Collection of soil data in SOTER format from 14 Danube strategy countries, at scale 1:250 000

EU-JRC Service Contract #936108

Deliverable D.A3

Country terrain unit shapefiles for the EU Danube basin

Maria Ruiperez Gonzalez and Niels H. Batjes

March 2019

ISRIC World Soil Information

Cranfield University

Contents

List of figures

List of tables

1. Introduction

To support the Danube strategy the European Commission Joint Research Centre (JRC) has proposed soil data collection within the Danube basin (Figure 1) based on SOTER methodologies. The SOTER™ (SOil and TERrain) methodology is a comprehensive framework making soil and terrain data available for end users. The methodology was developed by ISRIC in collaboration with FAO, UNEP, JRC and a range of national partners (Van Engelen 2011; van Engelen and Dijkshoorn 2013) and implemented in a range of countries[1].

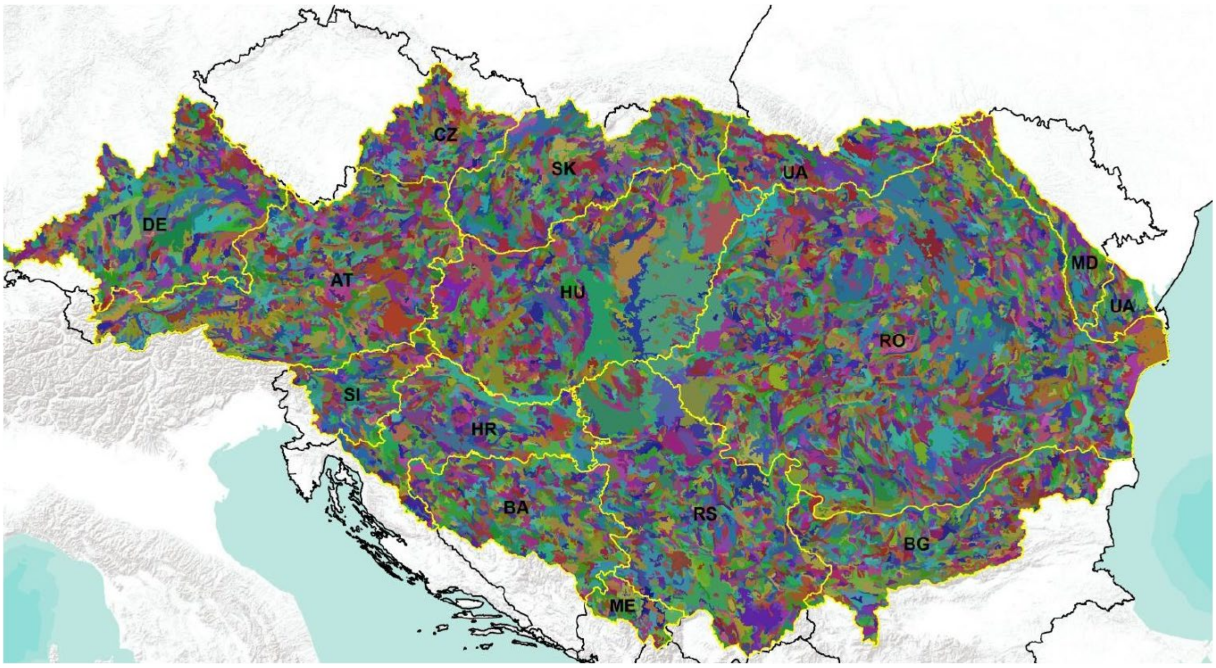


Figure 1. Terrain units for the Danube region basin

In short, the project includes preparing 'empty' SOTER database templates and GIS files of terrain units mapped for the participating countries, filling the 'national' SOTER databases with representative soil profiles, standardisation and harmonisation to the World Reference Base for Soil Resources (2015 update), and ultimately exporting data in SOTER format to GML (Geography Markup Language) INSPIRE-compliant format.

This report describes how the country GIS-shapefiles of terrain units for the Danube basin were made. The country shapefiles can be linked to the 'empty' SOTER attribute database for the respective countries through the unique ISOCSUID identifier (e.g. ISO code for country and sequential number of the given SOTER (terrain) unit). The procedure is illustrated in Figure 2 for a 'simplified' SOTER database (see Batjes and Ribeiro 2019a).

[1] https://www.isric.org/projects/soil-and-terrain-soter-database-programme

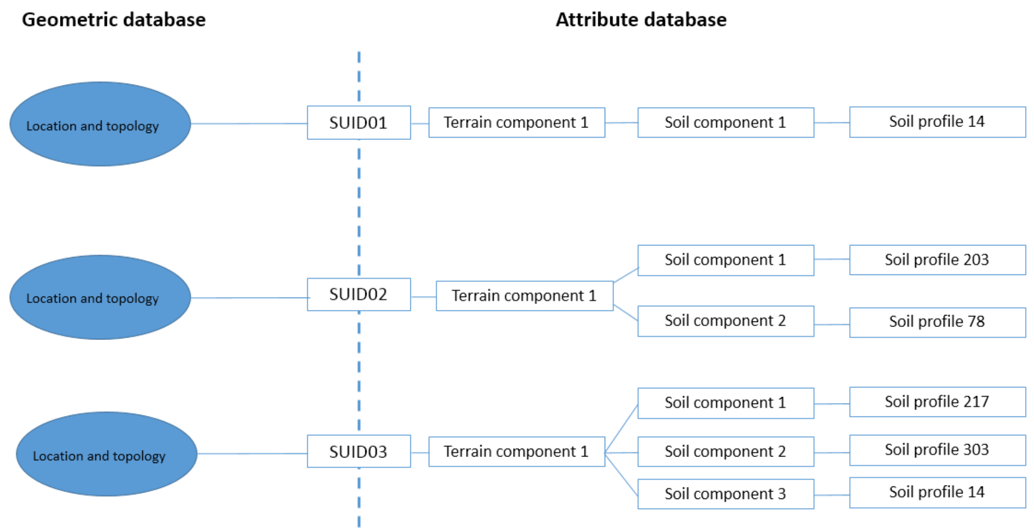


Figure 2. Schematic representation of a 'simplified' SOTER database with its geographic and attribute data component.

2. Data and materials

2.1 GIS files

The Danube basin covers fourteen countries. Terrain unit shapefiles for nine EU countries (Croatia, Germany, Austria, Hungary, Czech Republic, Slovak Republic, Slovenia, Bulgaria and Romania) and five non-EU countries (Serbia, Bosnia-Herzegovina, Montenegro, Ukraine and Moldova), generated subsequent to the EU FP7-funded eSOTER (2012), provided the geographic basis for the current work (Table 1).

Table 1. GIS files used for the SOTER Danube basin project.

GIS shapefiles	Source
SOTER terrain units for whole area	https://esdac.jrc.ec.europa.eu/public_path/shared_folder/SOTER/Danube_Terrain_Units.zip
Outline of Danube basin	https://esdac.jrc.ec.europa.eu/public_path/shared_folder/SOTER/Danube-basin-outline-and-14countries-boundaries.zip, derived from EEA (2016).
Country-boundary for the 14 countries	https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/countries, derived from Eurostat, 2013 version.

Shapefiles for the respective countries were created using a mask of the 'outline of the Danube basin and country-boundary shapefile for the fourteen countries' as provided by JRC (Appendix 1. The country boundaries are derived from Eurostat, version 2013; the Danube basin outline is from EEA (European Environment Agency), 2016 version.

The shapefiles were provided in the following projection ETRS_1989_LAEA, EPSG: 3035, see https://spatialreference.org/ref/epsg/etrs89-etrs-laea/. Being raster-based, several polygons have irregularly shaped boundaries.

2.2 Preparation of country-specific terrain unit shapefiles

The shapefiles described in Section 2.1 provided the basis for compiling 14 sets of terrain units using ArcMap®. In case of terrain units occurring in two neighbouring countries, the corresponding terrain units were allocated the ISOC code for the respective countries, while preserving the original SUID.

On visual inspection, some of the resulting polygons were rather small requiring some 'cleaning up'. As a rule of thumb, the minimum size of a single SOTER unit is 0.25 cm^2 on the map (Van Engelen and Dijkshoorn, 2013, p. 29). At a scale of 1:250,000 this corresponds with ~1.6 km^2 in the field. This is the smallest area that can still be cartographically presented at the given scale. Small polygons that 'fulfil' the above criterion were eliminated by merging them with neighbouring polygons that have the largest area or the longest shared border. As a result, the total number of terrain units per country, as detailed in Table 2, will be less than would have been obtained for a 'raw' clip with the original layer.

The full clipping and overlaying procedure is described in Appendix 2. Scripts were written in R (for final figures) and Python using tools from the ArcPy package for geo-processing. The scripts are provided in Appendix 2 to make the procedure transparent and reproducible.

The original layers like 'Outline of Danube basin' and 'SOTER (terrain) units for whole area' required post-processing as described in Appendix 2, step 3.

3. Results

Table 2 gives an overview of terrain units mapped for each country. The corresponding GIS attribute tables were exported to Ms Excel (XX_SOTER.xlsx, resp. XX_SOTER.xls). Next the xls files were imported into an 'empty' SOTER attribute database for the corresponding country (see Batjes and Ribeiro 2019a) and appended to the terrain table. The resulting 'empty' SOTER attribute databases are named XX_SOTER.mdb, where XX is the ISO country code. Together with corresponding GIS files these are available to project partners via a protected website; login procedures are described in Appendix 3. These materials, plus the accompanying reports, provide the basis for the actual data compilation by the respective partners in the Danube basin.

Table 2. Number of terrain units mapped per country

Country	Number of terrain units	Average area (km²)	Total area (km²)	Minimum area (km²)	Maximum area (km²)
AT	1020	79.05	80630.54	1.69	2422.1
BA	504	74.24	37415.33	1.61	902.21
BG	682	69.23	47217.18	1.66	1201.91
CZ	301	70.22	21135.7	1.75	1232.8
DE	656	85.79	56278.71	1.64	1858.51
HR	432	80.68	34854.28	1.61	1771.85
HU	801	116.12	93012.98	1.72	11174.27
MD	185	67.1	12413.54	2.31	838.36
ME	111	61.38	6812.94	1.72	587.24
RO	2546	93.6	238308.9	1.7	13206.49
RS	899	90.95	81762.68	1.9	3178.33
SI	248	66.21	16419.25	1.74	746.43
SK	585	80.44	47060.2	1.79	1111.97
UA	386	77.19	29794.72	2.08	1724.77

For each country, an html file (e.g. AT_soter.html) was created to facilitate the pre-visualization of the files using the information for elevation, relief intensity and slope as attribute information; results can be explored in Appendix 3.

4. Towards SOTER data compilation

This report with data files corresponds with deliverable D.A3 of the Danube basin project. It should be consulted in conjunction with the preceding reports on the SOTER methodology and database (Batjes and Ribeiro 2019a, b). These materials provide the basis for compiling the actual SOTER databases for the fourteen Danube Basin countries.

Filling of the respective country SOTER attribute database(s) will be done by the regional partners, under a subcontract with Cranfield University, in the context of Work Package 3 on data compilation and WRB correlation. This will be done according to the 'terrain units with one terrain component with one or more soil components' case, as adopted for the SOTER Danube project (Batjes and Ribeiro 2019b).

Meanwhile ISRIC will develop an application to generate XML/GML files for the country SOTER databases according to INSPIRE specifications. In first instance, the procedure will be developed and tested using a hypothetical example (deliverable D.A6). Towards the end of the project (Work package 3, month 18), ISRIC will provide a tool to generate (GML) files for the SOTER's compiled by the 14 partners, as well as the merged SOTER database for the whole Danube basin (D.B2).

Acknowledgments

Funding for the SOTER Danube Basin project was obtained through EU-tender RC/IPR/2017/D.3/0002/OC. The project is implemented under EU-JRC Service Contract #936108.

The SOTER Danube basin consortium is led by Cranfield University (Centre for Environmental and Agricultural Informatics) in partnership with to ISRIC – World Soil Information and 14 regional partners from countries in the Danube basin (Austria (AU), Bosnia and Herzegovina (BA), Bulgaria (BG), Czech Republic (CZ), Germany (DE), Hungary (HU), Croatia (HR), Moldova (MD), Montenegro (ME), Romania (RO), Serbia (RS), Slovakia (SK), Slovenia (SI), and Ukraine (UA).

We thank Dr Stephen Hallett (Cranfield University) for his editorial comments.

References

Batjes NH and Ribeiro E 2019a. *Updated MS-Access SOTER template for the EU Danube basin project with worked through examples (Deliverabes A1, A2 and A4 EU JRC Service Contract ##936108, Collection of soil data in SOTER format from 14 Danube strategy countries, at scale 1:250 000.)*, ISRIC - World Soil Information, Wageningen

Batjes NH and Ribeiro E 2019b. *Guidelines for compiling a 1:250,000 SOTER database (Deliverabe A5, EU JRC Service Contract #936108, Collection of soil data in SOTER format from 14 Danube strategy countries, at scale 1:250 000)*, ISRIC - World Soil Information, Wageningen

eSOTER 2012. eSOTER database at scale 1:1 million. https://www.esoter.net/products

Van Engelen VWP 2011. Standardizing soil data (*e-SOTER* regional pilot platform as EU contribition to a Global Soil Information System). *International Innovation* 6, 48-49. https://cordis.europa.eu/project/rcn/88557/factsheet/en

van Engelen VWP and Dijkshoorn JA 2013. *Global and National Soils and Terrain Digital Databases (SOTER) - Procedures manual (Ver. 2.0)*. ISRIC Report 2013/04, IUSS, ISRIC and FAO, Wageningen, 198 p. http://www.isric.org/sites/default/files/isric_report_2013_04.pdf

Appendices

Appendix 1. Description of the Danube basin Terrain Units layer

(Source: Marc Van Liedekerke, JRC)

The Danube Basin Terrain Units layer at scale 1:250,000 has been created using a methodology similar to the one developed during the eSOTER (2012) FP7 project. The resulting layer is an ESRI® shapefile, the polygons of which delineate Terrain Units (TU) as defined in the SOTER methodology at scale 1:250,000.

The shapefile contains a number of attributes of which the TUID, being the unique Terrain Unit identifier is the most important; the other attributes are the results of various input and subsequent computations that have led to this layer. They can be regarded as additional information, which may be useful when compiling the actual SOTER attribute databases for each Danube Basin country.

TUID [a]	Unique identifier for the Terrain Unit [b]	
HYPSCLASS	Hypsometry classes (elevation above sea level in metres)	hypsclass codes
	0-10	1
	10-50	2
	50-100	3
	100-200	4
	200-300	5
	300-600	6
	600-1500	7
	1500-3000	8
	3000-5000	9
	5000+	10
SLOPECLASS	Slope percentage (%)	Slopeclass code
	0-2	1
	2-5	2
	5-10	3
	10-15	4
	15-30	5
	30-45	6
	45+	7
RICLASS	Relief Intensity (elevation range (m) within 1 km^2)	Class codes
	0-5	1
	5-20	2
	20-50	3
	50-100	4
	100-300	5
	300+	6
SURFCOND	Surface condition, consolidated and unconsolidated parent material Values: Unconsolidated, Consolidated, Water	
GENETICS	Values: Eluvial-colluvial, Marine, N/A	
TEXTURE2	Surface texture class: Values: Clay, Gravel, Loam, Sand, n/a (refers to polygons with water and peat, where texture is not appropriate)	

[a] Known as SUID, see van Engelen and Dijkshoorn (2013).

[b] Note: The maximum length of the TUIDs numbers in the shapefiles exceeds the SOTER default (for SUID). Hence, the format has been set to 'Long Integer' in the MS Access database to be compatible with the terrain unit shapefile provided for the Danube basin.

Appendix 2. Procedure for delineating country terrain unit shapefiles

Step 1 : Danube basin adaptation to fit with the countries boundaries

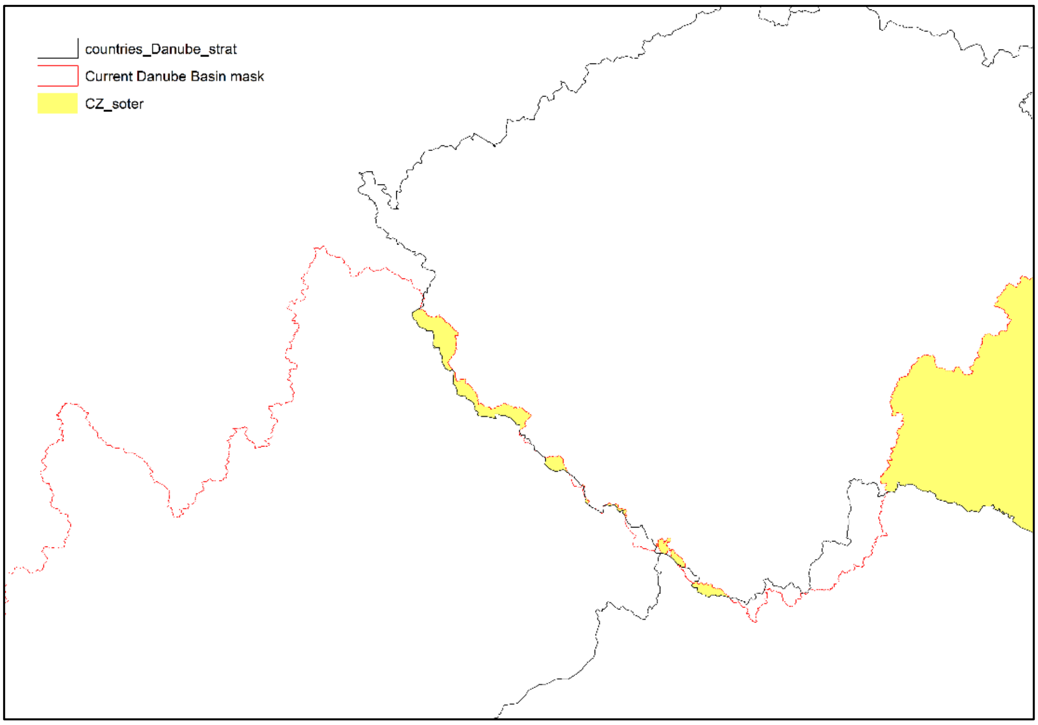


Figure 3. Masks issue (mismatch between boundaries of Danube basin and countries boundaries)

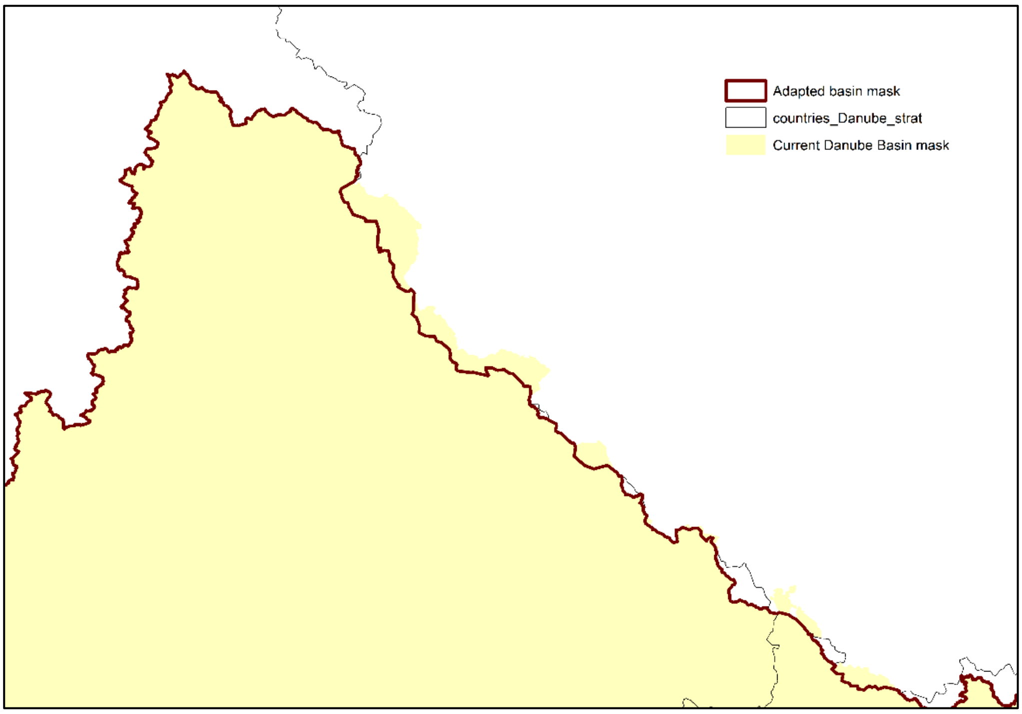


Figure 4. Solution (keep only the area common to the basin mask and the country boundaries and eliminate small polygons along the borders)

```python
##################################################################################
##########
## Adapting Danube Basin Mask
## M. Ruiperez-Gonzalez
## ISRIC - World Soil Information
## Version 1 -  March 2019
## Project: EU-JRC Service Contract #936108; Deliverable D.A3
##################################################################################
##########

### initialization
import arcpy, os
import numpy
from arcpy import env
from arcpy.sa import *

### settings
arcpy.env.overwriteOutput = True
arcpy.CheckOutExtension("spatial")
arcpy.env.workspace = "E:/GIS_2019/Danube-basin-outline-and-14countries-
boundaries/"
#path = "E:/GIS_2019/Danube-basin-outline-and-14countries-boundaries/"

#### define auxiliary function
# update area (km2)
def update_area(input, field):
    arcpy.AddField_management(input, field, "Double")
    expression = "{0}".format("!SHAPE.area@SQUAREKILOMETERS!")
    arcpy.CalculateField_management(input, field, expression, "PYTHON", )
    return(input)

### intermediate outputs path and name
union = "E:/GIS_2019/Intermediate/union.shp"
union_multi = "E:/GIS_2019/Intermediate/union_multi.shp"

### processing
# union of the original Danube mask  with the countries boundary and update area
field
arcpy.Union_analysis("countries_Danube_strat.shp;Danube_mask.shp", union, "ALL",
"", "GAPS")
arcpy.MultipartToSinglepart_management(union, union_multi) # union --> multipart
update_area (union_multi, field="area")

# remove small polygons <  300 km2 that overlay with the  basin mask and  the
country boundaries
arcpy.MakeFeatureLayer_management(union_multi, "layer") ; size = "300"  # make a
feature layer
arcpy.SelectLayerByAttribute_management("layer", "NEW_SELECTION", '"FID_Danube" <>
' + str("-1")  ) # select polygons that overlay in the Danube mask; id <> -1
arcpy.SelectLayerByAttribute_management("layer", "SUBSET_SELECTION", '"FID_countr"
<> ' + str("-1") ) # select from those polygons that overlay in the country
boundary mask; id <> -1
arcpy.SelectLayerByAttribute_management("layer", "SUBSET_SELECTION", '"area" > ' +
str(size) ) # select from those the ones bigger than 300 km2
arcpy.CopyFeatures_management("layer", "final.shp") # export  the selected polygons
arcpy.Dissolve_management("final.shp", "Danube_mask_adapted.shp") # dissolve the
output layer

##################################################################################
########## end of script;
```

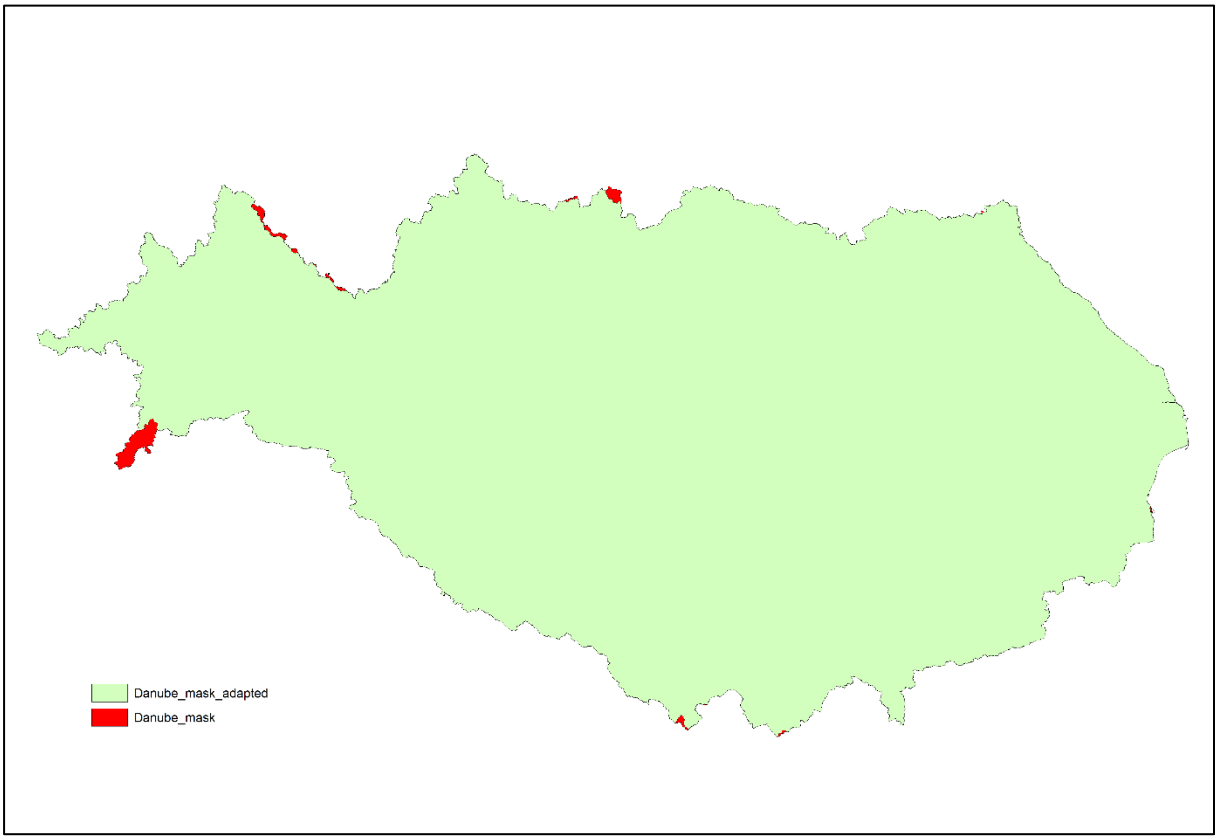


Figure 5. Differences between the masks

Step 2: Eliminate small polygons (≤ 1.6 km^2) from the SOTER terrain units for whole area

There are 2389 polygons ≤ 1.6 km^2 in the original layer corresponding with 739.8 km2 of the total area of the original layer. These were 'eliminated' by merging them with neighbour polygons. After the clip with the basin boundary layer, 837 terrain units were merged with a bigger unit. In total this corresponds with 339.21 km^2 of the total area (803116.9 km^2).

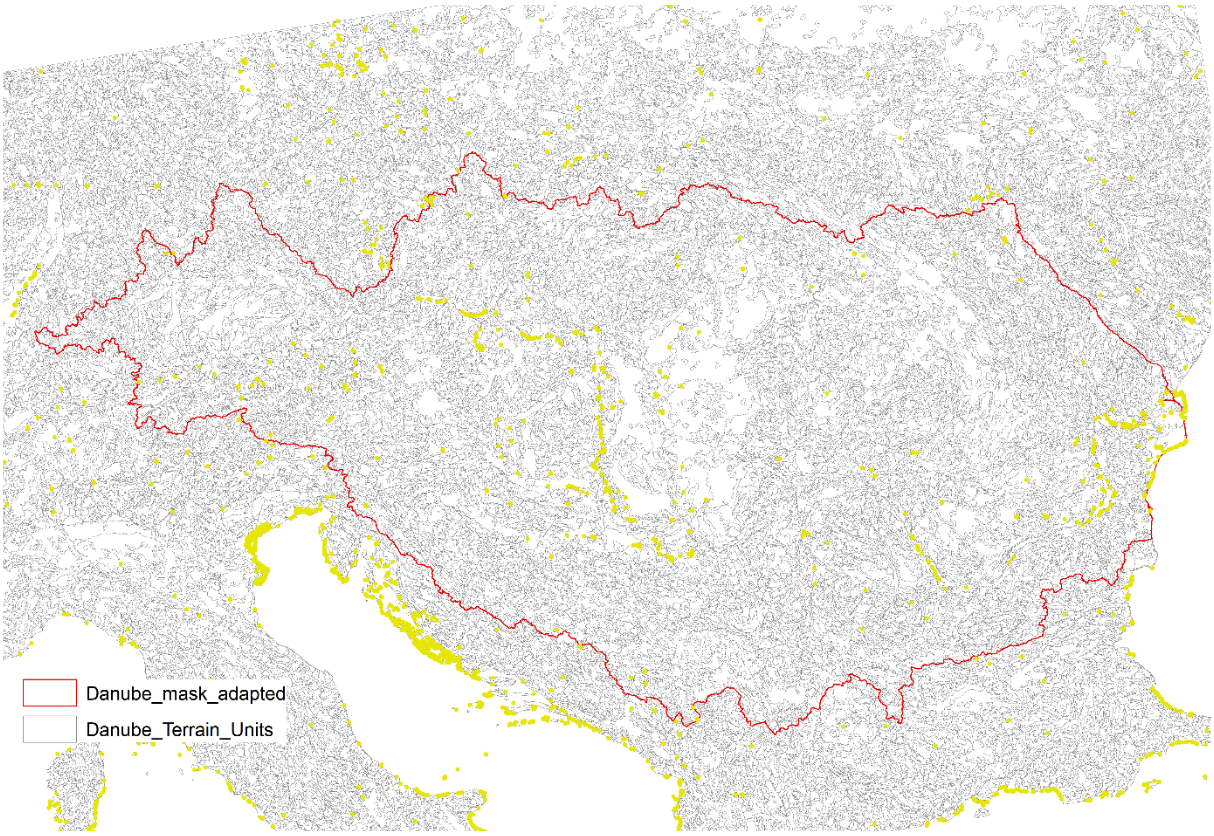


Figure 5. Small polygons in yellow (≤ 1.6 km²)

```
################################################################################
##########
## Eliminating <= 1.6 km2 terrain units from the original terrain units file
## M. Ruiperez-Gonzalez
## ISRIC - World Soil Information
## Version 1 - March 2019
## Project: EU-JRC Service Contract #936108; Deliverable D.A3
################################################################################
##########

### summary
# 2389 units of the original file have area_mvl <= 1.6 km²

### initialization
import arcpy, os
import numpy
from arcpy import env
from arcpy.sa import *

#### settings
path = "E:/GIS_2019/"
arcpy.env.workspace = str(path) + "/_scratch/" # used for intermediate results
arcpy.env.overwriteOutput = True

#### define auxiliary function
# update area (km2)
def update_area(input, field):
    arcpy.AddField_management(input, field, "Double")
    expression = "{0}".format("!SHAPE.area@SQUAREKILOMETERS!")
    arcpy.CalculateField_management(input, field, expression, "PYTHON", )
    return(input)

# eliminate polygons by merging them with neighboring polygons that have the
```

```python
largest area or the longest shared border. Uses an area threshold.
def eliminate_smallpoly(input, output, threshold):
    arcpy.MakeFeatureLayer_management(input, "layer")
    arcpy.SelectLayerByAttribute_management("layer", "NEW_SELECTION", '"area" <= '
+ str(threshold))
    arcpy.Eliminate_management("layer", "layer_.shp", "LENGTH")
    arcpy.Dissolve_management("layer_.shp",
output,"TUID;HYPSCLASS;SLOPECLASS;RICLASS;SURFCOND;GENETICS;CARBONATE;TEXTURE2;",
"", "SINGLE_PART", "DISSOLVE_LINES")
    update_area(output,  "area")
    return (output)

#### processing
# mask the terrain units file using the Danube mask adapted
mask = str(path) + "/Danube-basin-outline-and-14countries-
boundaries/Danube_mask_adapted.shp" # basin mask adapted to the country boundary
arcpy.Clip_analysis(str(path) + "/Danube_Terrain_Units/Danube_Terrain_Units.shp",
mask,  "tu_v0.shp", "") # clips the soter map with the country boundary
update_area("tu_v0.shp",   "area") # update the area

# eliminate small polygons using a threshold
input = "tu_v0.shp";     output = "_tu_v0.shp";    threshold = "1.6"

# apply eliminate_smallpoly function N times
n = 3 # needs some iteration to eliminate all the small polygons

for x in range(n):
    eliminate_smallpoly(input, output, threshold)
    arcpy.CopyFeatures_management(output,input)  # recycle the input name in case
we need to iterate on this shapefile
    if x == n:
        break

# copying the file in the inputs folder
arcpy.CopyFeatures_management(input,str(path) +
"/Danube_Terrain_Units/Danube_Terrain_Units_adapted.shp")  # recycle the input name
in case we need to iterate on this shapefile

##################################################################################
######### end of script;
```

Step 3: Clip terrain units map using the fourteen target country boundaries and the adapted basin mask

Main steps in the process were to:

- clip country boundaries with adapted basin mask
- split per country (using CNTR_ID); 14 target countries
- extraction of terrain units that overlay with each of the countries boundaries; loops through the countries files.
- apply same process looping through countries
- organize fields in the attribute table; add new fields such as country code and ISOCSUID code (concatenation of country code and TUID).
- eliminate polygons by merging them with neighbouring polygons that have the largest area or the longest shared border. Uses an area threshold.

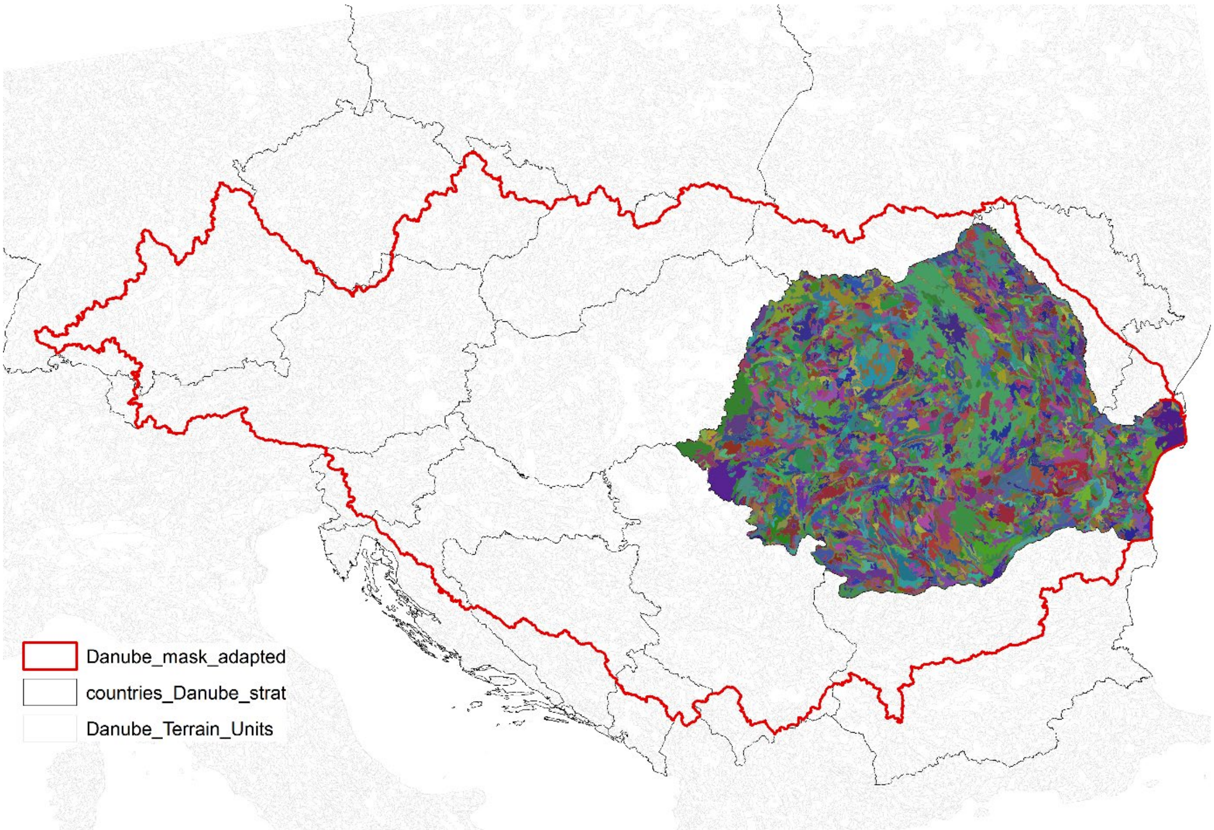


Figure 6. Romania terrain units clip (example)

```
################################################################################
##########
## M. Ruiperez-Gonzalez
## ISRIC - World Soil Information
## Version 1 - March 2019
## Project: EU-JRC Service Contract #936108; Deliverable D.A3
################################################################################
##########

#### import libraries
```

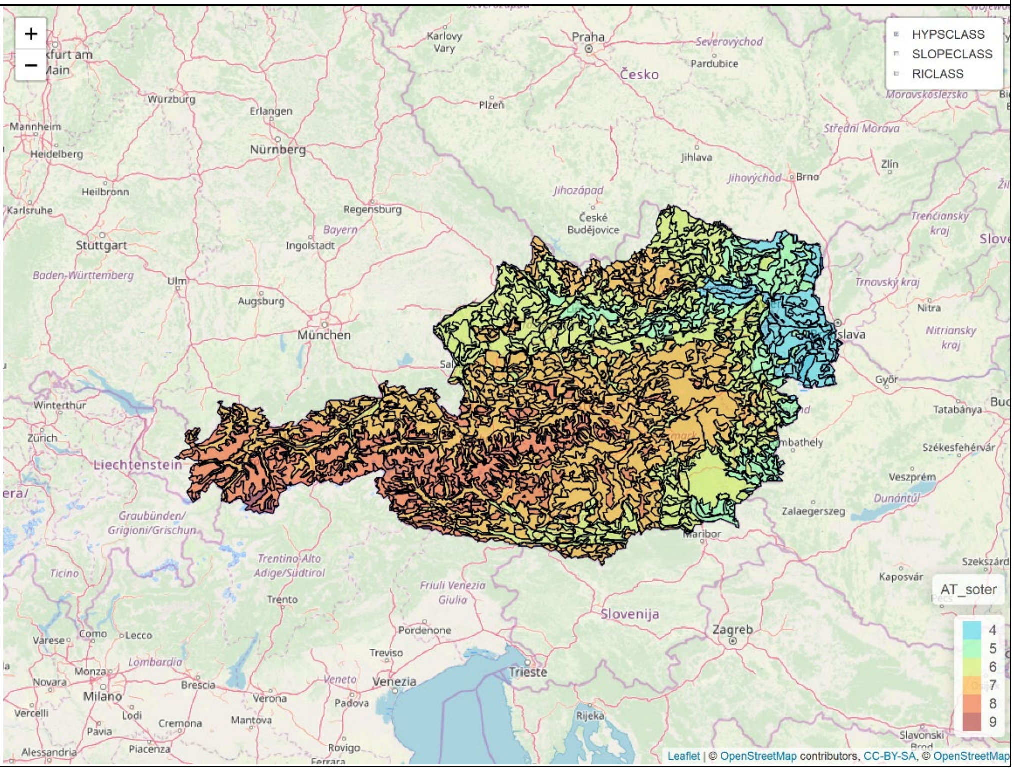
```
import arcpy, os
import numpy
from arcpy import env
from arcpy.sa import *
arcpy.CheckOutExtension("spatial")

#### settings
path = "E:/GIS_2019/"
output_path = str(path) + "/Deliverable_D.A3/"
arcpy.env.overwriteOutput = True

#### start with a clean folders structure
arcpy.Delete_management(str(path) + "/_scratch/")
arcpy.CreateFolder_management(str(path), "/_scratch/")
arcpy.env.workspace = str(path) + "/_scratch/" # used for intermediate results

#### define auxiliary function
# update area (km2)
def update_area(input, field):
    arcpy.AddField_management(input, field, "Double")
    expression = "{0}".format("!SHAPE.area@SQUAREKILOMETERS!")
    arcpy.CalculateField_management(input, field, expression, "PYTHON", )
    return(input)

# eliminate polygons by merging them with neighbouring polygons that have the
largest area or the longest shared border. Uses an area threshold.
def eliminate_smallpoly(input, output, threshold):
    arcpy.MakeFeatureLayer_management(input, "layer")
    arcpy.SelectLayerByAttribute_management("layer", "NEW_SELECTION", '"area" <= '
+ str(threshold))
    arcpy.Eliminate_management("layer", "layer_.shp", "LENGTH")
    arcpy.Dissolve_management("layer_.shp",
output,"ISOCSUID;TUID;COUNTRY;HYPSCLASS;SLOPECLASS;RICLASS;SURFCOND;GENETICS;CARBON
ATE;TEXTURE2;", "", "SINGLE_PART", "DISSOLVE_LINES")
    update_area(output,  "area")
    return (output)

#### paths for input data
mask = str(path) + "/Danube-basin-outline-and-14countries-
boundaries/Danube_mask_adapted.shp" # basin mask adapted to the country boundary
admin =  str(path) + "/Danube-basin-outline-and-14countries-
boundaries/countries_Danube_strat.shp"
#soter = str(path) + "/Danube_Terrain_Units/Danube_Terrain_Units_adapted.shp"

#### processing

## clip country boundaries with basin mask
arcpy.Clip_analysis(admin, mask, str(path) + "/Danube-basin-outline-and-
14countries-boundaries/admin_masked.shp")

## split per country (using CNTR_ID); 14 target countries
arcpy.Split_analysis(str(path) + "/Danube-basin-outline-and-14countries-
boundaries/admin_masked.shp", str(path) + "/Danube-basin-outline-and-14countries-
boundaries/admin_masked.shp", "CNTR_ID", str(arcpy.env.workspace), "")

# extraction of  terrain units that overlay with each of the countries boundaries;
loops through the countries files.
# apply process in a loop; define expression and list of fields
#expression = "{0}".format("!SHAPE.area@SQUAREKILOMETERS!")
#field = "area"
# define field list for dissolve
fieldList = ["ISOCSUID", "TUID", "COUNTRY",
"HYPSCLASS","SLOPECLASS","RICLASS","SURFCOND","GENETICS","CARBONATE","TEXTURE2"]
fieldListDiss =
"ISOCSUID;TUID;COUNTRY;HYPSCLASS;SLOPECLASS;RICLASS;SURFCOND;GENETICS;CARBONATE;TEX
TURE2"

## apply process looping through countries
```

```
for shapefile in arcpy.ListFiles("*.shp"): # done per country

    name = shapefile.split(".")[0] # variable with the country name
    arcpy.Clip_analysis(str(path) +
"/Danube_Terrain_Units/Danube_Terrain_Units_adapted.shp", str(shapefile),
"layer0.shp", "") # clips the soter map with the country boundary

    # organize fields in the attribute table
    arcpy.AddField_management("layer0.shp", "COUNTRY", "TEXT") # add a new field
for the country code
    arcpy.CalculateField_management("layer0.shp", "COUNTRY", '"{}"'.format(name),
"PYTHON_9.3")  # calculate country code
    arcpy.AddField_management("layer0.shp", "ISOCSUID", "TEXT") # add a new field
for the ISOCSUID code
    arcpy.CalculateField_management("layer0.shp", "ISOCSUID",
'"{}{}"'.format(name,'!TUID! '), "PYTHON_9.3") # calculate ISOCSUID code
    arcpy.Dissolve_management("layer0.shp", "layer2.shp", "ISOCSUID;", "",
"SINGLE_PART", "DISSOLVE_LINES")

    # ISOCSUID as first column in the attribute table
    inFeatures = "layer2.shp";     joinField = "ISOCSUID";    joinTable =
"layer0.shp"
    # join  by the ISOCSUID code and keep fields of interest
    arcpy.JoinField_management(inFeatures, joinField, joinTable,
joinField,fieldList)
    # update area field
    update_area(inFeatures,  "area")

    # create a country folder to store results
    arcpy.CreateFolder_management(str(output_path), name)  # creates a folder for
the country in the deliverables folder
    input = "layer2.shp";     output = "layer3.shp";    threshold = "1.6"

    # apply eliminate_smallpoly function N times
    n = 3 # needs some iteration to eliminate all the small polygons

    for x in range(n):
        eliminate_smallpoly(input, output, threshold)
        arcpy.CopyFeatures_management(output,input)  # recycle the input name in
case we need to iterate on this shapefile
        if x == n:
            break

    # update area field
    arcpy.Dissolve_management(input, str(output_path) + "/" + name + "/" +
str(name) + "_soter.shp", str(fieldListDiss), "", "MULTI_PART", "DISSOLVE_LINES")
    update_area(str(output_path) + "/" + name + "/" + str(name) + "_soter.shp",
"area")

###################################################################################
######### end of script;
```

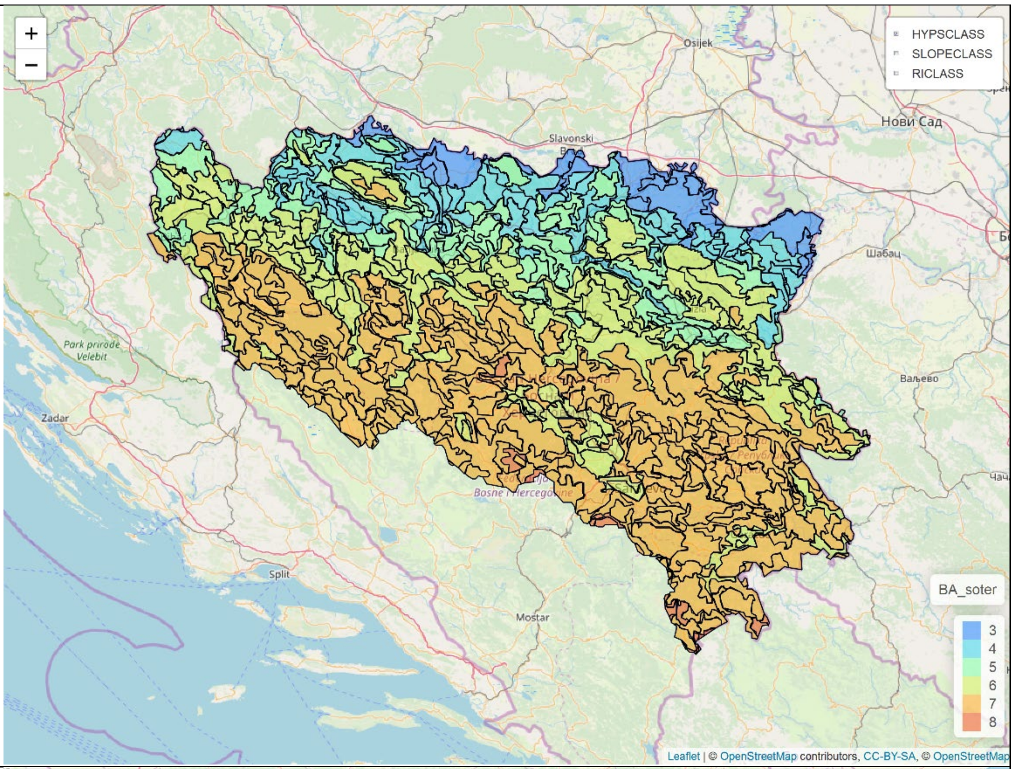
Appendix 3. Results

For each country there is an html file that presents a quick overview for the country. The HYPSCLASS, SLOPECLASS and RICLASS (see Appendix 1) can be displayed on screen.

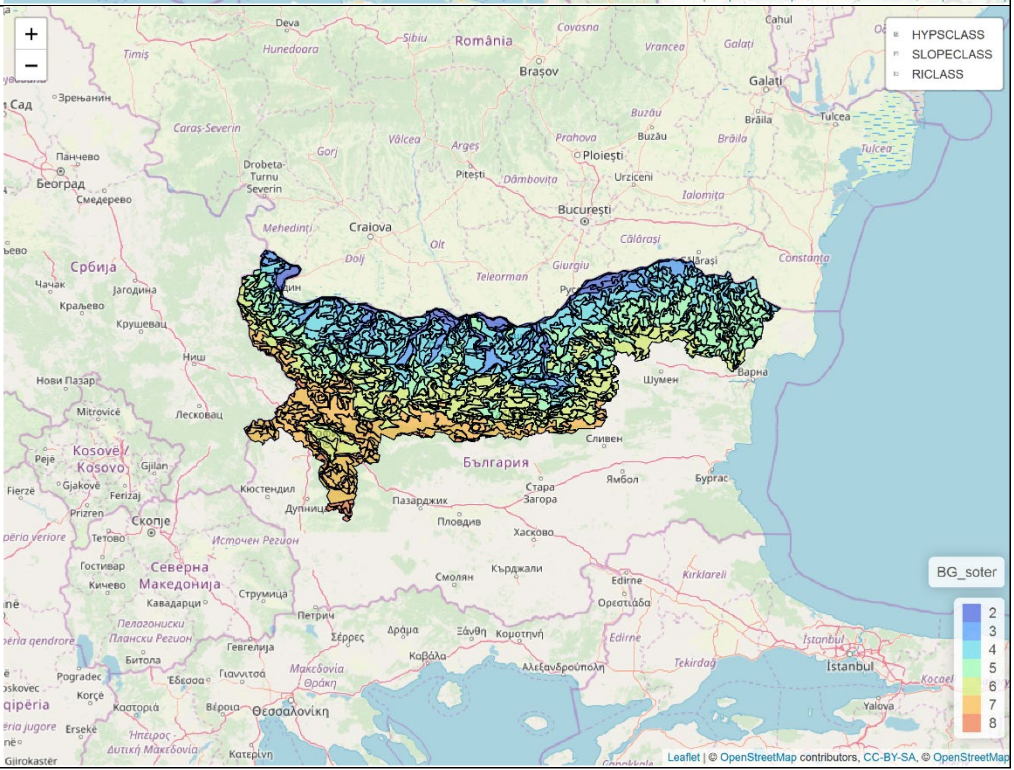
These files are named XX_soter.html and stored in folder \XX\html_files, where XX is the country ISO code.

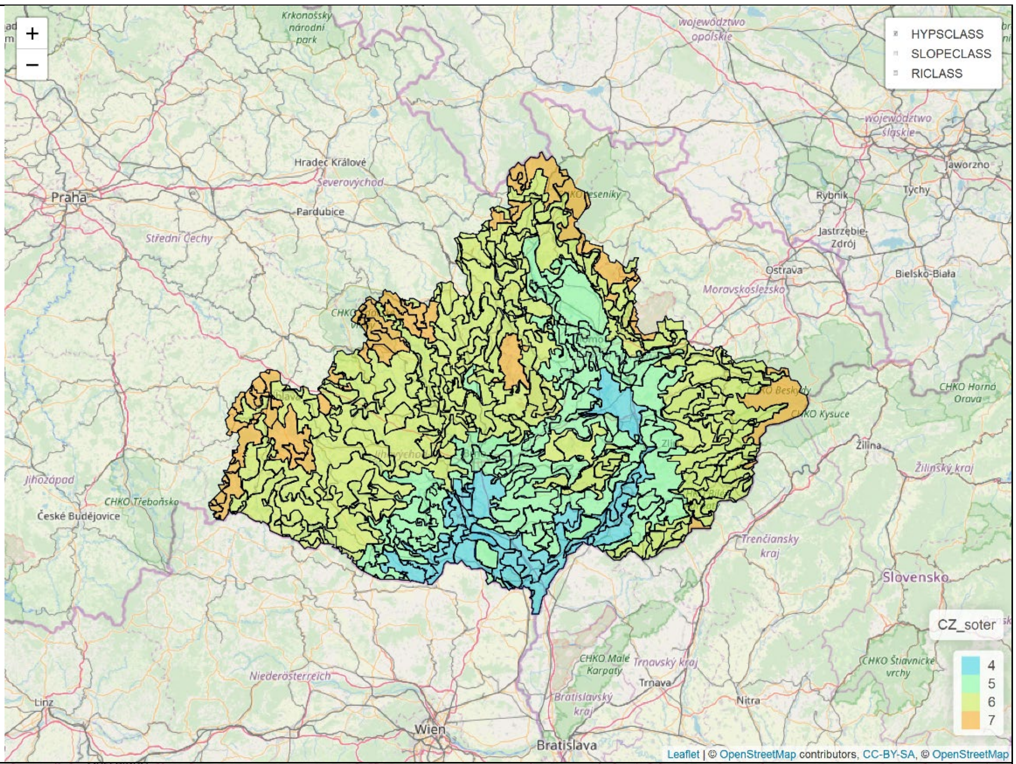
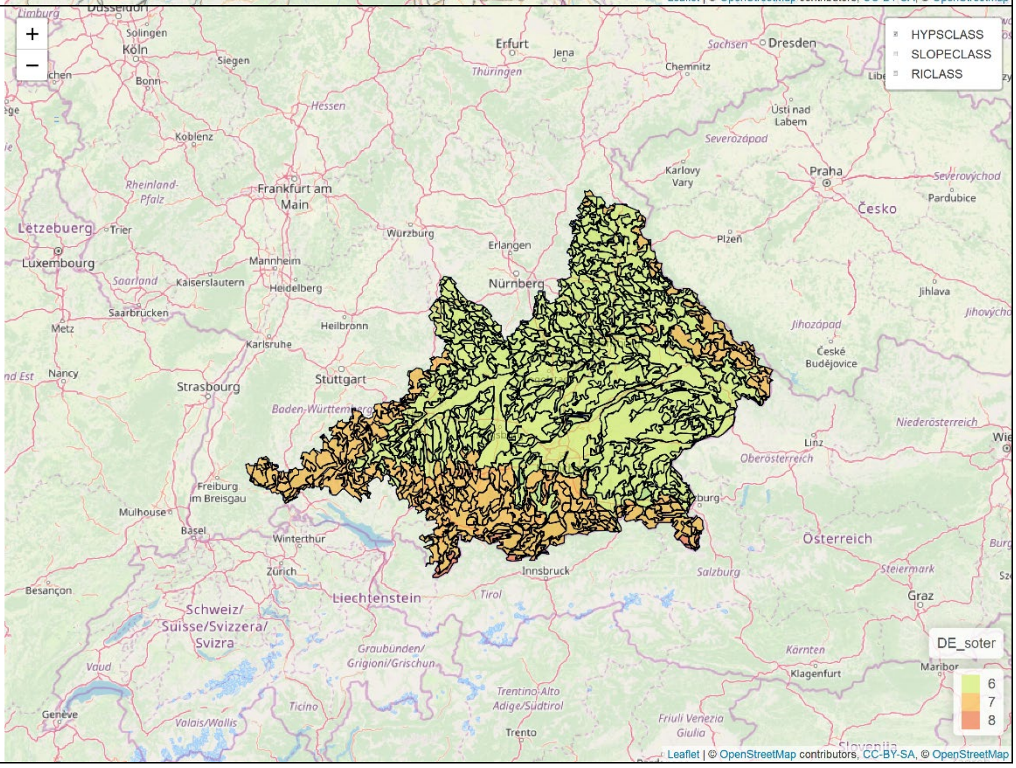
Country[a]	**SOTER terrain units delineations** (showing HYPSCLASS)
AT	

BA


HYPSCLASS
SLOPECLASS
RICLASS
BA_soter
3
4
5
6
7
8
Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap

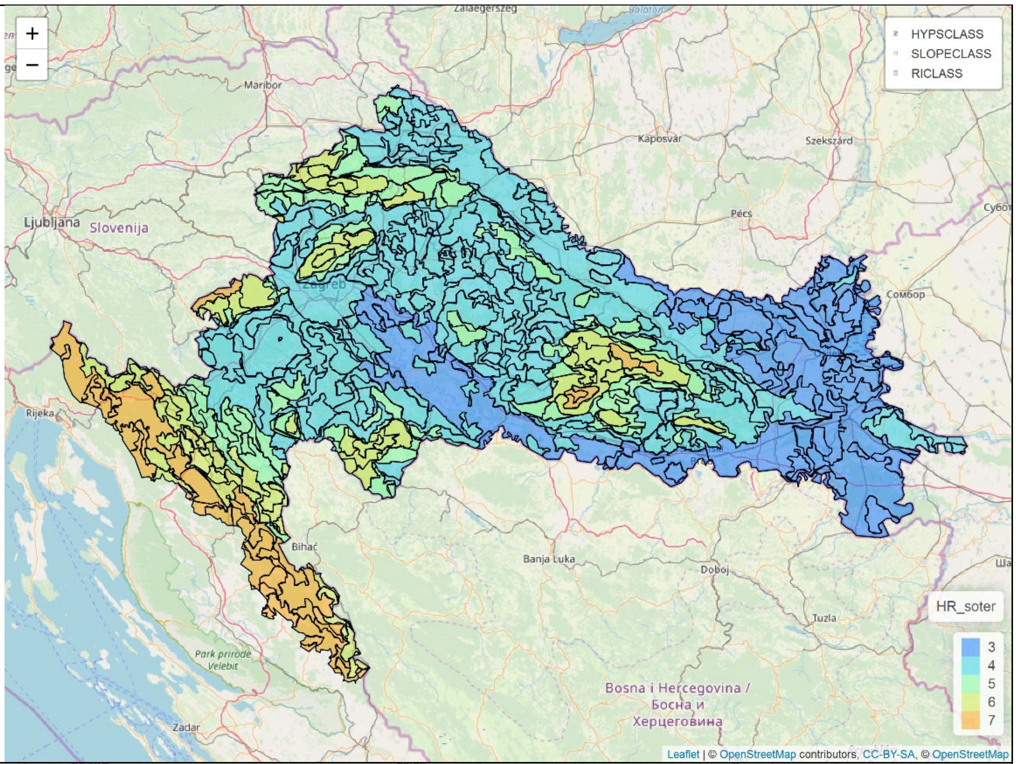
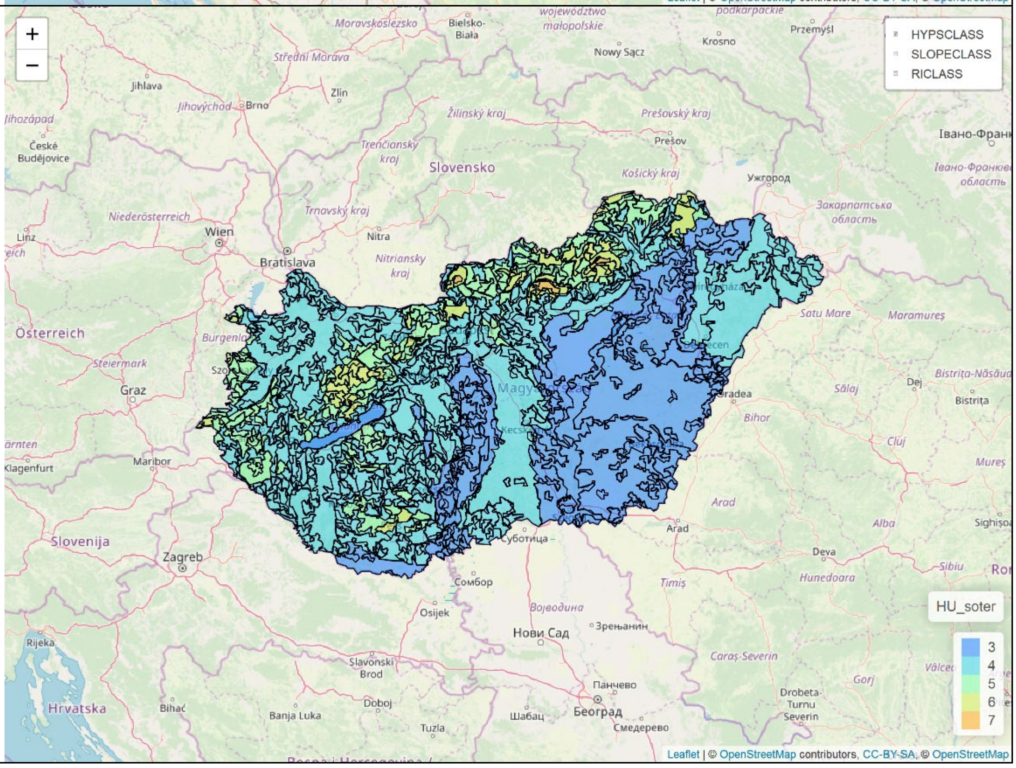
BG


HYPSCLASS
SLOPECLASS
RICLASS
BG_soter
2
3
4
5
6
7
8
Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap

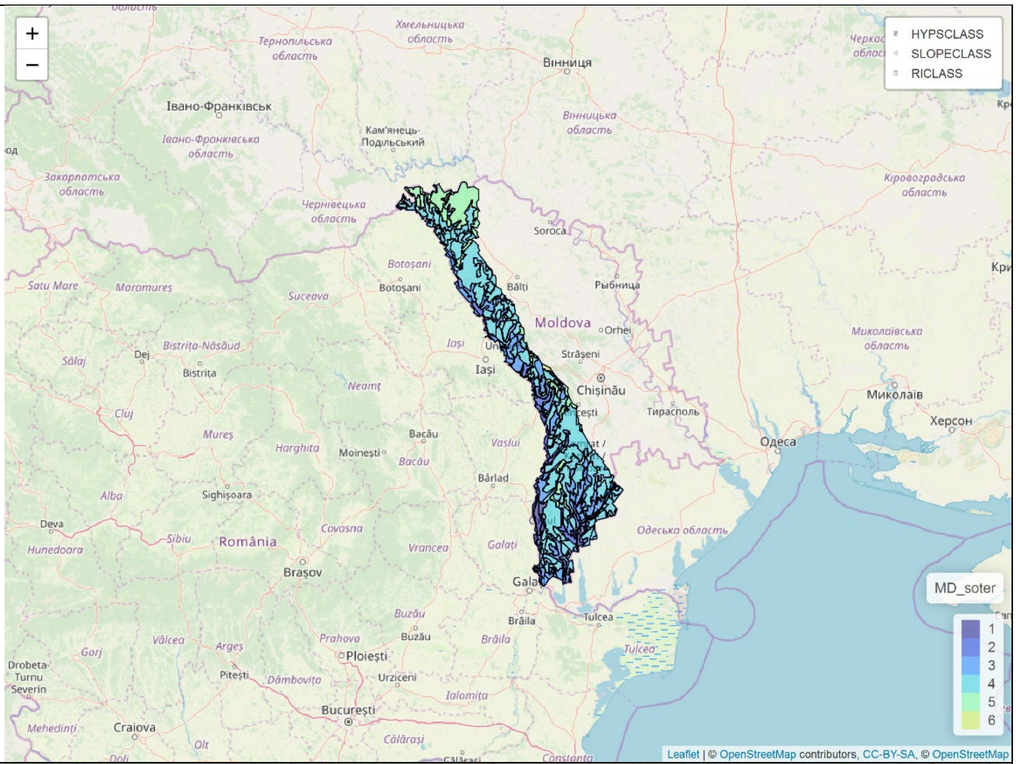
CZ	
DE	

HYPSCLASS
SLOPECLASS
RICLASS
CZ_soter
4
5
6
7
Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap

HYPSCLASS
SLOPECLASS
RICLASS
DE_soter
6
7
8
Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap

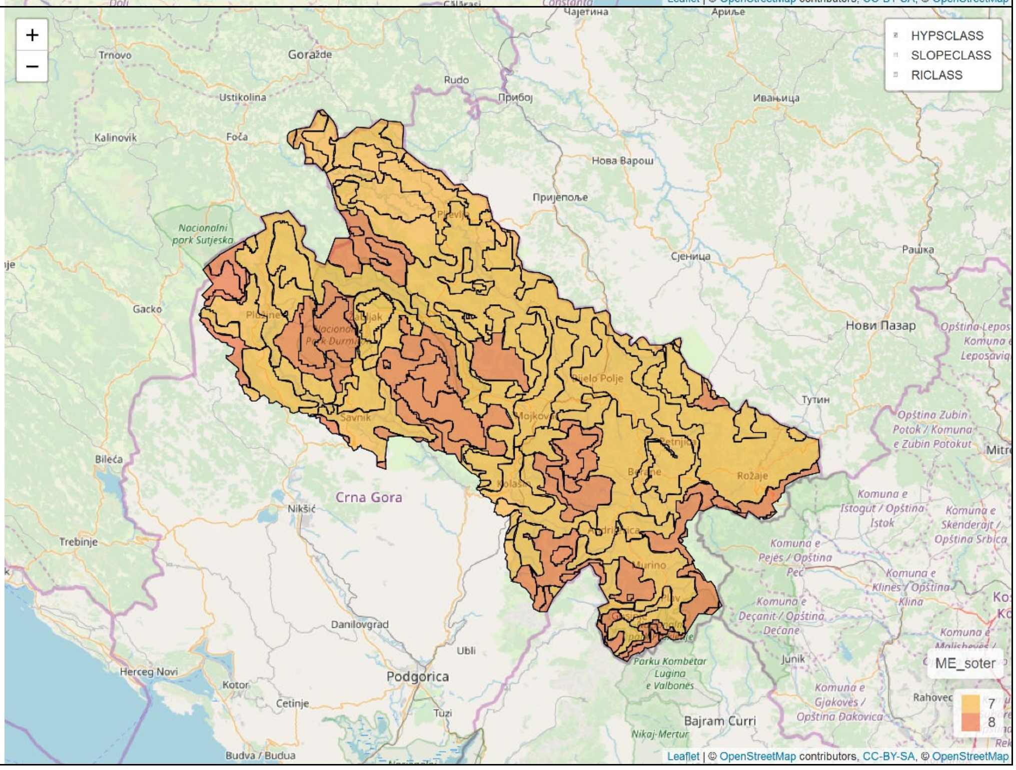
<table>
<tr><td>HR</td><td>
</td></tr>
<tr><td>HU</td><td>
</td></tr>
</table>

MD



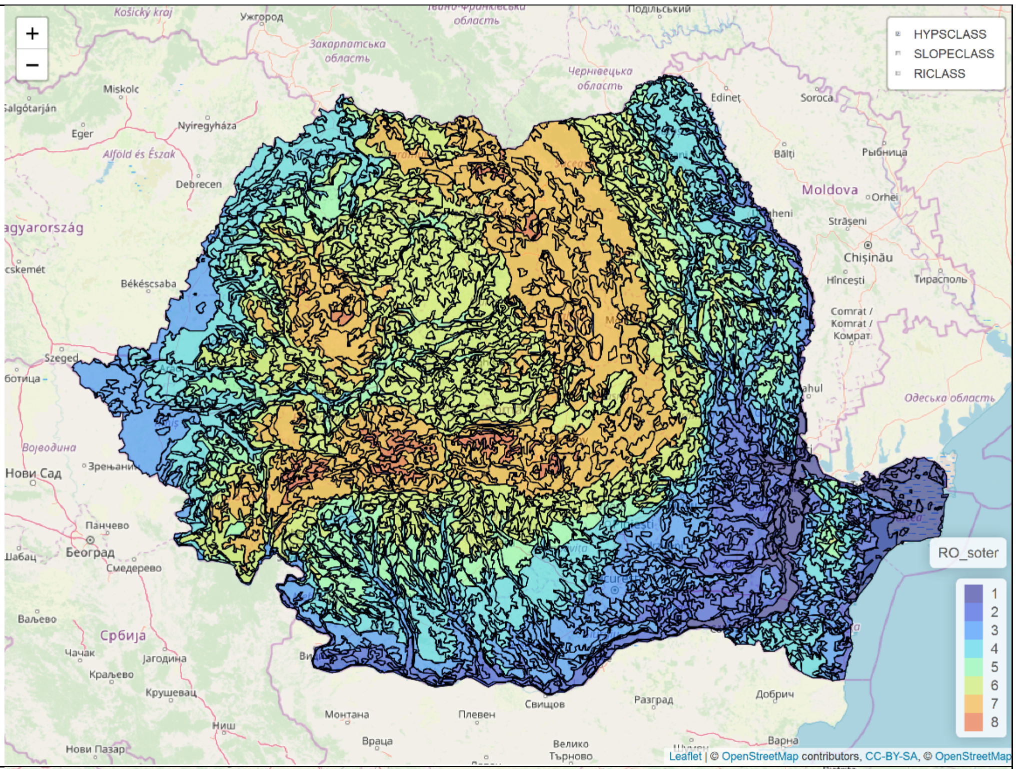
HYPSCLASS
SLOPECLASS
RICLASS
MD_soter
1
2
3
4
5
6
Moldova
Chişinău
România
Iaşi
Bălţi
Botoşani
Braşov
Bucureşti
Ploieşti
Craiova
Bacău
Bârlad
Galaţi
Tulcea
Одеса
Миколаїв
Херсон
Вінниця
Івано-Франківськ
Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap

ME

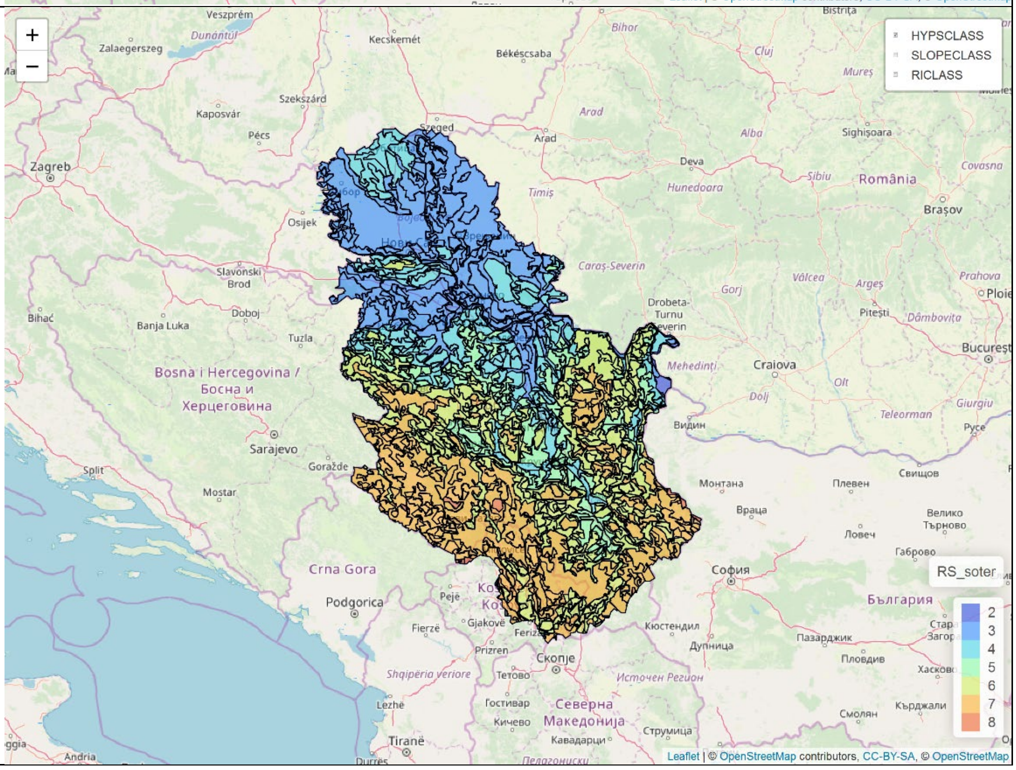


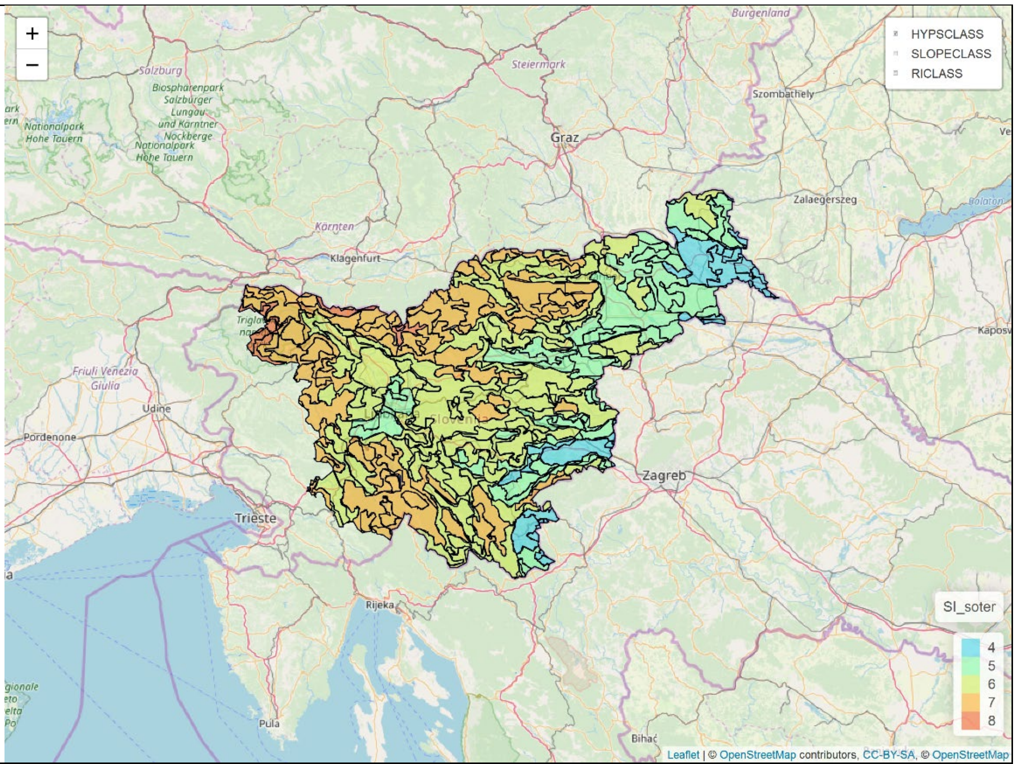
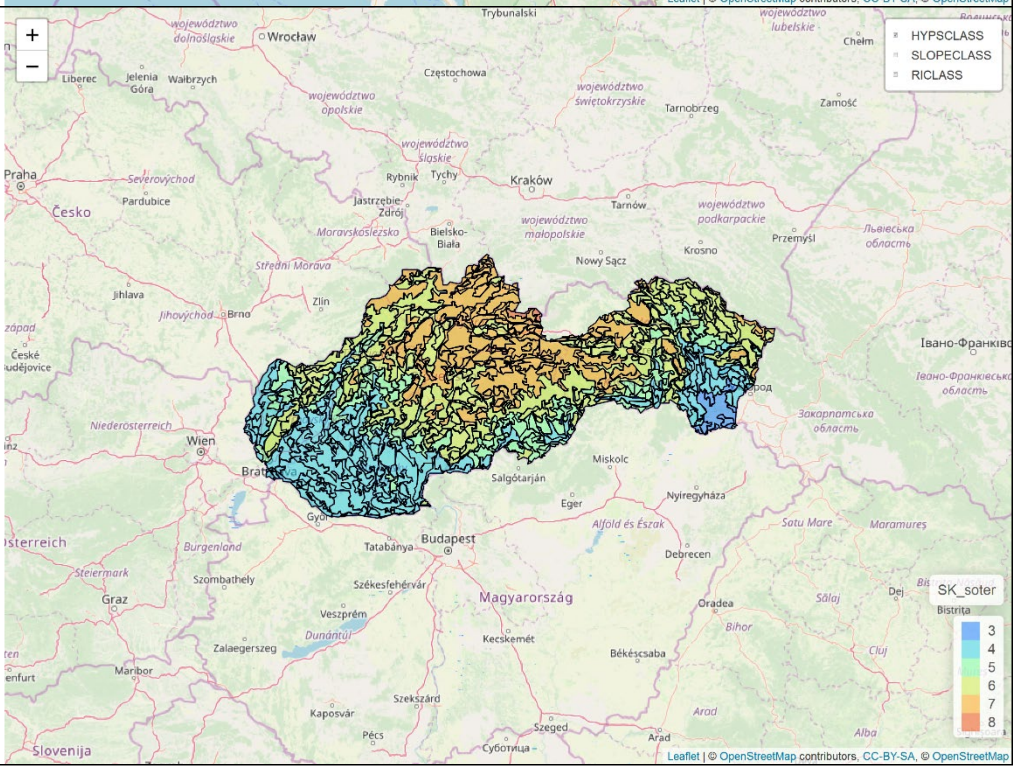
HYPSCLASS
SLOPECLASS
RICLASS
ME_soter
7
8
Crna Gora
Podgorica
Nikšić
Danilovgrad
Herceg Novi
Cetinje
Trebinje
Bileća
Gacko
Foča
Goražde
Нови Пазар
Bajram Curri
Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap

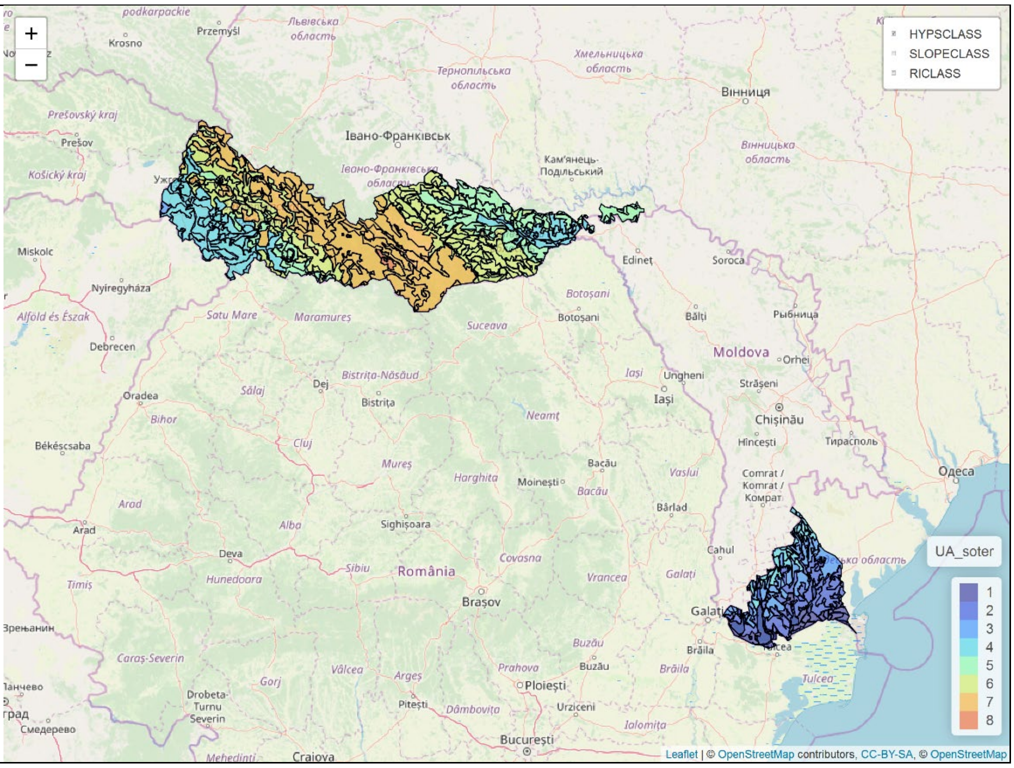
RO


HYPSCLASS
SLOPECLASS
RICLASS
RO_soter
1
2
3
4
5
6
7
8
Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap

RS


HYPSCLASS
SLOPECLASS
RICLASS
RS_soter
2
3
4
5
6
7
8
Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap

SI	HYPSCLASS SLOPECLASS RICLASS SI_soter 4 5 6 7 8 Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap
SK	HYPSCLASS SLOPECLASS RICLASS SK_soter 3 4 5 6 7 8 Leaflet | © OpenStreetMap contributors, CC-BY-SA, © OpenStreetMap

UA	

[a] Part of county falling within the Danube basin.