This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 776804 — H2020-SC5-2017.

NEXT

New Exploration Technologies

DELIVERABLE 4.13

Appendix 1

Technical documentation

Horizon 2020 Project: **NEXT**

Author(s): **Andreas Kempe, Peggy Hielscher**

Institution: **Beak Consultants GmbH**

Date: **27.10.2020**

Disclaimer

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information as its sole risk and liability. The document reflects only the author's views and the Community is not liable for any use that may be made of the information contained therein.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The document describes the software design and the testing procedure of the ESRI ArcGIS SOM Toolbox. The *nextsomcore* is the core of the implementation of SOM in advangeo®. *nextsomcore* was developed by the GTK as part of the European Union funded H2020 project NEXT (see deliverable 4.11). It provides all functions for the processing of SOM as well as interfaces for the data and parameter exchange so that it can be integrated in other software products like advangeo® (see deliverable 4.12) or ESRI ArcMap.

2. SOFTWARE DESIGN

SOM Clustering toolbox and nextsomcore are implemented in python whereas the learning file (LRN) creating and raster data processing tools are implemented in VB.net.

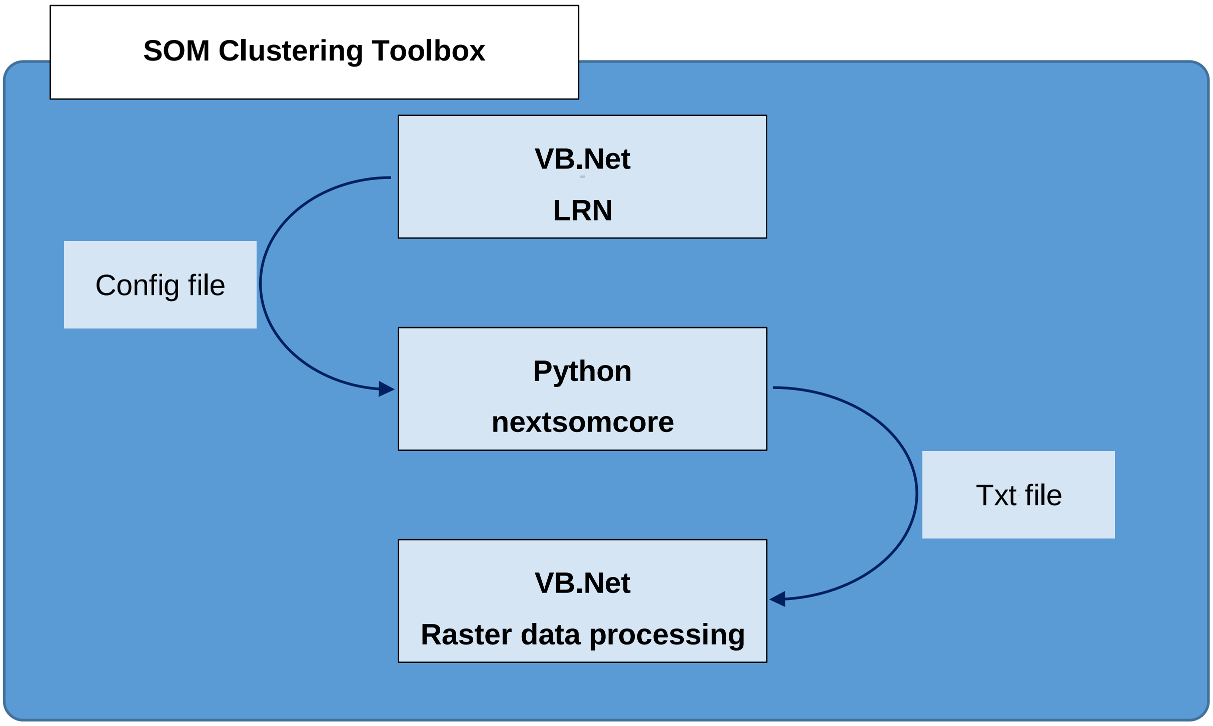


Figure 1. Structure of the self-organizing maps toolbox developed in the NEXT project.

The SOM clustering toolbox contains several functions which are explained below. These are executed consecutively in background while running the script without need for further input. Each function’s result is a requirement for calculating the subsequent function until the final result is created. The following functions appear in the order of execution and can be grouped into pre-processing, processing and post-processing.

All functions are executed by calling the related EXE file except for the raster calculator.

3. SOM TOOLBOX FUNCTIONS

3.1. Input data

Table 1. Input parameters.

Name	Type	Description	Default
Workspace	folder	*Path the working directory*	-
Input raster	Esri Grid	*Features used for SOM calculation*	-
SOM parameter	Integer	Number of cells in X and Y direction, Number of epochs	-
K-means parameter	Integer	Min. number of clusters, Max. number of clusters, Number of initial means	2, 25, 5

3.2. Preprocessing: Creating a configuration file

Purpose:

- The configuration file is mandatory for the nextsomcore function.

Input:

- Path to workspace location
- SOM parameters: number of cells in X and Y direction of the SOM space, number of epochs, optionally for k-means clustering number of clusters and number of initial means

Output:

- XML file: SOM.xml, containing input information from above, workspace path, SOM parameters as number of cells in X and Y direction, number of epochs, and further default properties to see in the example.
- 2 initial TXT files, named somspace.txt and geospace.txt, will be created in the defined workspace which will be filled by the nextsomcore function.

Example:

```
<?xml version="1.0"?>
<som_configuration>
        <som_files>
                <input>\\vs-daten\Projekte\... \SOM.lrn</input>
                <output_somspace>\\vs-daten\Projekte\...\somspace.txt</output_somspace>
                <output_geospace>\\vs-daten\Projekte\...\geospace.txt</output_geospace>
                <output_folder>\\vs-daten\Projekte</output_folder>
        </som_files>
        <som_parameters>
                <som_x>25</som_x>
                <som_y>25</som_y>
                <nEpoch>10</nEpoch>
                <mapType>toroid</mapType>
                <gridType>rectangular</gridType>
                <neighborhood>gaussian</neighborhood>
                <std_coeff>0.5</std_coeff>
                <radius0>0</radius0>
                <radiusN>1</radiusN>
                <tradiuscooling>linear</tradiuscooling>
                <scale0>0.1</scale0>
                <scaleN>0.01</scaleN>
                <scalecooling>linear</scalecooling>
                <verbose>not implemented</verbose>
                <codebook>not implemented</codebook>
                <globalBmus>not implemented</globalBmus>
                <uMatrix>not implemented</uMatrix>
                <compact_support>not implemented</compact_support>
                <kernelType>not implemented</kernelType>
        </som_parameters>
</som_configuration>
```

3.3. Raster calculator

Purpose:

- A raster as mask is created for localizing the extent of only congruent regions. This mask has pixel values either of 1 or NoData whereas 1 is the result of coincident rasters and NoData of raster values from one raster overlapping with NoData in another raster.
- The mask is mandatory for creating an LRN file and again for processing the raster data.

Input:

- Multiple input raster

Output:

- One mask raster (GRID)

Example:

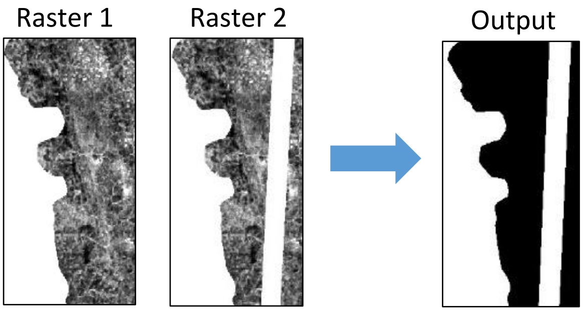


Figure 2. ***Rasters with different regions of NoData (white) result in output with calculated region (black) and NoData (white).***

3.4. Preprocessing: Creating an LRN file

Purpose:

- LRN is a text file format, where data are defined for SOM and k-means computation. The LRN file contains pixel values from the input rasters in table columns.
- The LRN file is mandatory for the nextsomcore function since input can only imported in the LRN format.

Input:

- Path to the LRN file
- Input rasters (for formats see 3.1)
- Mask raster (GRID)

Output:

- LRN file: SOM.lrn

Structure of LRN file:

$\#$ Comment line

% N_{dat}

% N_{col}

$a_1\ a_2\ .....\ a_{Ncol}$

id x y z attr$_1$ attr$_2$ .... attr$_{Natt}$

1 $x_1\ y_1\ z_1\ val_{11}\ val_{12} \ldots val_{1,Natt}$

2 $x_2\ y_2\ z_2\ val_{21}\ val_{22} \ldots val_{2,Natt}$

....

$\boldsymbol{N_{dat}\ x_{Ndat}\ y_{Ndat}\ z_{Ndat}\ val_{Ndat,1}\ val_{Ndat,2} \ldots val_{Ndat,Natt}}$

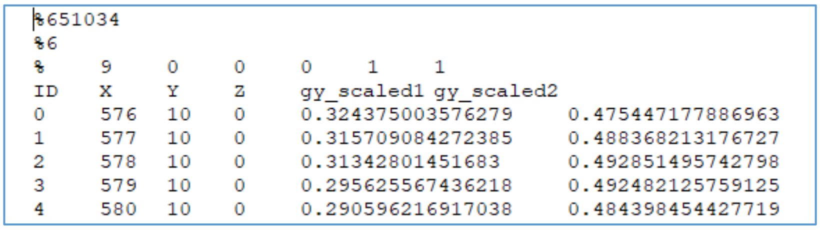
- Optionally, N comment lines starting with a hash sign.
- N_{dat} = Number of data points, i.e., number of rows after the header line. The line begins with a percent sign.

- N_{col} = Total number of items in the header. The line begins with a percent sign. N_{col} items must appear on each data row below the header row.
- a_i = Indicates which columns in the data table, i.e., the rows below the header, are used in SOM training: 0 = do not use, 1 = use. Separator = tab.

Header row indicates the meaning of each column, i.e., data attribute, on the preceding rows containing the data points (separator = tab):

- i_d = unique index of data points. This column is required as the first column.
- x, y, z = optionally, there can be one of more spatial coordinates. The name of the corresponding columns must be defined as “x”, “y” or “z”.
- attri = data attributes
- Data points according to the header. One data point per line. Column separator = tab.

Example:



```
%651034
%6
%     9     0     0     0     1     1
ID    X     Y     Z     gy_scaled1 gy_scaled2
0     576   10    0     0.324375003576279     0.475447177886963
1     577   10    0     0.315709084272385     0.488368213176727
2     578   10    0     0.31342801451683      0.492851495742798
3     579   10    0     0.295625567436218     0.492482125759125
4     580   10    0     0.290596216917038     0.484398454427719
```

Figure 3. ***Exemplary detail of SOM.lrn.***

3.5. nextsomcore

Purpose:

- The nextsomcore function performs the SOM and k-means computations using the external SOM computation package Somoclu, developed by Wittek et al. (2013), and is integrable to other software. The result is written to TXT files according to geospace and SOM-space, the somspace.txt and geospace.txt. These files are mandatory for processing the raster data in the next step.

Input:

- LRN file: SOM.lrn

Output:

- geospace.txt
- somspace.txt

Structure of geospace.txt:

X Y (Z) som_x som_y cluster b_attr$_1$ battr$_2$ .... b_attr$_{Natt}$ attr$_1$ attr$_2$ .... attr$_{Natt}$ qerror
$x_1\ y_1\ z_1\ sx_1\ sy_1\ cl1\ cb_{11}\ cb_{12} \ldots cb_{1,Natt}\ val_{11}\ val_{12} \ldots val_{1,Natt}\ qe_1$
$x_2\ y_2\ z_2\ sx_2\ sy_2\ cl2\ cb_{21}\ cb_{22} \ldots cb_{2,Natt}\ val_{21}\ val_{22} \ldots val_{2,Natt}\ qe_2$
....
$\boldsymbol{x_{Ndat}\ y_{Ndat}\ z_{Ndat}\ sx_{Ndat}\ sy_{Ndat}\ cl_{Ndat}\ cb_{Ndat,1}\ cb_{Ndat,2} \ldots cb_{Ndat,Natt}\ val_{Ndat,1}\ val_{Ndat,2} \ldots val_{Ndat,Natt}}$

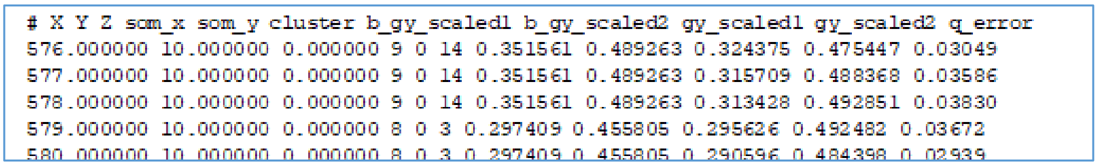
The geospace output file is a space separated text file.

Header row indicates the meaning of each column (separator = space):

- X, Y, Z = optionally there can be one of more spatial coordinates
- som_x, som_y = SOM coordinates
- cluster = SOM codebook vector cluster (from the cluster() or clusters() function)
- b_attri = Best matching codebook vector
- attri = input data data
- qerror = quantization error for the data point

Afterwards follow output for each xj, yj, (zj) combination, one point per line with columns according to the header. Column separator is space.

Example:



```
# X Y Z som_x som_y cluster b_gy_scaled1 b_gy_scaled2 gy_scaled1 gy_scaled2 q_error
576.000000 10.000000 0.000000 9 0 14 0.351561 0.489263 0.324375 0.475447 0.03049
577.000000 10.000000 0.000000 9 0 14 0.351561 0.489263 0.315709 0.488368 0.03586
578.000000 10.000000 0.000000 9 0 14 0.351561 0.489263 0.313428 0.492851 0.03830
579.000000 10.000000 0.000000 8 0 3 0.297409 0.455805 0.295626 0.492482 0.03672
580.000000 10.000000 0.000000 8 0 3 0.297409 0.455805 0.290596 0.484398 0.02939
```

Figure 4. ***Exemplary detail of geospace.txt.***

Structure of somspace.txt:

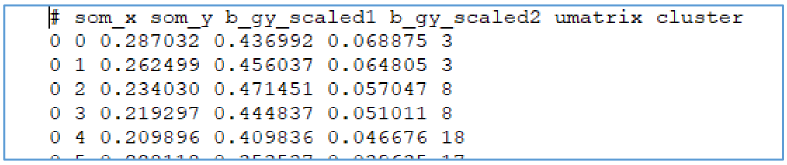
som_x som_y b_attr$_1$ battr$_2$ .... b_attr$_{Natt}$ umatrix cluster
$sx_1\ sy_1\ cb_{11}\ cb_{12} \ldots cb_{1,Natt}\ um_1\ cl_1$
$sx_2\ sy_2\ cb_{21}\ cb_{22} \ldots cb_{2,Natt}\ um_2\ cl_2$
....
$\boldsymbol{sx_{Ndat}\ sy_{Ndat}\ cb_{Ndat,1}\ cb_{Ndat,2} \ldots cb_{Ndat,Natt}\ um_{Ndat}\ cl_{Ndat}}$

The SOM space output file is a space separated text file.

Header row indicates the meaning of each column (separator = space):

- som_x, som_y = SOM coordinates
- b_attr$_i$ = Best matching codebook vector
- umatrix = Umatrix value
- cluster = SOM codebook vector cluster (from the cluster() or clusters() function)

Afterwards follow the output for each sx_j, sy_j pair, one point per line with columns according to the header. Column separator is space.



```
# som_x som_y b_gy_scaled1 b_gy_scaled2 umatrix cluster
0 0 0.287032 0.436992 0.068875 3
0 1 0.262499 0.456037 0.064805 3
0 2 0.234030 0.471451 0.057047 8
0 3 0.219297 0.444837 0.051011 8
0 4 0.209896 0.409836 0.046676 18
```

Figure 5. Exemplary detail of somspace.txt.

3.6. Raster data processing

Purpose:

- Based on the SOM output files (somspace.txt and geospace.txt), the cluster data as well as the parameter maps are converted to ESRI Grids.

Input:

- somspace.txt
- geospace.txt
- mask (GRID)
- number of cells in X and Y direction

Output:

- Geospace
    - Geo-cluster (GRID)
    - Quantization error (GRID)
    - Input raster data (GRID)
- Somspace
    - SOM-cluster (GRID)
    - U-matrix (GRID)
    - Input raster data (GRID)

3.7. Postprocessing: Loading results to ArcMap

After successful calculation results will be loaded to ArcMap. Predefined layer's colour and order in the TOC are provided by referencing two layers (ColorSource.lyr, EmptyLayer.lyr) included in the toolbox directory.

3.8. Optional deleting of temporary files

Temporary data can be deleted by enabling the intended checkbox before running the tool. These are SOM.lrn, SOM.xml, geospace.txt, somspace.txt and the mask.

4. TESTING REPORT

nextsomcore performance in computation speed was tested. The *nextsomcore* was applied with different parameter values to check computation times and the influence of the parameters to the calculation time (Table Table 12).

Table 2. Processing time by nextsomcore.

Number of MID	Dimension of MID	Processing time by *nextsomcore* in minutes	X and Y Dimension to generate SOM	Number of epochs to run
3	700 x 1570	2	10 x 10	10
3	700 x 1570	27	72 x 72	10
3	700 x 1570	30	100 x 100	10
5	700 x 1570	26	72 x 72	10
10	700 x 1570	37	72 x 72	10
3	700 x 1570	15	72 x 72	5
5	700 x 1570	16	72 x 72	5
10	700 x 1570	17	72 x 72	5

The SOM clustering toolbox performance in computation speed was tested as well. Different parameter values were performed to reveal computation times and the influence of the parameters to the calculation time (Table Table 13).

Table 3. Processing time by SOM Clustering Toolbox.

Number of MID	Dimension of MID	Processing time by *nextsomcore* in minutes	X and Y Dimension to generate SOM	Number of epochs to run	K-means number of initial means
2	700 x 1570	11	10 x 10	5	5
2	700 x 1570	12	10 x 10	5	10
2	700 x 1570	11	10 x 10	10	5
3	700 x 1570	12	10 x 10	5	5
4	700 x 1570	14	10 x 10	5	5

5. REFERENCES

Kohonen T., 2001. Self-organizing maps, Third Extended Edition, *Springer Series in Information Sciences*, 30.

Wittek P., Gao S. C., Lim I. S., and Zhao L., 2013. Somoclu: An efficient parallel library for selforganizing maps. arXiv preprint arXiv:1305.1422.