Master thesis of Elvis Burchia https://orcid.org/0000-0001-8925-2009
Software and documentation are available online: https://github.com/elvisburchia/DigitalSoilMappingSoftware

APPENDIX: WORKFLOW EXPLANATION

1. Introduction 1
2. Method 1
3. Instruction manual 5
a. Input data 5
b. Program changes in R needed 7
c. Tips 11
4. Output explanation 13
5. References 22
6. Figures and tables 23

1 INTRODUCTION

Digital soil mapping (DSM) is extremely important, because knowledge of soil condition is a prerequisite for appropriate soil protection and sustainable cultivation. For this reason a program in R, based on DSM methods was developed. The program does, starting from chemical soil samples, a large-scale prediction for areas around the sampled locations (**Figure 1**). It can predict any continuous chemical soil properties (e.g. SOM, pH, etc.) with the use of several predictors (e.g. aspect, slope, etc.). High resolution soil property maps, which are the final output, can fill knowledge gaps of detailed spatially-distributed information of chemical soil parameters in any areas.

2 METHOD

The program starts by checking if the source chemical soil data is normally distributed, because geostatistical methods work best when this condition is given and when the mean / variance of data do not vary significantly. After computing data transformations (logarithm, square, root, inverse) of each dependent variable, they are checked for the listed requirements, because significant deviations from normality can affect the kriging estimators (Lark, 2000). Only transformations with a Shapiro statistic greater than 90% are considered for the analysis (**Table 1**).

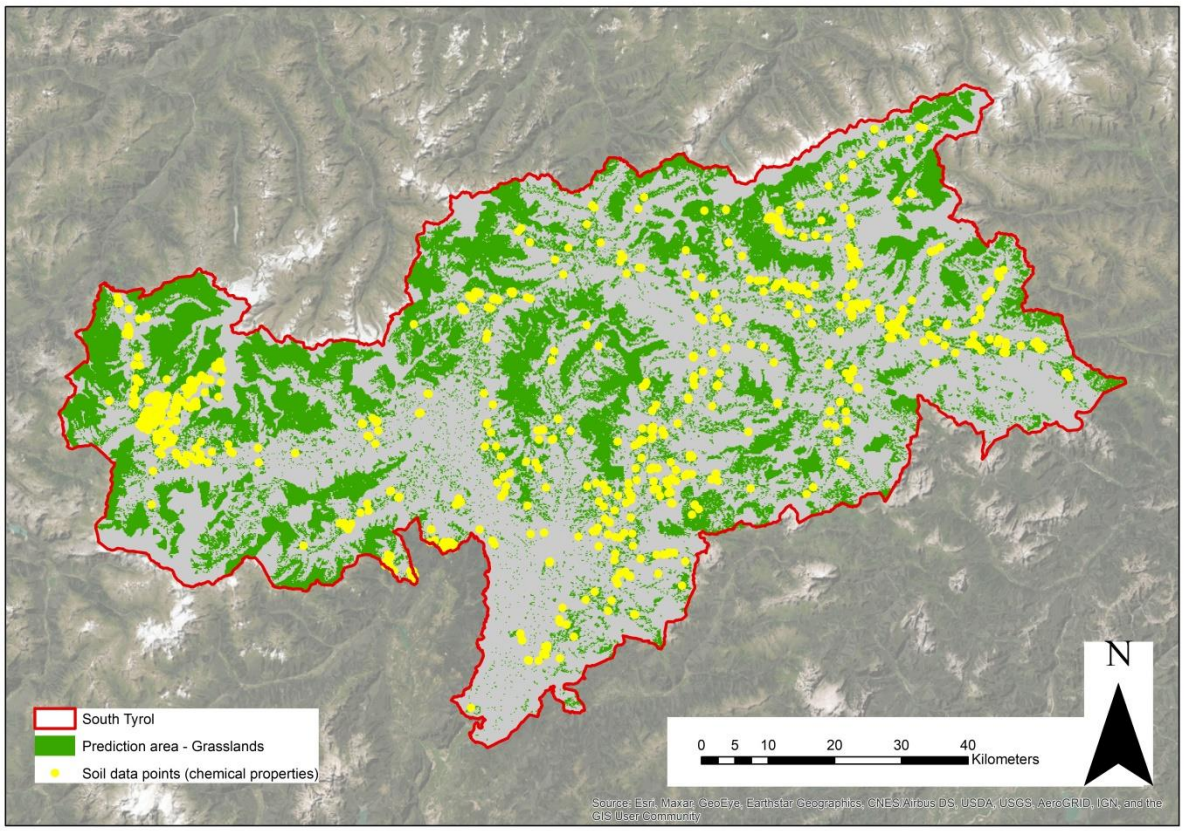


Figure 1: Yellow points with known coordinates and chemical soil properties (e.g. SOM, pH, P, K, C, N) used for prediction in the green area of interest.

Table 1: Check normality and stationary of different source data transformations.

	transformation	skewness	kurtosis	shapiro_p.value	shapiro_statistic	Normal_distributed_data
1	raw	2.7261235	14.9771617	5.400101e-28	0.8107730	NO
2	log	0.2284843	0.5531135	1.562020e-02	0.9947544	YES
3	sqrt	1.2059857	3.5348276	2.629747e-16	0.9399253	YES
4	quad	8.6059409	108.8193115	1.611069e-42	0.4256776	NO
5	inverse	1.4615519	4.0646882	6.796240e-20	0.9113354	YES

The program consists in two parts.

The first part is a regression analysis. Random Forest (RF) is the regression method used (Liaw and Wiener, 2002). It is a powerful ensemble-learning method proposed by Breiman (Breiman, 2001), in which more decision trees are run in parallel and the mean prediction for the continuous dependent variables will be outputted (Chen *et al.*, 2017). The regression analysis outputs two plots for each variable and data transformation (**Figure 2**). One plot is the goodness of fit, which describes how similar the observed and the predicted values are. The second plot is the residual variogram that gives an assessment of the variance of each variable. This is necessary to compute a spatial interpolation (Genova, 2017).

The crosses in **Figure 2** correspond to the "experimental variogram" on the residuals, which explains the semivariance (gamma) computing the distance between each couple of points' values of the dataset and regrouping the distances in bins. The green curve corresponds to the "theoretical variogram" that describes how semivariance changes. It is fitted on the experimental variogram in order to find the model parameters for the variable to be interpolated.

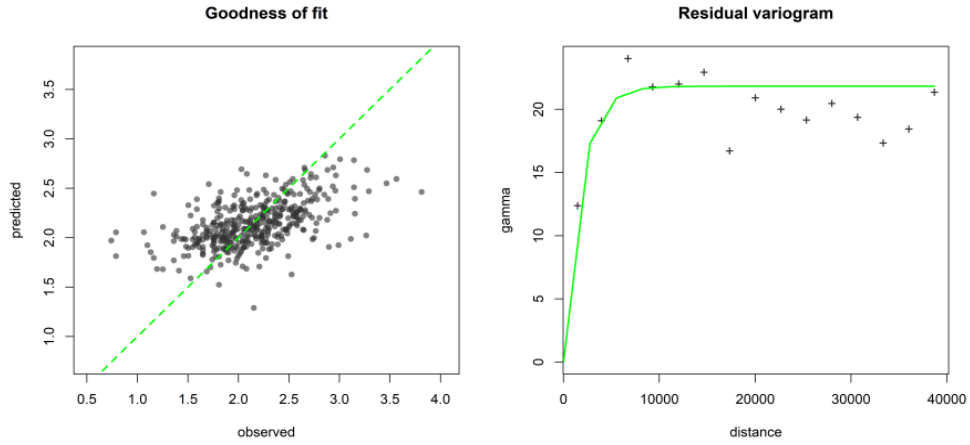


Figure 2: Output plots of the Random Forest regression analysis.
The goodness of fit describes the accuracy of the regression and the residual variogram gives an assessment of the variance of each variable. The crosses are the "experimental variogram" and the green curve is the "theoretical variogram".

The second part is the prediction part. The Ordinary Kriging (OK) (Krige, 1951) interpolation method is used to create the final map rasters of the chemical soil properties (Li and Heap, 2014). It is a widely adopted method that provides a solution to the problem of estimation based on a continuous model of stochastic spatial variation. The variation of the chemical properties is determined through the variogram model. The theoretical variogram model from the first part is connected to the kriging estimator to interpolate the value for each cell of a raster mask (Genova, 2017). So this spatial model gives the possibility to estimate a value at a point of an unsampled region for which a variogram is known. The estimation is done by the following linear combination of values at known locations

$$unsampled\ region\ prediction = \sum_{i=1}^{n} weight_{\ i} * value_{\ i}$$

where *i* is the running variable and *n* is the number of known chemical soil data points (**Figure 3**). Every location has a weight, an importance, which is given by the theoretical variogram. The known locations that are further away will become less weight than the closer ones.

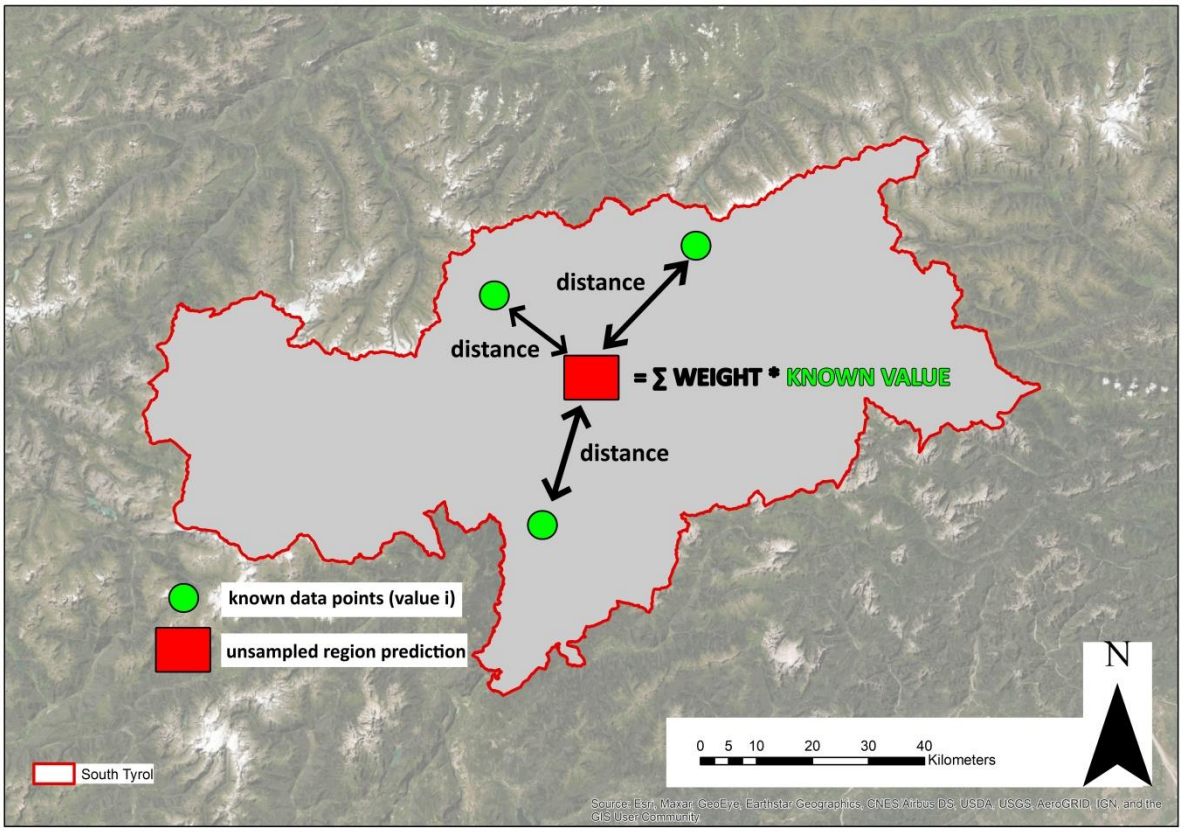


Figure 3: Simple graphic illustration of how Ordinary Kriging (OK) interpolation works.

In order to allow the model validation the source data (chemical soil properties) is split into two parts, a training dataset with 80% of the source data and a validation dataset with the remaining randomly chosen 20% of the source data. The training dataset is used to fit the models (RF and OK). The validation dataset is not used for the analysis, but it is overlayed with the predicted values from the OK interpolation. The R-Squared (R^2) and the Root Mean Square Error (RMSE) are calculated to help choosing the best model (**Table 2**).

Table 2: Possible overview table with the most important output values of each program run.

No.Run	Transformation	Cutoff	Predictors	Mean_Of_Squared_Residuals	%VarExplained	RMSE	NRMSE	NRMSEMAXMIN	R_SQUARED
122	log	30000	1/3/4/5/6/7/9/10/22/23/27/28/29	0.1467528	31.94	2.751960	71.9	13.6	0.4927847
107	log	32000	1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23	0.1398088	35.16	2.775192	72.5	13.7	0.4856251
126	log	29000	1/3/4/5/6/7/9/10/22/23	0.1614357	25.13	2.784648	72.8	13.8	0.4682315
109	log	30000	1/2/3/4/5/6/7/8/9/10/22/23/27/28/29	0.1483269	31.21	2.796911	73.1	13.8	0.4786584
116	log	29000	1/2/3/4/5/10/22/23	0.1700704	21.13	2.815992	73.6	13.9	0.4540478

The structure of the main program is as described here (**cf. p. 12**).

3 INSTRUCTION MANUAL

Before starting the program the input data needs to be checked. Based on the source data, a few changes in the R-Script are needed. There are also some tips to consider. They might be useful.

a Input data

The input data (1-5) has to be stored in the following folders (**Figure 4**):

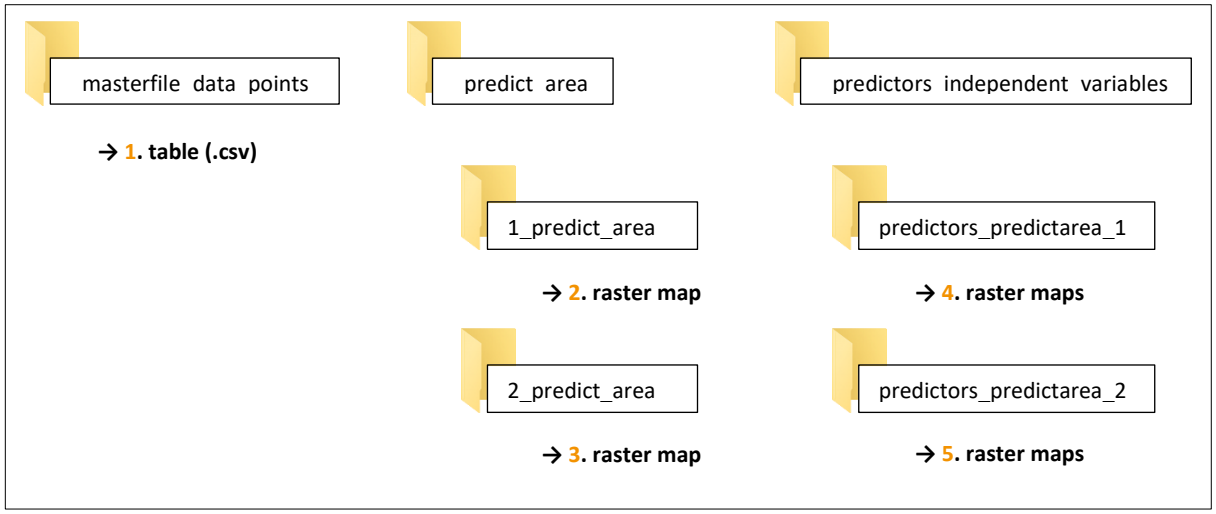


Figure 4: Overview of the input data folders. An explanation of the single files (1-5) follows.

1. Masterfile with the soil data points (**dependent variables**)

   Microsoft Excel or SPSS can be used to organize and prepare the source data, so to become at the end a table (.csv) containing coordinates with information about chemical soil properties. These coordinates correspond to the yellow points in **Figure 1**. The table has to correspond to the following layout (**Table 3**). The chemical values must start in the ninth column and the last two columns must contain the geographic coordinates.

Table 3: Layout of the table containing the source data of the chemical values (dependent variables).

FID	Source	Code	GIS_ID	Sample_Type	Habitat_Type	Town	Year	SOM	pH	P_mg100g	K_mg100g	C	N	CN	x	y
0	1	61001967	61001967	Soil	Grasslands	LAAS	2006	7.60	7.30	23.0	42.0	-9999.00	-9999.00	-9999.00	628836	5165114
1	1	61001998	61001998	Soil	Grasslands	ANTHOLZ	2006	2.20	6.40	7.0	26.0	-9999.00	-9999.00	-9999.00	737420	5195407
2	1	61005017	61005017	Soil	Grasslands	TOBLACH	2006	7.60	5.90	27.0	53.0	-9999.00	-9999.00	-9999.00	747596	5180101
3	1	61005302	61005302	Soil	Grasslands	ALDEIN	2006	6.30	5.50	17.0	50.0	-9999.00	-9999.00	-9999.00	678866	5135892
4	1	61005306	61005306	Soil	Grasslands	ALDEIN	2006	7.20	5.40	5.0	24.0	-9999.00	-9999.00	-9999.00	679167	5135742
5	1	61005307	61005307	Soil	Grasslands	ALDEIN	2006	5.00	5.20	5.0	12.0	-9999.00	-9999.00	-9999.00	679172	5136002

2. First area of interest

The area of interest is where the prediction of the soil chemical properties has to be done. It is possible to enter two prediction/spatial interpolation areas simultaneously, because the program can treat them separately. This means that it provides an output for each dependent variable and for each area of interest. It has to be a raster file with a specific resolution that is determined as follows. The raster resolution is related to the density of samples per area of interest. The denser the measurement points, the larger the scale of mapping, the smaller the resolution. The formula used to determine it, is the one described by (Hengl, 2006)

$$grid\ resolution = 0.0791 * \sqrt{\frac{A}{N}}$$

where *A* is the surface of the area of interest in m^2 and *N* is the total number of the points with chemical measurements. Programs for preparing these raster maps and the following ones, always with the same resolution, can be ArcGIS (closed source) or QGIS (open source).

3. Second area of interest

The second raster can be saved here. If there is no second area of interest, the folder "2_predict_area" must be deleted.

4. Predictors (**independent variables**) of the first area of interest

The raster maps of the predictors have to be saved here. It is important to distinguish between categorical and numerical data. For numerical data (e.g. elevation, slope, etc.), there must be one raster map per predictor. In the case of categorical data (e.g. geology, land use, etc.), the information with similar characteristics are grouped. For this type of data a binary raster for each category of each predictor is required. This means that only 0 and 1 (not two other values) can be included in the single rasters, it applies or it doesn't. An example is the following. If the predictor "geology" is taken into account, there can be more categories, so for example acidic soil types and basic soil types. For each category of the categorical predictors a binary raster is needed. In this case for example one for acidic soil types and another for basic soil types.

5. Predictors (**independent variables**) of the second area of interest

It can be that different predictors are taken into account for the second area of interest. These predictors can be saved here, following the same explanations as above. If there is no second area of interest the folder "predictors_predictarea_2" must be deleted.

The naming of the predictors' rasters (**cf.** 4 and 5) must be as follows:

- ⇨ **P_1_***Name of the first predictor*_**resolution**
- ⇨ **P_2_** *Name of the second predictor* **_resolution**
- ⇨ **…**

The parts in bold must be exactly the same as here. Instead of "resolution", the calculated grid resolution (**cf. p. 6**) must be entered here, for example "150". As you can see the predictors have to be sequentially numbered. These numbers are very important, because in the result tables they will appear instead of the predictors' names. The program can consider more predictors combinations that are determined with the predictors' numbers (**cf. p. 9**).

b Program changes in R needed

In the line 62 the calculated grid resolution (**cf. p. 6**) must be entered. Since the program considers more combinations of predictors, the code in line 64 offers the possibility to determine the mode of combination, "personalized" or "automatic" (**Figure 5**). If the number of the predictors is less than or equal to five, the automatic mode can be choosen. Otherwise the program takes too much time to process the data (**cf. p. 11**). A detailed explanation of the combinations' settings follows (**cf. p. 9**).

```
62  res <- 150
63  resolution <- paste0("",res,"")
64  predictors_combination <- "personalized"  # or "automatic"
```

Figure 5: Program changes in lines 62 and 64 ("res" is an abbreviation of resolution).

The next coming changes are for the labels of the dependent variables in graphs and plots. In line 298 ff. there are several if-else statements whose number depends on the number of the dependent variables. The dependent variables are those included in the masterfile (**cf.** 1). The general syntax of an if statement is the following (**Figure 6**):

```
if (text_expression) {
      statement
} else if (text_expression) {
      statement
} else {
      statement
}
```

Figure 6: General syntax of an if statement.

The different "text_expressions" and "statements" that have to be modified are

Line 298 ff.

if – text_expression 1: variables[j] == "masterfile's column name of the first variable"

statement 1: einheit = "unit of the first variable"

else if - text_expression 2: variables[j] == "masterfile's column name of the second variable"

statement 2: einheit = "unit of the second variable"

…

else if - text_expression n: variables[j] == "masterfile's column name of the n-th variable"

statement n: einheit = "unit of the n-th variable"

else – text_expression n+1: /

where *n* is the number of dependent variables. The orange parts have to be substituted by the right name of each variable that corresponds to the masterfile's column names (**Table 3**) and the right unit of the variables. At the end the number of "else if statements" must be *n-1*. The other parts remain unchanged. Here an example with the same dependent variables from **Table 3** (**Figure 7**).

```
297     # variable description for graphics
298     if (variables[j]=="SOM"){
299       varr = variables[j]
300       einheit = "%"
301       varbez = paste0(varr," ","[",einheit,"]")
302       } else if (variables[j]=="pH"){
303         varr = variables[j]
304         einheit = " "
305         varbez = varr
306       } else if (variables[j]=="P_mg100g"){
307         varr = "P"
308         einheit = "mg/100g"
309         varbez = paste0(varr," ","[",einheit,"]")
310       } else if (variables[j]=="K_mg100g"){
311         varr = "K"
312         einheit = "mg/100g"
313         varbez = paste0(varr," ","[",einheit,"]")
314       } else if (variables[j]=="C"){
315         varr = variables[j]
316         einheit = "%"
317         varbez = paste0(varr," ","[",einheit,"]")
318       } else if (variables[j]=="N"){
319         varr = variables[j]
320         einheit = "%"
321         varbez = paste0(varr," ","[",einheit,"]")
322       } else if (variables[j]=="CN") {
323         varr = variables[j]
324         einheit = " "
325         varbez = variables[j]
326       } else {
327         varbez = variables[j]
328       }
```

Figure 7: Example of how to lable the dependent variables in graph and plots (changes in line 298 ff.).

In the lines 1095 ff. and 1181 ff. the denomination of the variables for the correlation coefficients' plots/tables has to be changed. The naming must be as follows:

```
Line 1095 ff. (for the first area of interest)
        names(corr_parameters1)[names(corr_parameters1)
                == " P_1_Name of the first predictor_resolution (cf. p. 9) "] <- "P1_your name"
        names(corr_parameters1)[names(corr_parameters1)
                == " P_2_Name of the second predictor_resolution (cf. p. 9) "] <- "P2_your name"
        …

Line 1181 ff. (for the second area of interest)
        names(corr_parameters2)[names(corr_parameters2)
                == " P_1_Name of the first predictor_resolution (cf. p. 9) "] <- "P1_your name"
        names(corr_parameters2)[names(corr_parameters2)
                == " P_2_Name of the second predictor_resolution (cf. p. 9) "] <- "P2_your name"
        …
```

The orange parts have to be substituted by the right raster files' name of each predictor (**cf. 4** and **5**). Instead of the blue parts, any name for each predictor, which will then appear in the correlation coefficients' plots/tables, can be choosen. If there is no second area of interest, the changes in line 1181 ff. can be ignored.

The pre-last changes, in line 1274 ff. and line 1322 ff., refer to the different predictors' combinations executed by the program. This part should only be considered if the mode in line 64 is set to "personalized" (**Figure 5**). Before defining the combinations it can be useful to check autocorrelation between numeric variables and between the single predictors (**cf. X**). The amendments are

```
Line 1274 ff. (1st area of interest) and Line 1322 ff. (2nd area of interest)
nrofcombinations <- "number of combinations inputted in the following lines"
predictors_combi_1 <- c(1,2,3,…)
predictors_combi_2 <- c(1,2,3,…)
…
predictors_combi_n <- c(1,2,3,…)
```

where *n* must correspond to the number of combinations (`nrofcombinations`) in the line 1274. The predictors are sequentially numbered (**cf. p. 7**). These numbers are inputted instead of the orange parts to decide what predictors must be taken into account for the different combinations. Remember that the more combinations, the more time the program needs for processing (**cf. p. 11**).

The last changes, in lines 1699 ff., 1792 ff., 1885 ff., 2170 ff., 2263 ff., 2356 ff., 2641 ff., 2734 ff., 2827 ff., 3159 ff., 3252 ff., 3345 ff., 3631 ff., 3724 ff., 3817 ff., 4146 ff., 4239 ff., 4332 ff., are about the plots and tables that describe the influence of the individual predictors on the dependent variables, which results from the Random Forest regression. These changes

Line 1699 ff. / 2170 ff. / 2641 ff. / 3159 ff. / 3631 ff. / 4146 ff.

```
if (i==1) {   # [for the first area of interest]
    names(incmse)[names(incmse) ==
      " P_1_Name of the first predictor_resolution (cf. p. 9) "] <- " P1_your name "
    names(incmse)[names(incmse) ==
      " P_2_Name of the second predictor_resolution (cf. p. 9) "] <- " P2_your name "
    …

} else if (i==2) {   # [for the second area of interest]
    names(incmse)[names(incmse) ==
      " P_1_Name of the first predictor_resolution (cf. p. 9) "] <- " P1_your name "
    names(incmse)[names(incmse) ==
      " P_2_Name of the second predictor_resolution (cf. p. 9) "] <- " P2_your name "
    …

} else {}
```

Line 1792 ff. / 2263 ff. / 2734 ff. / 3252 ff. / 3724 ff. / 4239 ff.

```
if (i==1) {   # [for the first area of interest]
    names(IncNodePurity)[names(IncNodePurity) ==
      " P_1_Name of the first predictor_resolution (cf. p. 9) "] <- " P1_your name "
    names(IncNodePurity)[names(IncNodePurity) ==
      " P_2_Name of the second predictor_resolution (cf. p. 9) "] <- " P2_your name "
    …

} else if (i==2) {   # [for the second area of interest]
    names(IncNodePurity)[names(IncNodePurity) ==
      " P_1_Name of the first predictor_resolution (cf. p. 9) "] <- " P1_your name "
    names(IncNodePurity)[names(IncNodePurity) ==
      " P_2_Name of the second predictor_resolution (cf. p. 9) "] <- " P2_your name "
    …

} else {}
```

Line 1885 ff. / 2356 ff. / 2827 ff. / 3345 ff. / 3817 ff. / 4332 ff.

```
if (i==1) {   # [for the first area of interest]
    names(importancesd)[names(importancesd) ==
      " P_1_Name of the first predictor_resolution (cf. p. 9) "] <- " P1_your name "
    names(importancesd)[names(importancesd) ==
      " P_2_Name of the second predictor_resolution (cf. p. 9) "] <- " P2_your name "
    …

} else if (i==2) {   # [for the second area of interest]
    names(importancesd)[names(importancesd) ==
      " P_1_Name of the first predictor_resolution (cf. p. 9) "] <- " P1_your name "
    names(importancesd)[names(importancesd) ==
      " P_2_Name of the second predictor_resolution (cf. p. 9) "] <- " P2_your name "
    …

} else {}
```

The orange parts have to be substituted by the right raster files' name of each predictor (**cf. 4** and **5**). Instead of the blue parts, any name for each predictor, which will then appear in the plots and tables, can be choosen. If there is no second area of interest, the else if statement can be ignored.

c Tips

Program execution time

It is important to know about the time that the program takes to process the data. The execution time depends from:

- the number of dependent variables to predict (**cf. 1**),
- the number of soil data points of each variable (**cf. 1**),
- the number of predictors' combinations that have been processed (**cf. p. 9**),
- the number of the areas of interest s (one or two) and
- the area size of the areas of interest (**cf. 2** and **3**).

Since the program checks several combinations of the predictors, the mode of defining them must be selected at the beginning of the program, in line 64 (**Figure 5**). The program executes each predictors' combination three times for every data transformation that provides a Shapiro statistic greater than 90% (**Table 1**). The structure of the program, based on nested for loops, is

```
1) Exploratory data analysis
2) Main code – nested for loops
for (j in 1:lenvariables) {}            # (dependent variables)
   for (i in 1:nrpredictareas) {}         # (areas of interest)
      for (k in 1:lange) {}                 # (data transformations)
         for (s in 1:diffcutoff) {}           # (runs with different cutoffs)
            for (l in 1:nrofcombinations) {}    # (predictors' combinations)
```

where "**lenvariables**" is the number of dependent variables, "**nrpredictareas**" is the number of areas of interest, "**lange**" is the number of data transformations with a Shapiro statistic greater than 90%, "**diffcutoff**" is the number of runs done with different cutoff settings for the Random Forest regression and "**nrofcombinations**" is the number of the predictors' combinations. The variable "diffcutoff" is three, because the Random Forest regression is done one time with the minimum cutoff, a second time with the average cutoff and a third time with the maximum possible cutoff.

The time that the program takes to process the input data can be estimated with the second R script "*test_execution_time*" that is also contained in the folder "*DSM*". In line 10 the number of the predictors' combinations (*no.combi*) that have to be taken into account has to be specified. It must be executed before the main R script "*dsm*". The file "*estimated_execution_time.txt*" which is outputted contains all details. You cannot exactly rely on these times, because there can be deviations. It gives you only an idea of how long the program will take to process the input data.

Geographic coordinate system (GCS)

It is important to have all input data rasters (**cf. 2**, **3**, **4** and **5**) with the same reference framework that defines the locations of features.

File names and file paths' options

A helpful advice is to keep file names short, but meaningful. Abbreviations often help create such names that should be easy to read and understand. Do not use umlauts and follow the guidelines naming the predictors' rasters (**cf. p. 7**). The file paths cannot be too long, therefore it is a good idea to store the folder "dsm" on the desktop. It is also important to check the amount of free space on the hard disk. The necessary storage space depends on the source data and cannot be determined in advance.

Define area of interest 1 and area of interest 2

From the beginning it should be clear which the first area of interest and which the second area of interest is. Sometimes the area of interests file names appear in the output plots and tables and sometimes it simply says "area of interest 1" or "area of interest 2". It is also very important to know what predictors are exactly taken into account for the individual areas and its sequential numbers. For this reason it might be helpful to make a list of the predictors of each area of interest containing the name and the sequential number of the single predictors. The results will be then easier to interpret and hopefully no wrong interpretations will happen.

Plot pane in Rstudio

The plot window in Rstudio must have a certain size (half screen size) so that no errors occur.

Avoid computer overload

When the program is running all available resources should be accessible for it, so that it can performed optimally. In order to allow this, several open applications should be stopped.

Split up the run of the program

The estimated execution time (**cf. p. 11**) of the program can be long. It may be necessary to split the process. In this case the computer can set into sleep mode. As soon as the computer is restarted, the program resumes its activity automatically.

4 OUTPUT EXPLANATION

The program creates two output folders: "**output**" and "**output_overview**".

The folder "**output**" contains all outputs of all runs of the program. The number of the runs of the program (*no. runs*) for each area of interest can be calculated with the formula

$$no.runs = \sum_{j=1}^{lenvariables} x_j * nrofcombinations * 3$$

where an explanation of the single variables is given above (**cf. p. 12**). The variable "x_j" corresponds to the number of data transformations (logarithm, square, root, inverse) of the dependent variable *j* that provides a Shapiro statistic greater than 90% (**Table 1**). This number corresponds to the amount of folders contained in the following file paths: *"…/DSM/output/prediction area*/dependent variable j"*. If there are two areas of interest the result *no. runs* must be doubled.

If there are too few available soil data points it may also be that the Random Forest regression may not work. In this case no data is outputted by the program. As a consequence also the number of runs decreases and the formula above is no longer valid. The correct number of runs can be taken from the output based on the last variable processed by the program. It can be found in the file path *"…/DSM/output/prediction area 1 /last dependent variable processed by the program/data transformation that begins with the latest letter in the alphabet*" if there is one area of interest. If there are two, the file path to consider is the following: *"…/DSM/output/prediction area whose name begins with the latest letter in the alphabet/last dependent variable processed by the program/data transformation that begins with the latest letter in the alphabet*". The order of which variables are processed first is based on **Table 3**. The columns' number determines the order. So the variables that have a smaller number are processed first. In the given paths there are the folders of all program runs' outputs. They are named as follows: "*No.Run_n*", where *n* corresponds to the run number. The highest *n* in the path to consider is the right number of all runs of the program. The file "*estimated_execution_time.txt*" (**cf. p. 12**) also contains a number, but it is not 100% reliable. It is possible that the source data of some dependent variables is not good enough, therefore no prediction and output data can be generated and so there can be less program runs. In this case the source data must be better and a TXT file will appear in the output folders. It explains that the source data is not good enough to do a prediction analysis.

The most important outputs are stored in the folder "**output_overview**". **Figure 8** on the next page describes an overview of its content. The variable **n** corresponds to the number of dependent variables. In this folder there are only outputs of the runs that could provide models with a high Root Mean Square Error (**RMSE**) and a high R-squared (**R^2**). The **runs with the five lowest RMSE values and the five highest R^2 values are picked out**. It is up to you whether you give more weight to RMSE or R^2. Both are important measures of the goodness of the model. The RMSE describes the error of the model, so it is a direct measure of when a prediction is wrong. In addition it has the same unit of measure as the studied variable. This information is very important in practical terms, for example for farmers or soil scientists who want to use the prediction models outputted by the program. The R^2 tells how much of the natural variability of the studied variables is captured by the model. Therefore it is a more general measure and it is also often affected by outliers. However, there is no right or wrong between RMSE and R^2. For the purpose of soil maps perhaps it is better to prioritise measuring how much predictions are wrong than measuring how much variability is accounted. The most important output of the program to take into account is a multi-page PDF file (**cf. IX**).

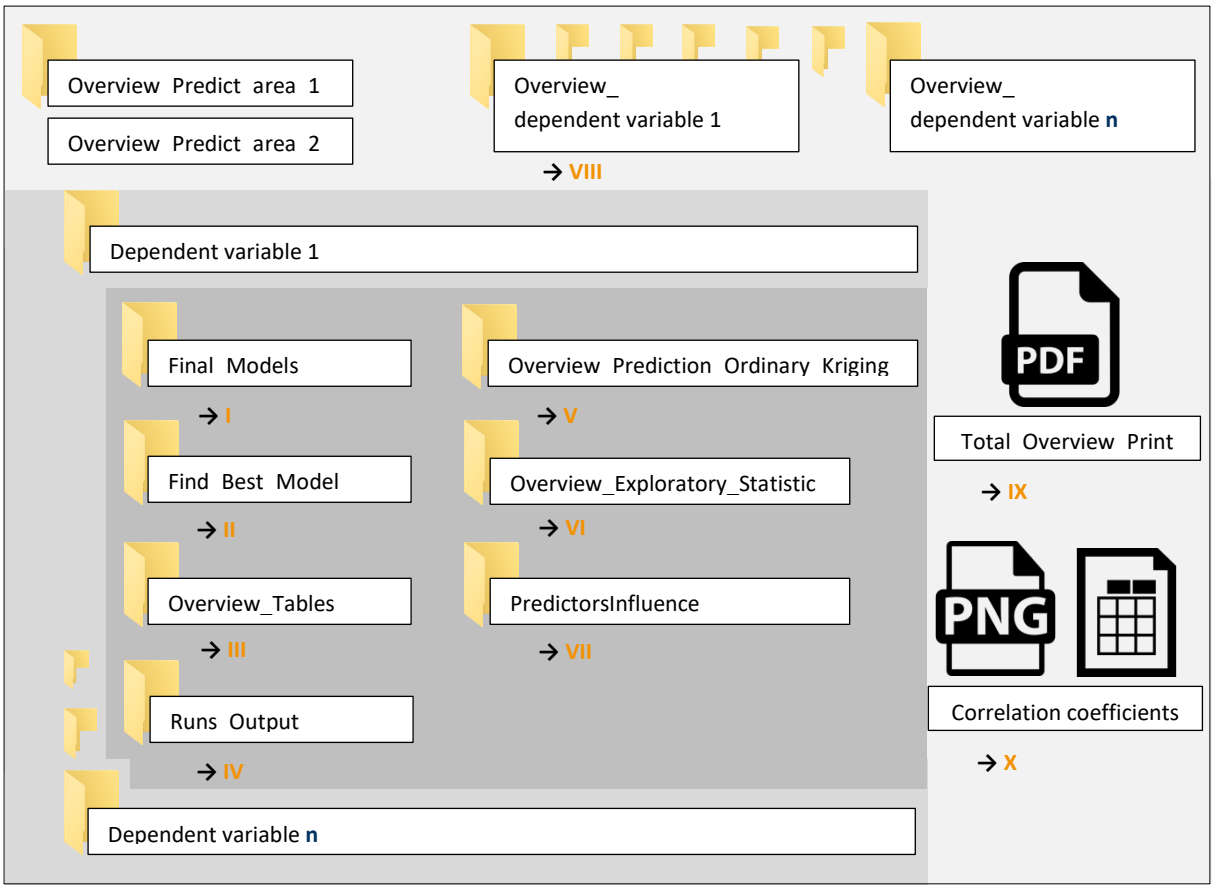


Figure 8: Overview of the content of the folder "output_overview". The explanation of the outputs I-X follows.

The output folders **for each variable and area of interest** are the following (I-VII):

I. Final models

The final raster maps of the predicted dependent variables in each area of interest are stored here. There is a PNG file (.png) of the map, an R Workspace file (.RData) that contains the most important values outputted by the program and a GeoTIFF file (.tif) of the map. The R Workspace file can be used to load important values (RMSE, R^2, etc.) back by R. The GeoTIFF file contains georeferencing information which is necessary to establish an exact spatial reference for the file. It can be loaded in geospatial processing programs like ArcGIS (closed source) or QGIS (open source) to finalise the layout. The resolution of the final maps corresponds to that calculated as described here (**cf. p. 6**).

II. How to find the best model?

As already explained the program's runs with the five lowest RMSE values and the five highest R^2 values are picked out (**cf. p. 14**). It is up to you whether to give more weight to RMSE or R^2. The number of the run which provides the model with the best prediction values can be read from the two PNG files contained in the folder "*Find_Best_Model*". The overview tables (**cf. III**) and the goodness of fit of the Ordinary Kriging model from **V** are included. The two PNG files can be also found in the outputs **VIII** and **IX**.

III. Overview tables

There are more overview tables. The files "*8_Output_Overview_Total_predictarea*.xlsx (csv)*" contain the most important values outputted by each run of the program (RMSE, R^2, etc.). The same table is contained in the files "*8_Output_Overview_Total_predictarea*_HighestR2.xlsx (csv)*" and "*8_Output_Overview_Total_predictarea*_HighestR2.xlsx (csv)*", but the values are ordered in ascending order of RMSE in the first case and in descending order of R^2 in the second case. Then the first five lines of each table are picked out and written in two new tables: "*8_Output_Overview_5HighestR2_predictarea*.xlsx (csv/png)*" and "*8_Output_Overview_5LowestRMSE_predictarea*.xlsx (csv/png)*". These tables are also contained in **II**.

IV. Runs' outputs

There are all outputs of the five runs which have the lowest RMSE and R^2 values. It is not necessary to look at them, because all other important outputs are based on the files contained in this folder.

V. Prediction part – Ordinary Kriging

The output plots of the prediction part of the five runs of the program which have the lowest RMSE and highest R^2 values are stored here. The most important plot is the one showing the goodness of fit / accuracy of the model (in the upper right corner) (**Figure 9**). This means how similar the observed soil chemical values and the predicted values are. The straight line describes the optimal situation, where the observed values are equal to the predicted values. The further away the points are from the line, the greater the uncertainty. These plots are contained in the outputs **VIII** and **IX**, because they can be useful to choose the best prediction model, together with the RMSE and R^2 values.

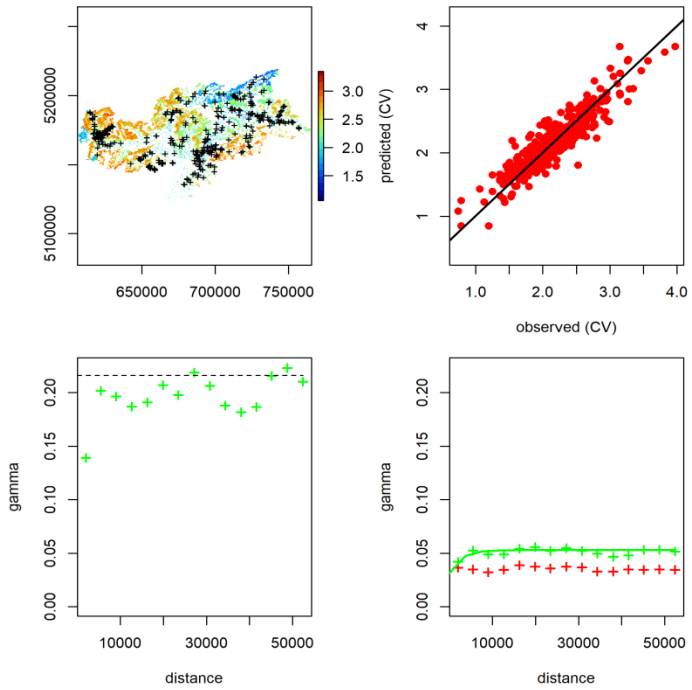


Figure 9: Example of the output plots of the prediction part (Ordinary Kriging).

VI. Exploratory / Descriptive data analysis

The files stored here give information about the descriptive/exploratory analyses done by the program (**Figure 10**). The file "*Descriptive_Statistic_*.png*" contains the minimum, maximum, median and quartile values of each transformation (raw, logarithm, square, root, inverse) of the sample data. The last column of the table says if the data is normal distributed or not. This information is given by the file "*Choose_Transformation_*.png*" where the sample data is tested for normality. Only transformations with a Shapiro statistic greater than 90% are considered for the analysis (**Table 1**), because geostatistical methods work best when the data is normally distributed and its mean and variance do not vary significantly. A Normal Q-Q plot ("*Overview_QQPlot.png*"), as visual check, can also be used to check if the assumption is plausible, and if not, how the assumption is violated and what data points contribute to the violation. It is a scatterplot created by plotting two sets of quantiles against one another. If both sets of quantiles came from the normal distribution, the points should form a line that's roughly straight (Ford, 2015). The file "*Descriptive_Statistic_*.png*" is used to plot a boxplot, the sampling density and the histogram of each transformation of the sample data. The plots are contained in the files "*Overview_BoxplotDensity.png*" and "*Overview_Histogram.png*". They can tell about outliers and if the data is symmetrical.

transformation	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Normal_distributed_data
raw	2.10000000	6.20000000	8.1000000	9.3441915	11.1000000	52.9426516	NO
log	0.74193734	1.82454929	2.0918641	2.1289484	2.4069451	3.9692093	YES
sqrt	1.44913767	2.48997992	2.8460499	2.9751084	3.3316662	7.2761701	YES
quad	4.41000000	38.44000000	65.6100000	111.0924191	123.2100000	2802.9243627	NO
inverse	0.01888836	0.09009009	0.1234568	0.1312897	0.1612903	0.4761905	YES

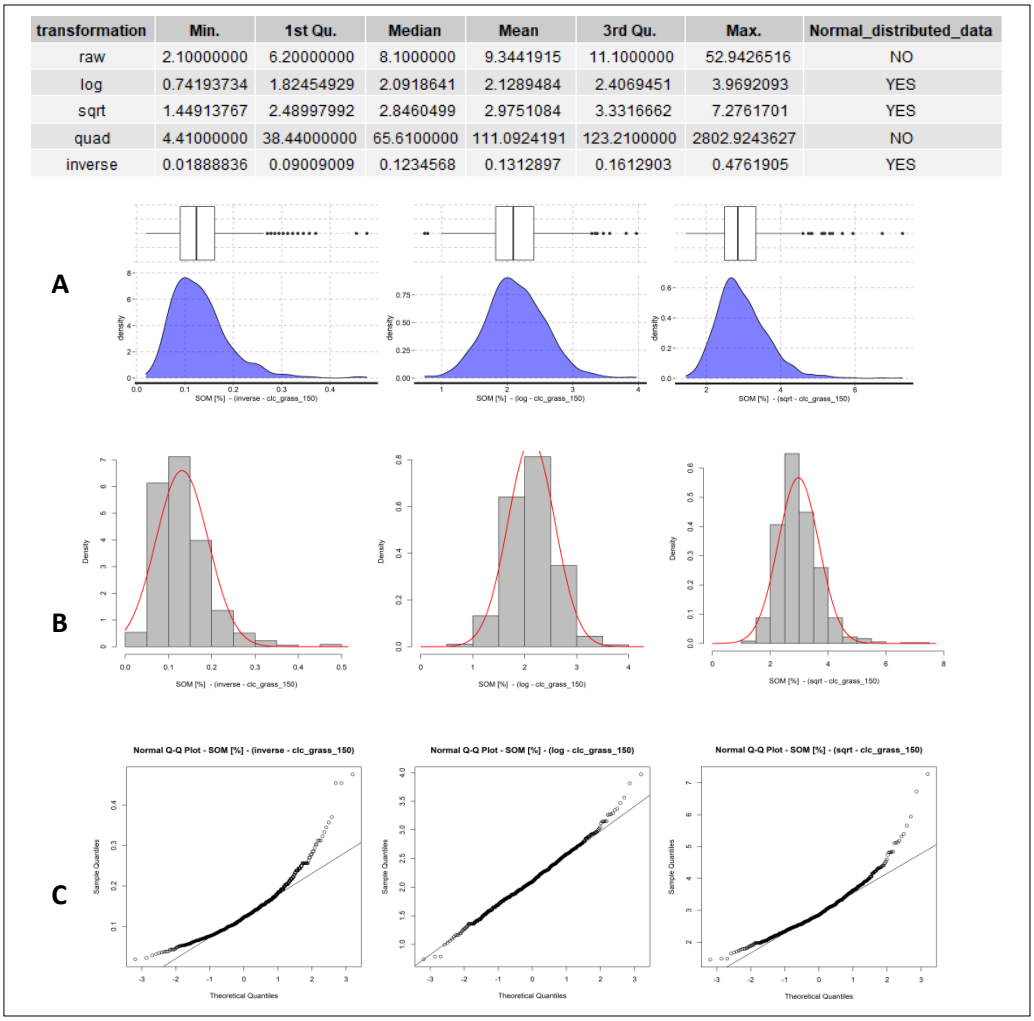
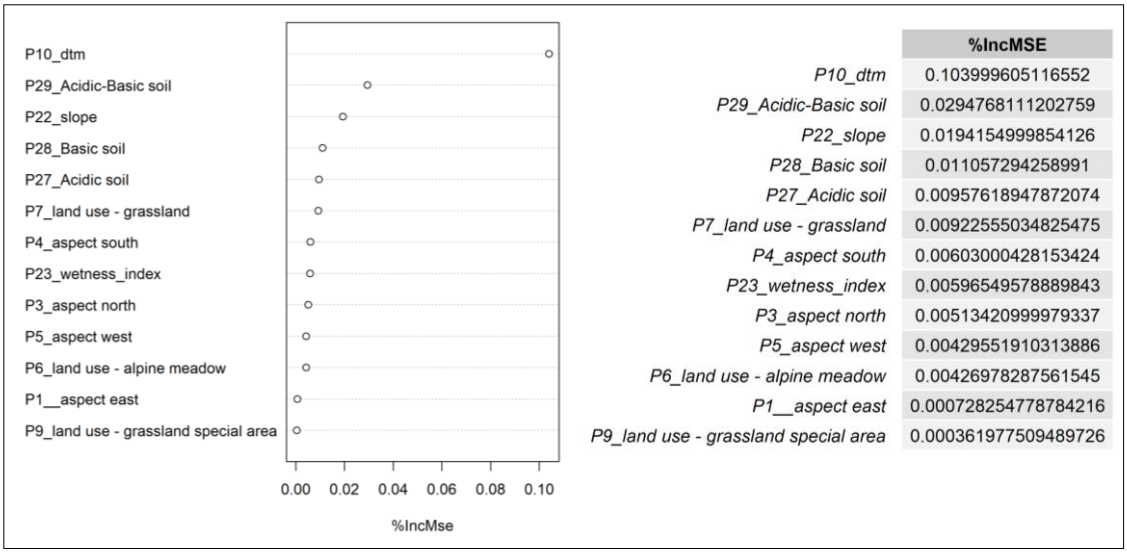


Figure 10: Example of the output of the exploratory / descriptive data analysis
(A: boxplot and sampling density, B: histogram, C: normal Q-Q plot)

VII. Influence of the individual predictors on the dependent variables

The output files describing the influence of the single predictors on the dependent variables are stored here. There are just the outputs of the five runs of the program which have the lowest RMSE and the highest R^2 values. The parameters that describe the influence are the %IncMse ("*4_IncMse_Plot (Table)_*.png*" and "*4_IncMse_ValuesTable_*.xlsx (.csv)*"), and IncNodePurity ("*4_IncNodePurity_Plot (Table)_*.png*" and "*4_ IncNodePurity _ValuesTable_*.xlsx (.csv)*"). They are all outputted by the first part of the program, which computes the regression based on the Random Forest method (**cf. p. 2**). The %IncMSE is the most robust and informative measure (**Figure 11**). It describes the increase of the mean square error (MSE) of the predictions as a result of the permutation of the variables, where the values are randomly shuffled. The higher the %IncMSE number, the more important is that predictor and its influence on the dependent variable taken into account. The same interpretation applies to the IncNodePurity. The importanceSD is also calculated by the program but does not need to be looked at more closely.



	%IncMSE
P10_dtm	0.103999605116552
P29_Acidic-Basic soil	0.0294768111202759
P22_slope	0.0194154999854126
P28_Basic soil	0.011057294258991
P27_Acidic soil	0.00957618947872074
P7_land use - grassland	0.00922555034825475
P4_aspect south	0.00603000428153424
P23_wetness_index	0.00596549578889843
P3_aspect north	0.00513420999979337
P5_aspect west	0.00429551910313886
P6_land use - alpine meadow	0.00426978287561545
P1__aspect east	0.000728254778784216
P9_land use - grassland special area	0.000361977509489726

Figure 11: Example of outputs of the Random Forest regression describing the influence of the single predictors on a specific dependent variable (%IncMSE value).

VIII. Find the best model that predicts a specific dependent variable

Here you find the most important files that help you to choose the best prediction model of each dependent variable. Only the PDF files are important. They contain the outputs **II**, **V** and the descriptive/exploratory statistics' tables of **VI** (**Figure 12**). They give an overview of the best outputs of the program. Only the values of the five runs of the program, which produce the lowest RMSE and the highest R^2 values, are listened.

IX. Find the best models for each dependent variable – Print multi-page document

This document is a merge of all PDF files from **VIII**, of each dependent variable and area of interest. This is **the most important output of the program** that can be printed. It gives you the possibility to find the best prediction model of each dependent variable and each area of interest. You have to decide whether to give more weight to RMSE or R^2 (**cf. p. 14**). After this decision you can **choose the best prediction model** of a specific dependent variable by its number ***x*** of program run. The final raster maps are contained in the folder called "*No_Run_**x***", which is itself contained in the folder "*Final_Models*" (**cf. I**).

X. Correlation coefficients

Before defining the predictors' combination to take into account (**cf. p. 9**) it can be useful to check autocorrelation between numeric variables and between the single predictors. The Pearson's ("*Pearson_Correlation_Predictarea*.png (xlsx, csv)*") and Spearman ("*Spearman_Correlation_Predictarea*.png (xlsx, csv)*") correlation coefficients are

calculated. The values, contained in the table files, range between -1 and 1. A correlation of -1 shows a perfect negative correlation while a correlation of 1 shows a perfect positive correlation. A correlation of 0 shows no linear relationship between the two variables taken into account. The PNG files do not contain the correlation coefficients directly, but they are grafically described with points (**Figure 13**). The color blue stands for a positive correlation and the color red for a negative correlation. The radius and the color strength describe the strength of the correlation. The larger and more intense the background of the points, the stronger the correlation between two specific variables. The Pearson correlation evaluates the linear relationship between two continuous variables. A relationship is linear when a change in one variable is associated with a proportional change in the other variable. The Spearman correlation coefficients measure the monotonic relationship between two variables. Here the variables tend to change together and the Spearman correlation coefficient is based on the ranked values for each variable rather than the raw data (Minitab, 2019).

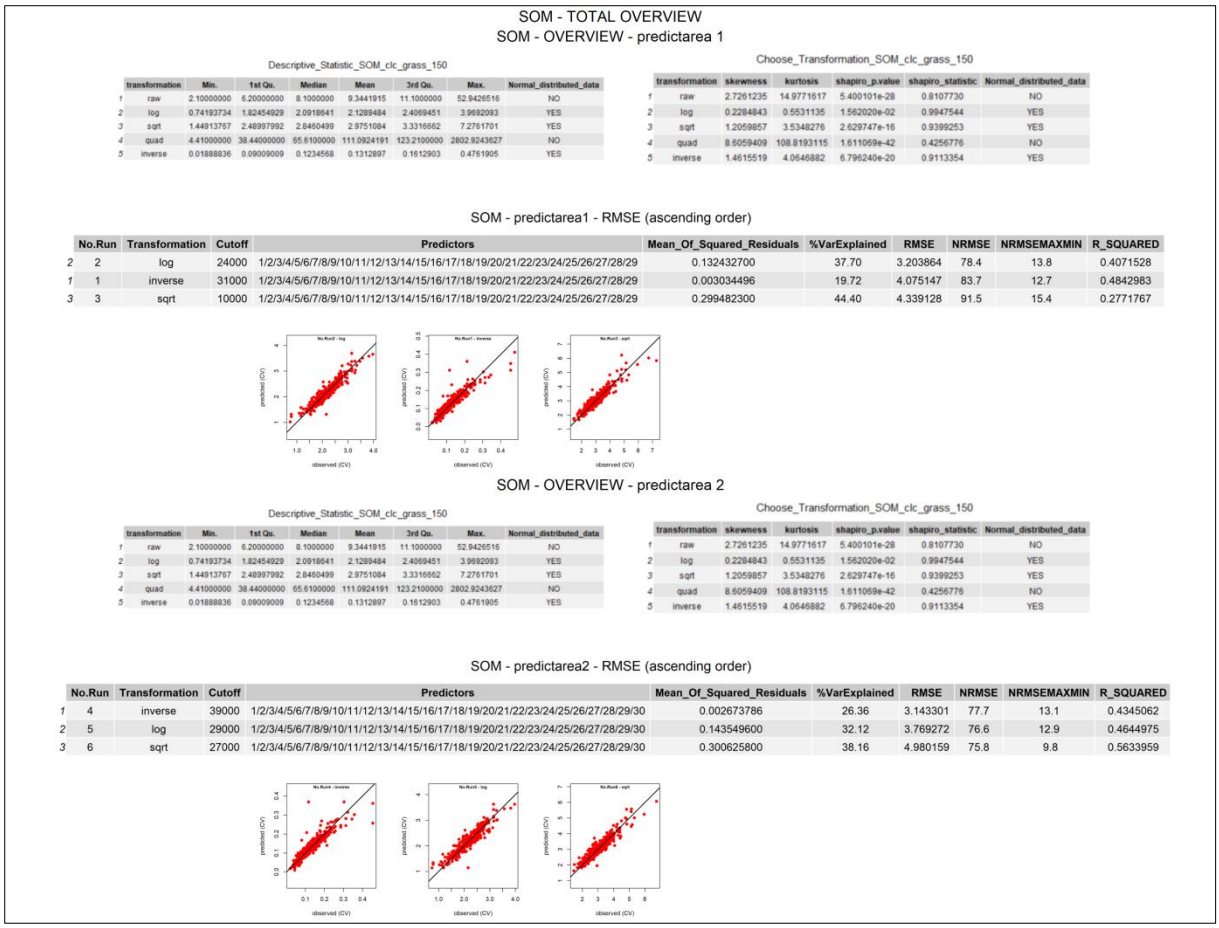


Figure 12: Example of the output that allows to choose the best prediction model of a specific dependent variable and area of interest.

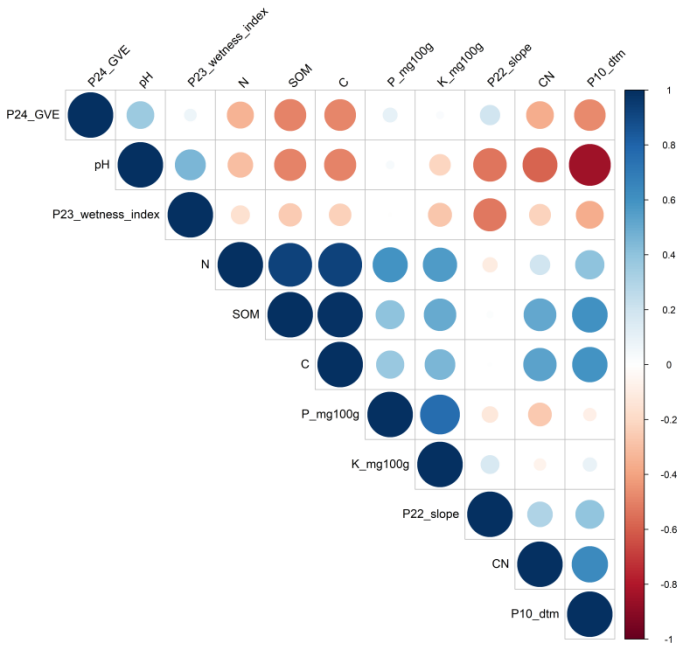


Figure 13: Example of a possible rapresentation of correlation coefficients.

5 REFERENCES

Breiman, L. (2001), "Random Forests", *Machine Learning*, Vol. 45 No. 1, pp. 5–32.

Chen, W., Xie, X., Wang, J., Pradhan, B., Hong, H., Bui, D.T., Duan, Z. and Ma, J. (2017), "A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility", CATENA, Vol. 151 No. 1, pp. 147–160.

Ford, C. (2015), "Understanding Q-Q Plots", *University of Virginia Library*, https://data.library.virginia.edu/understanding-q-q-plots/ (access on 3rd February 2021)

Genova, G. (2017), "Spatial Distribution Assessment of Cu, Zn, PH and Soil Organic Matter in South Tyrolean Permanent Crops", Master Thesis dissertation, *Università degli studi della Tuscia*.

Hengl, T. (2006), "Finding the right pixel size", *Computers & Geosciences*, Vol. 32 No. 9, pp. 1283–1298.

Krige, D.G. (1951), "A statistical approach to some basic mine valuation problems on the Witwatersrand", *Journal of the Southern African Institute of Mining and Metallurgy*, Vol. 52 No. 6, pp. 119-139(21)

Lark, R.M. (2000), "A comparison of some robust estimators of the variogram for use in soil survey", *European Journal of Soil Science*, Vol. 51 No. 1, pp. 137–157.

Li, J. and Heap, A.D. (2014), "Spatial interpolation methods applied in the environmental sciences: A review", *Environmental Modelling & Software*, Vol. 53 No. 9, pp. 173–189.

Liaw, A., Wiener, M., 2002. Classification and regression by randomForest. *R news*, 2(3), 18-22.

Minitab (2019), "A comparison of the Pearson and Spearman correlation methods", *Minitab,* https://bit.ly/39MWmod (access on 4th February 2021)

Krige, D.G. (1951), "A statistical approach to some basic mine valuation problems on the Witwatersrand", *Journal of the Southern African Institute of Mining and Metallurgy*, Vol. 52 No. 6, pp. 119-139(21)

6 FIGURES AND TABLES

Figure 1: Yellow points with known coordinates and chemical soil properties (e.g. SOM, pH, P, K, C, N) used for prediction in the green area of interest..... 2

Figure 2: Output plots of the Random Forest regression analysis. The goodness of fit describes the accuracy of the regression and the residual variogram gives an assessment of the variance of each variable. The crosses are the "experimental variogram" and the green curve is the "theoretical variogram"..... 3

Figure 3: Simple graphic illustration of how Ordinary Kriging (OK) interpolation works. 4

Figure 4: Overview of the input data folders. An explanation of the single files (1-5) follows. 5

Figure 5: Program changes in lines 62 and 64 ("res" is an abbreviation of resolution)..... 7

Figure 6: General syntax of an if statement..... 7

Figure 7: Example of how to lable the dependent variables in graph and plots (changes in line 298 ff.)..... 8

Figure 8: Overview of the content of the folder "output_overview". The explanation of the outputs I-X follows..... 15

Figure 9: Example of the output plots of the prediction part (Ordinary Kriging). 15

Figure 10: Example of the output of the exploratory / descriptive data analysis (A: boxplot and sampling density, B: histogram, C: normal Q-Q plot)..... 15

Figure 11: Example of outputs of the Random Forest regression describing the influence of the single predictors on a specific dependent variable (%IncMSE value)..... 15

Figure 12: Example of the output that allows to choose the best prediction model of a specific dependent variable and area of interest. 15

Figure 13: Example of a possible rapresentation of correlation coefficients. 15

Table 1: Check normality and stationary of different source data transformations. 2

Table 2: Possible overview table with the most important output values of each program run..... 4

Table 3: Layout of the table containing the source data of the chemical values (dependent variables)..... 5

Elvis Burchia

https://orcid.org/0000-0001-8925-2009

Software and documentation are available online:

https://github.com/elvisburchia/DigitalSoilMappingSoftware

DOI 10.5281/zenodo.4683096