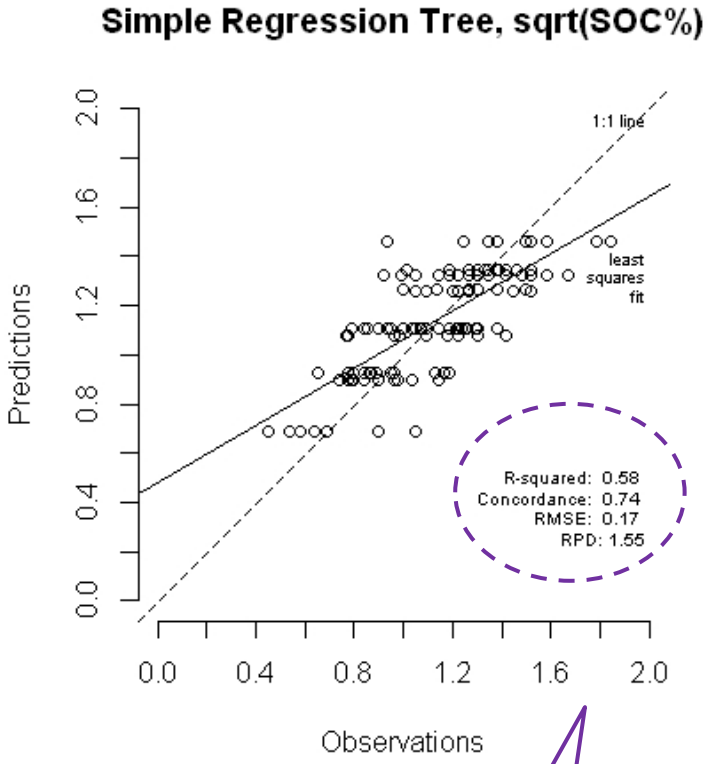




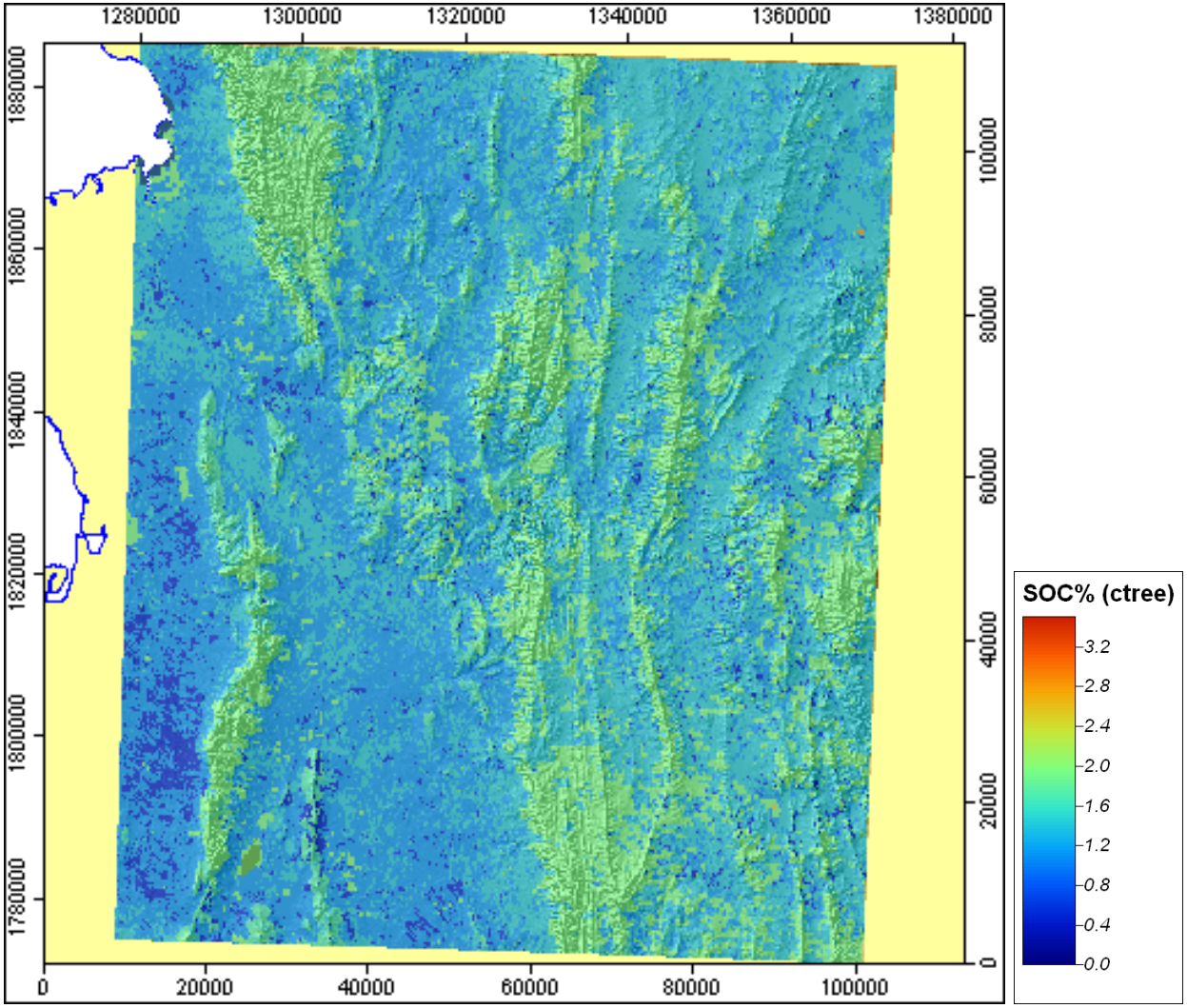
1) Linear Regression Model – Surface SOC%

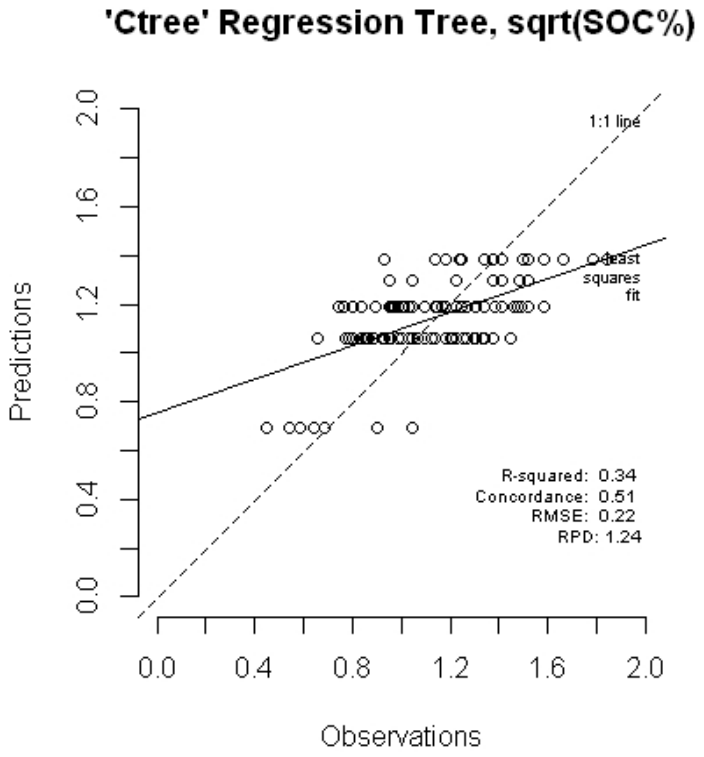


2) Simple Regression Tree – Surface SOC%

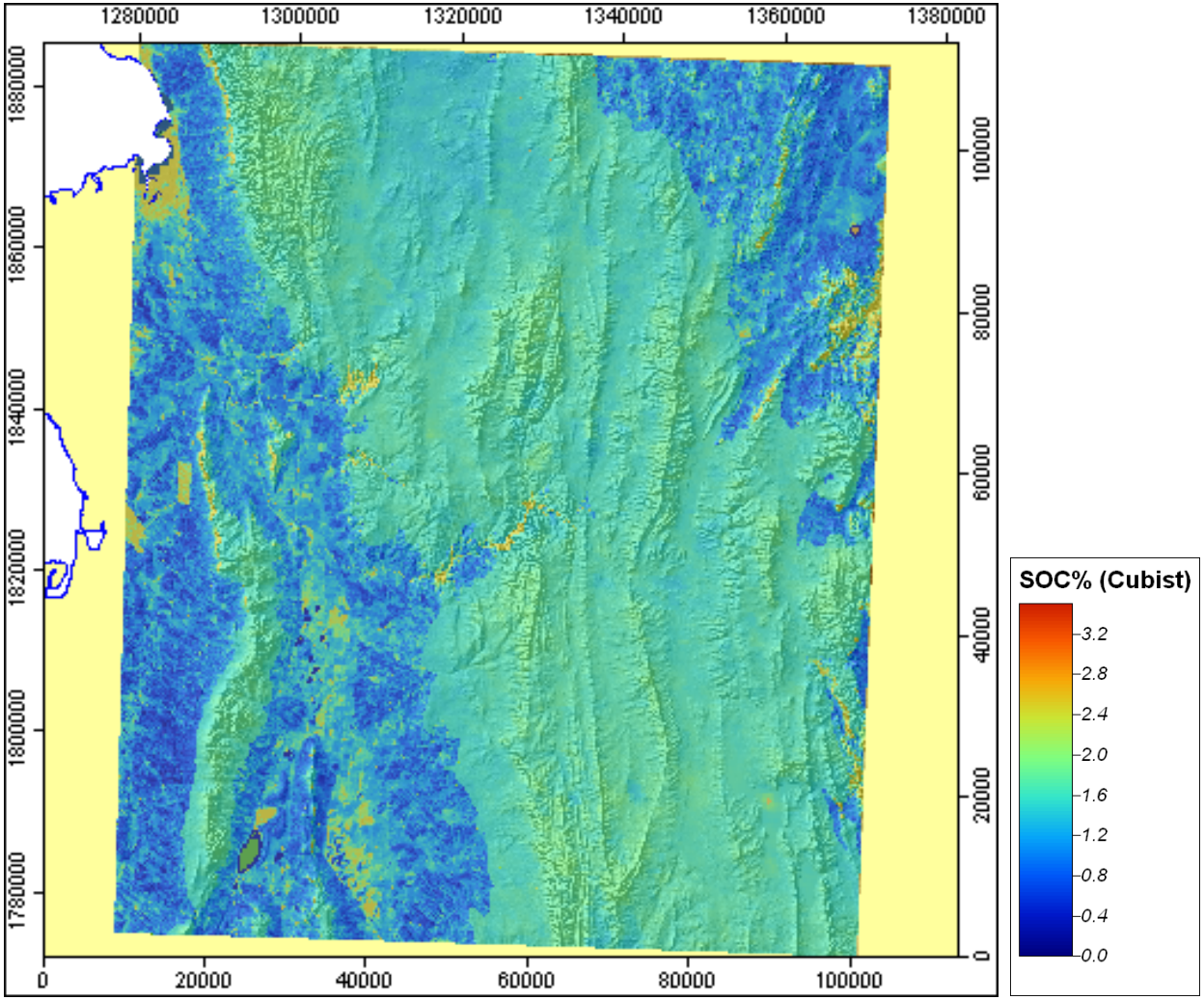


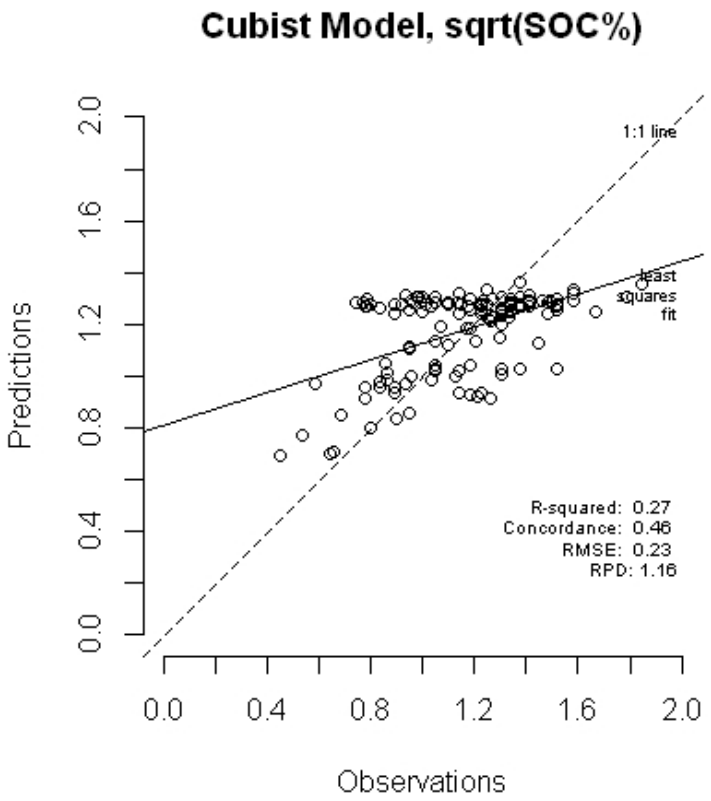
This model shows the best measures (metrics) of model performance



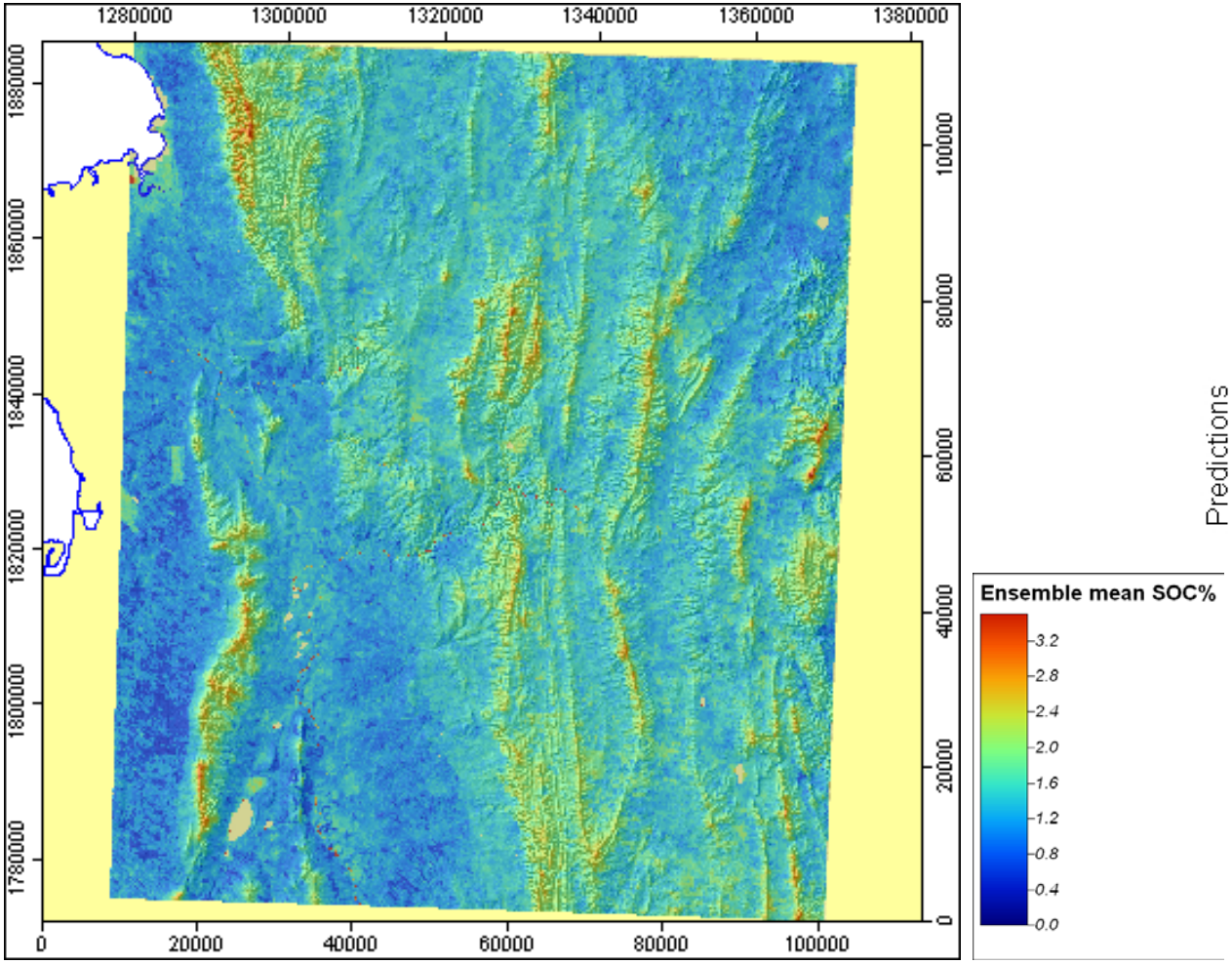


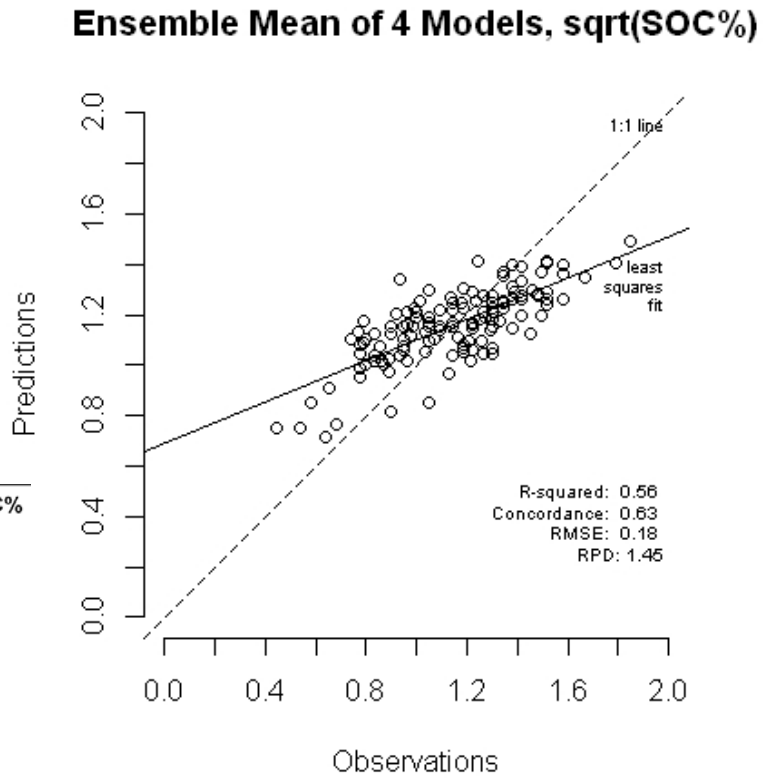
3) 'Ctree' Conditional Inference Tree – Surface SOC%





4) Cubist Model – Surface SOC%





5) Ensemble Mean of the 4 models – Surface SOC%

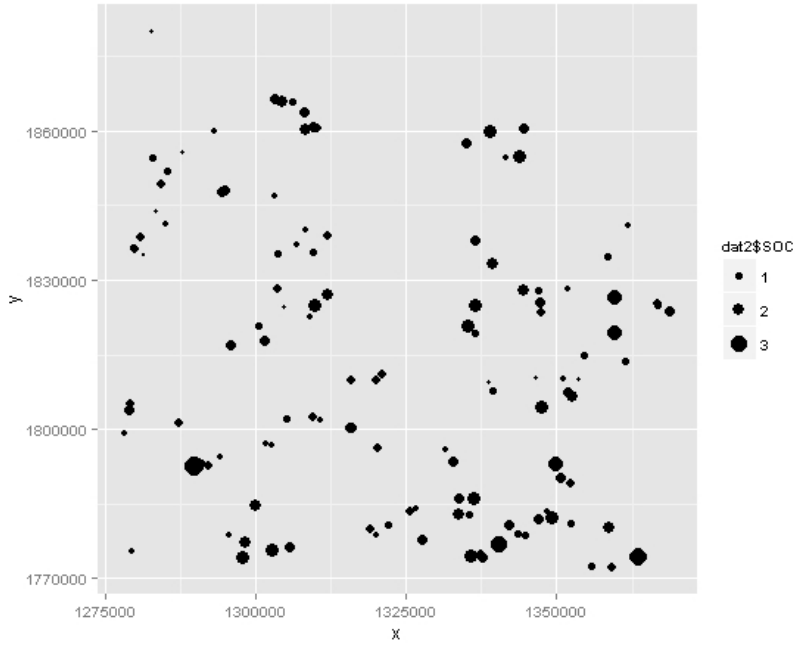
Appendix A. Example data analysis and outputs from the model building process

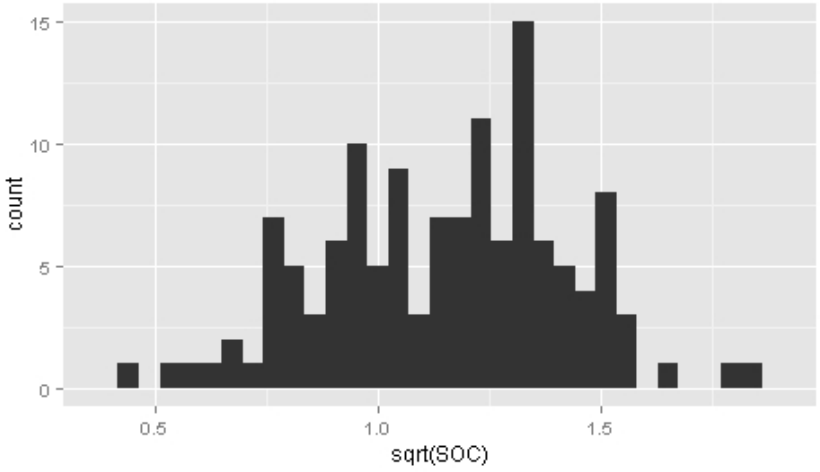
1. Site SOC% data

Distribution of sites and SOC % values are shown (see right) in the mapping domain (Lamberts Conformal Conic Projection, Datum GDA94).

A square-root transformation was applied to the SOC% data - to provide the 'target' variable. This transformation made the target data closer to a normal distribution (see histogram below) – to better conform to subsequent modelling assumptions.

This transformation also reduced the Skewness (-0.109) in the data set, compared to the original distribution. A log transformation did not perform as well as the square-root transformation.





2. Covariates

Covariates were imported from numerous sources including national DEM and terrain derivatives (1 arc-second) and selected 'SCORPAN' covariates assembled by TERN Soil Grid of Australia Facility (Ross Searle pers. comm. 16-May-2013). Additional 1 arc-second terrain derivatives were generated in SAGA-GIS. All covariates were resampled to match the 1 arc-second (~30m resolution) national DEM grid system. 49 covariates in total were included in this training exercise, as below:

```
[7] "hillshading"                  "aspect"
[9] "catchment_area"               "channel_network_base_level"
[11] "Convergence_Index"            "dem_s_1s"
[13] "dem_s_slopepct_fm300"         "dist_to_coast"
[15] "land_type"                    "Landforms"
[17] "landuse06"                    "LS_Factor"
[19] "mrvbf"                        "Plan_Curv"
[21] "rad_k"                        "rad_th"
[23] "rad_u"                        "Weathering_Index"
[25] "Profile_Curv"                 "rainfall_dewnr"
[27] "rel_slope_pos"                "slope"
[29] "TCI_low"                      "TRI"
[31] "TM_2004_b1"                   "TM_2004_b2"
```

```
[33] "TM_2004_b3"                   "TWI"
[35] "Valley_Depth"                 "Veg_FPAR_Mean"
[37] "Veg_FPAR_Median"              "Veg_FPAR_Min"
[39] "Veg_FPAR_StdDev"              "Veg_FractCover_Max_BS"
[41] "Veg_FractCover_Max_PV"        "Veg_FractCover_Mean"
[43] "Veg_FractCover_Mean_NPV"      "Veg_FractCover_Mean_PV"
[45] "Veg_FractCover_Min_NPV"       "Veg_FractCover_Min_PV"
[47] "Veg_FractCover_Std_BS"        "Veg_FractCover_Std_NPV"
[49] "Veg_FractCover_Std_PV"        "Veg_LandCoverTrend_evi_class"
[51] "Veg_LandCoverTrend_evi_max"   "Veg_LandCoverTrend_evi_mean"
[53] "Veg_LandCoverTrend_evi_min"   "Veg_Persistant_green_Veg"
[55] "VDCN" {Vertical dist to channel network}
```

3. Alternative modelling approaches

i) Stepwise Multiple Linear Regression Model - "carbonSqrtModel"

```
Call:
lm(formula = target$sqrtCarbon1 ~ catchment_area + channel_network_base_level +
    dem_s_1s + landuse06 + mrvbf + rad_u + TM_2004_b3 + Valley_Depth +
    Veg_FPAR_StdDev + Veg_FractCover_Max_BS + Veg_FractCover_Std_NPV +
    Veg_FractCover_Std_PV + Veg_LandCoverTrend_evi_max +
Veg_LandCoverTrend_evi_mean +
    Veg_LandCoverTrend_evi_min, data = dat2)

Residuals:
     Min       1Q   Median       3Q      Max
-0.46301 -0.17023  0.00581  0.15892  0.48782

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 1.135e+00  2.207e-01   5.146 1.11e-06 ***
catchment_area              1.671e-09  8.874e-10   1.883  0.06224 .
channel_network_base_level -7.074e-03  1.726e-03  -4.097 7.81e-05 ***
dem_s_1s                    6.997e-03  1.700e-03   4.116 7.29e-05 ***
landuse06                   1.414e-04  8.522e-05   1.659  0.09975 .
mrvbf                       2.322e-02  1.387e-02   1.674  0.09685 .
rad_u                       1.664e-01  6.403e-02   2.598  0.01060 *
TM_2004_b3                 -3.967e-03  1.589e-03  -2.497  0.01394 *
Valley_Depth                2.995e-03  1.754e-03   1.708  0.09032 .
Veg_FPAR_StdDev             2.240e+00  1.347e+00   1.663  0.09897 .
Veg_FractCover_Max_BS      -4.127e-03  2.066e-03  -1.998  0.04808 *
Veg_FractCover_Std_NPV      1.764e-02  1.296e-02   1.362  0.17599
Veg_FractCover_Std_PV      -3.595e-02  1.560e-02  -2.305  0.02297 *
Veg_LandCoverTrend_evi_max  7.950e+00  2.867e+00   2.773  0.00648 **
Veg_LandCoverTrend_evi_mean -2.402e+01  7.435e+00  -3.231  0.00161 **
Veg_LandCoverTrend_evi_min   2.988e+01  1.136e+01   2.630  0.00970 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.219 on 115 degrees of freedom
Multiple R-squared:  0.4055,   Adjusted R-squared:  0.3279
F-statistic: 5.229 on 15 and 115 DF,  p-value: 7.647e-08
```

ii) Regression Trees (RT)

DSM 2012 notes: these are non-parametric models which progressively split data into increasingly homogeneous subsets. Can accommodate continuous and categorical predictor variables, with virtually no statistical assumptions, and will determine which are most important to retain in the predictive model. However RTs can create nodes which are virtually categorical in nature (compared to Cubist below).

a) "simpleTree" (Regression Tree using package 'rpart')

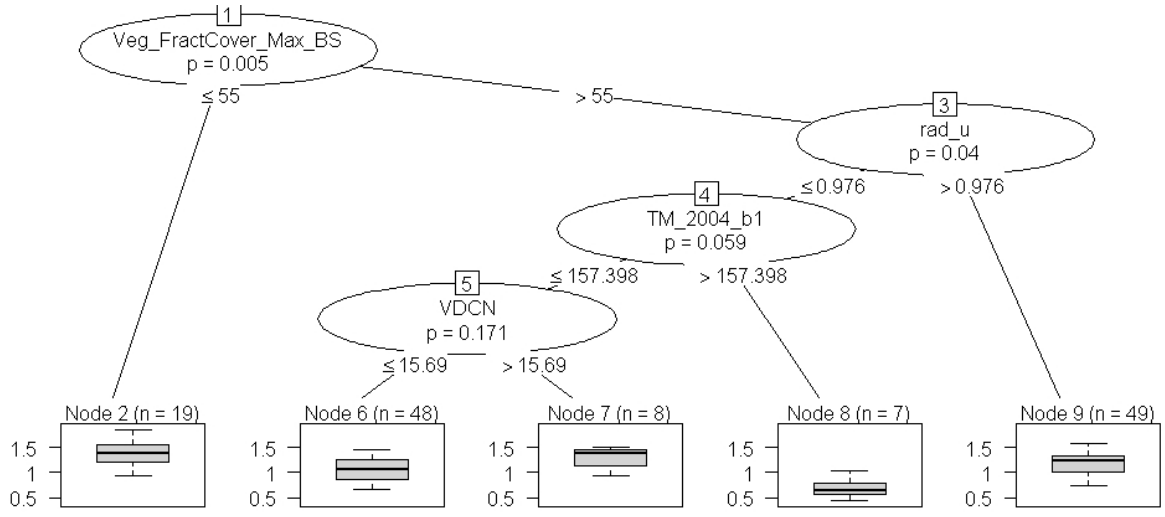
```
Variables actually used in tree construction:
 [1] Convergence_Index           land_type                   Profile_Curv
 [4] rainfall_dewnr              TCI_low                     TM_2004_b1
 [7] Veg_FPAR_Mean               Veg_FractCover_Max_BS
Veg_FractCover_Mean_NPV
[10] Veg_LandCoverTrend_evi_mean
```

```
Root node error: 9.2807/131 = 0.070845

n= 131
```

b) “predCtree” (using package ‘ctree’)

```
> predCtree
         Conditional inference tree with 5 terminal nodes
Response:  target$sqrtCarbon1
Inputs:  hillshading, aspect, catchment_area, channel_network_base_level,
Convergence_Index, dem_s_1s, dem_s_slopepct_fm300, dist_to_coast, land_type,
Landforms, landuse06, LS_Factor, mrvbf, Plan_Curv, rad_k, rad_th, rad_u,
Weathering_Index, Profile_Curv, rainfall_dewnr, rel_slope_pos, slope, TCI_low,
TRI, TM_2004_b1, TM_2004_b2, TM_2004_b3, TWI, Valley_Depth, Veg_FPAR_Mean,
Veg_FPAR_Median, Veg_FPAR_Min, Veg_FPAR_StdDev, Veg_FractCover_Max_BS,
Veg_FractCover_Max_PV, Veg_FractCover_Mean, Veg_FractCover_Mean_NPV,
Veg_FractCover_Mean_PV, Veg_FractCover_Min_NPV, Veg_FractCover_Min_PV,
Veg_FractCover_Std_BS, Veg_FractCover_Std_NPV, Veg_FractCover_Std_PV,
Veg_LandCoverTrend_evi_class, Veg_LandCoverTrend_evi_max,
Veg_LandCoverTrend_evi_mean, Veg_LandCoverTrend_evi_min,
Veg_Persistant_green_Veg, VDCN
Number of observations:  131

1) Veg_FractCover_Max_BS <= 55; criterion = 0.995, statistic = 15.154
  2)*  weights = 19
1) Veg_FractCover_Max_BS > 55
  3) rad_u <= 0.9760278; criterion = 0.96, statistic = 11.152
    4) TM_2004_b1 <= 157.3983; criterion = 0.941, statistic = 10.435
      5) VDCN <= 15.68976; criterion = 0.829, statistic = 8.367
        6)*  weights = 48
      5) VDCN > 15.68976
        7)*  weights = 8
    4) TM_2004_b1 > 157.3983
      8)*  weights = 7
  3) rad_u > 0.9760278
    9)*  weights = 49
```

Plot of ‘Ctree’ prediction



iii) Cubist models

These are another form of Regression Tree, except they have a linear predictive model at each terminal node, instead of a single value. Therefore, Cubist models can produce continuous estimates of the target variable.

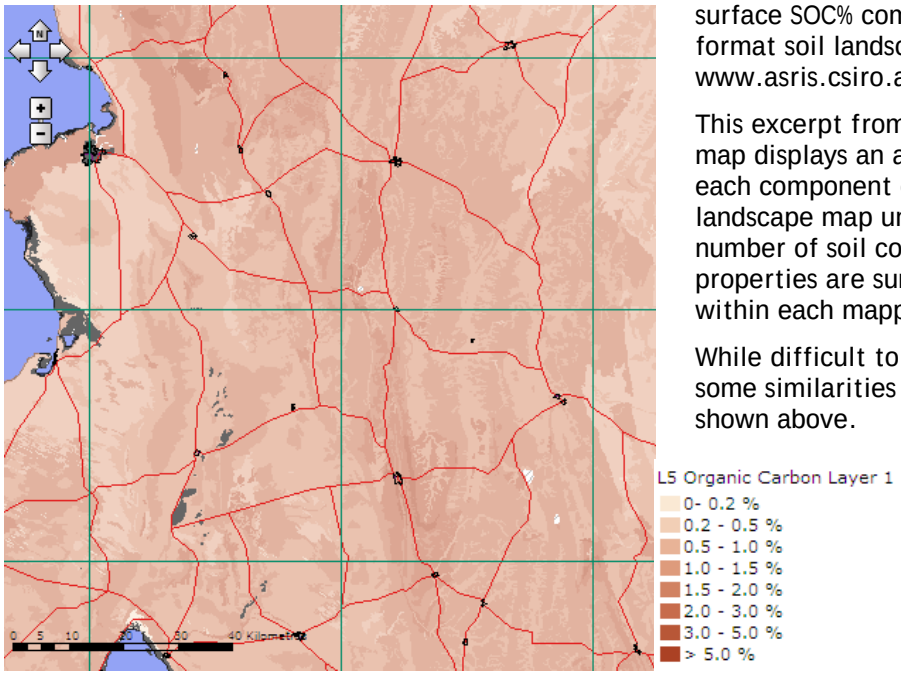
```
> summary(cubistPred)  # output summary

Call:
cubist.default(x = dat2[, 7:55], y = target$sqrtCarbon1, committees = 1,
control
```

```
 = cubistControl(unbiased = F, rules = 100, extrapolation = 5, sample = 0, seed
 = sample.int(4096, size = 1) - 1L, label = "outcome"))

Cubist [Release 2.07 GPL Edition]  Sat May 18 23:48:21 2013
-------------------------------
    Target attribute `outcome'
Read 131 cases (50 attributes) from undefined.data

Model:
  Rule 1: [48 cases, mean 1.0081406, range 0.4472136 to 1.516575, est err
0.1803136]
    if
        rainfall_dewnr <= 425.4863
    then
        outcome = 1.9072184 - 0.0082 TM_2004_b1 + 0.074 mrvbf
                  + 0.057 dem_s_slopepct_fm300 - 0.0012 Veg_FractCover_Mean

  Rule 2: [83 cases, mean 1.2338812, range 0.7416198 to 1.843909, est err
0.2036883]
    if
        rainfall_dewnr > 425.4863
    then
        outcome = 1.3919613 - 0.0062 Veg_FractCover_Mean

Evaluation on training data (131 cases):
    Average  |error|              0.2516341
    Relative |error|                   1.15
    Correlation coefficient            0.29

        Attribute usage:
          Conds  Model
          100%           rainfall_dewnr
                 100%    Veg_FractCover_Mean
                  37%    dem_s_slopepct_fm300
                  37%    mrvbf
                  37%    TM_2004_b1
```

Cross validation was also explored using 30% random holdback of site data.

Other options for Cubist models were also trialled, e.g. ‘committees’{~ boosting} and ‘neighbors’{adjusts predictions using instances nearby the validation data), however these generally resulted in a better fit to the training data (lower MSE, RMSE), at the expense of poorer validation performance (higher MSE,RMSE).

What does existing SOC% mapping look like?



Currently best available spatial mapping for surface SOC% comes from ASRIS or national-format soil landscape mapping (viewable at www.asris.csiro.au)

This excerpt from the ASRIS-format surface SOC% map displays an area-weighted average across each component contained within the soil landscape map unit polygons. In other words, a number of soil components with different properties are summarised for display purposes within each mapping polygon.

While difficult to test statistically, there are some similarities with the DSM training examples shown above.

Appendix B. Examples of R code using features of the ‘raster’ package

```
# read in a csv text file listing ‘path/filename’ and ‘names’ for covariates
cov_list<-
read.table("F:/TAS_trip_working_files/aaa_tile_e138s34/Grids_cleaned_v2_lam/Cov_list_v5_
lamberts.txt",sep=",",header=FALSE)

s<-stack() # create empty raster stack

# create covariate raster stack, and set projection for each raster layer
for(i in 1:no_cov){
  r<-raster(paste(cov_list[i,1]))
  slot(r,"crs")<- CRS("+proj=lcc +a=6378137.000000 +b=6356752.314140 +x_0=1000000.0
+y_0=2000000.0 +lon_0=135.0 +lat_1=-28.0 +lat_2=-36.0 +lat_0=-32.0 +no_defs") # Lamberts
CC projected CRS
  s<-addLayer(s,r)
}

# set names of each layer in the covariate stack
for(i in 1:no_cov){
  names(s)[i]<-paste(cov_list[i,2])
}
names(s)

# import soil site data as SpatialPointsDataFrame, and specify projection
sites<-
readOGR(dsn="C:/WORKSPACE/MOD/R_Projects/TAS_training_Apr2013/soil_site_data",layer="sit
es_loc_e138s34_lam")
sites@proj4string<-CRS("+proj=lcc +a=6378137.000000 +b=6356752.314140 +x_0=1000000.0
+y_0=2000000.0 +lon_0=135.0 +lat_1=-28.0 +lat_2=-36.0 +lat_0=-32.0 +no_defs") # Lamberts
CC projected CRS

# extract covariate values from RasterStack {s}
cov<-as.data.frame(extract(s,sites,method="simple"))
# method- 'simple', as includes class/integer data, otherwise could use 'bilinear'

# Create cubist model, from point/site observations + covariate dataframe (dat2)
cubistPred<-cubist(x= dat2[,7:55],y=target$sqrtCarbon1,
                   cubistControl(unbiased = F,rules = 100,extrapolation = 5,
                                 sample = 0,seed = sample.int(4096, size = 1) - 1L,
                                 label = "outcome"),committees = 1)

#Define function that applies model across covariate space:
applyModel <- function(covStack,model,pred_fxn,chunks,raster_filename) {

  out<-raster(covStack, layer=0)
  bs<-blockSize(covStack,chunksize=chunks)
  # default chunksize=1e+07
  print(paste("number of blocks is ",bs$n,sep=""))
  dy.tot<-ymax(out)-ymin(out) # total distance in y direction across tile
  dy.cell<-dy.tot/nrow(out) # 'height' of each cell in y direction
  out<-writeStart(out,filename=raster_filename,overwrite=TRUE)

  for (i in 1:bs$n){

    # this links row numbers of ‘blocks’ to a raster object ‘extent’, used to limit the
area that model is applied over
    block.ext<-c(xmin(out),xmax(out),ymax(out)-((bs$row[i]-1) + bs$nrows[i])*dy.cell,
ymax(out)-(bs$row[i]-1)*dy.cell) # Extent of each block
    r<-predict(covStack,model,filename=paste("temp_v4_",i,sep=""),fun=pred_fxn,
ext=extent(block.ext), na.rm=TRUE,inf.rm=TRUE,overwrite=TRUE)
    print(paste("calculating block ",i," of ",bs$n,sep=""))
    v<-getValuesBlock(r,row=1,nrows=bs$nrows[i])
    out<-writeValues(out,v,start=bs$row[i])
  }
  out<-writeStop(out)
  #return(out) # use this to return result to raster object in active memory
}

# Apply cubist model, write to file {alternatively can return a raster file into active
memory}
applyModel(covStack=s,model=cubistPred,pred_fxn=predict,
           chunks=0.5e+05,raster_filename="cubist_SqrtSOC_v4") # a low chunksize was
chosen in this example due to memory problems!

# read in raster that was written to file {in native format of ‘raster’ package}
Tile_cubist_SqrtSOC <- raster("cubist_SqrtSOC_v4")
```

```
# Back-transform to SOC%. Low memory specs on my computer meant I had to do this in
blocks again!
out2<-raster(s, layer=0)
(bs.cubist<-blockSize(Tile_cubist_SqrtSOC,chunksize=1e+07))
# default chunksize=1e+07
print(paste("number of blocks is ",bs.cubist$n,sep=""))
out2<-writeStart(out2,filename="raster_cubist_SOC",overwrite=TRUE)

for (i in 1:bs.cubist$n){
  v<-getValuesBlock(Tile_cubist_SqrtSOC,row=bs.cubist$row[i],nrows=bs.cubist$nrows[i])
  v<-v^2
  print(paste("calculating block ",i,sep=""))
  out2<-writeValues(out2,v,start=bs.cubist$row[i])
}
out2<-writeStop(out2)
writeRaster(out2,filename="cubist_SOC_fulltile_v5",format="SAGA",overwrite=TRUE)
```