



**NEXT**

New Exploration Technologies

## DELIVERABLE 4.11

### Open-source stand-alone SOM software

Horizon 2020 Project: **NEXT**

Author(s): **Johanna Torppa**

Institution: **Geological Survey of Finland**

Date: **30.04.2019**

#### *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.*

Deliverable administration			
No & name	<b>D 4.11 Open-source stand-alone SOM software</b>		
Status	Final	Due	M12
		Date	30.04.2019
Author(s)	Johanna Torppa		
Dissemination level	Open Research Data Pilot		
Description of the related task and the deliverable.	<b>Task 4.4 Development of self-organizing maps software for analysing geospatial data. Deliverable provides the system concept, class diagram, software design, testing report and user's manual of the self-organizing maps and k-means clustering software developed in NEXT specifically for analysing geospatial data.</b>		
Participants	Geological Survey of Finland, Beak Consultants GmbH		
Comments			
V	Date	Authors	Description
0.1	24.08.2018	J. Torppa	Version for Progress Meeting 1
0.2	20.03.2019	J. Torppa	Version for Progress Meeting 2
1.0	30.04.2019	J. Torppa	First release

## About NEXT

NEXT consortium consists of 16 partners from leading research institutes (3), academia (3), service providers (5) and industry (5). The members come from 6 EU member states (FI, FR, DE, MT, ES and SE) and represent the main metal producing regions of Europe, Fennoscandian Shield, Variscan Belt of Iberia and Central European Belt. These economically most important metallogenic belts of the EU have diverse geology with evident potential for different types of new mineral resource. The mineral deposits in these belts are the most feasible sources of critical, high-tech and other economically important metals in the EU. The project consortium has also a vast international collaboration network, e.g. 50% of the Advisory Board members have been invited from outside EU.

In addition to the variable geology, the vulnerability of the environment and the glacial sedimentary cover in the Arctic regions of northern Europe, and the thick weathering crust and more densely populated nature of the target areas in the Iberian and Central European belts influence the mineral exploration in different ways. New environmentally sound exploration concepts and technologies will be optimized and tested on diverse mineral deposit types.

NEXT will develop new geomodels, novel sensitive exploration technologies and data analysis methods which together are fast, cost-effective, environmentally safe and socially accepted. Methods developed reduce the current high exploration costs and enhance participation of civil society from the start of exploration, raising awareness and trust. Moreover, the reduced environmental impact of the new technologies and better knowledge about the factors influencing social licensing will help promote social acceptance of both exploration and mining and therefore support the further development of Europe's extractive industry.

## TABLE OF CONTENTS

1	Introduction .....	4
2	System Concept.....	4
3	Description of Deliverable.....	5
4	References.....	5

## LIST OF FIGURES

Figure 1.	Structure of the self-organizing maps software developed in the NEXT project. Blue colour refers to existing software, while green, orange and black components were implemented in NEXT.....	4
-----------	---	---

## LIST OF APPENDICES

- Appendix 1: Technical Specification – nextsomcore
- Appendix 2: Technical Specification – GisSOM
- Appendix 3: User manual - GisSOM

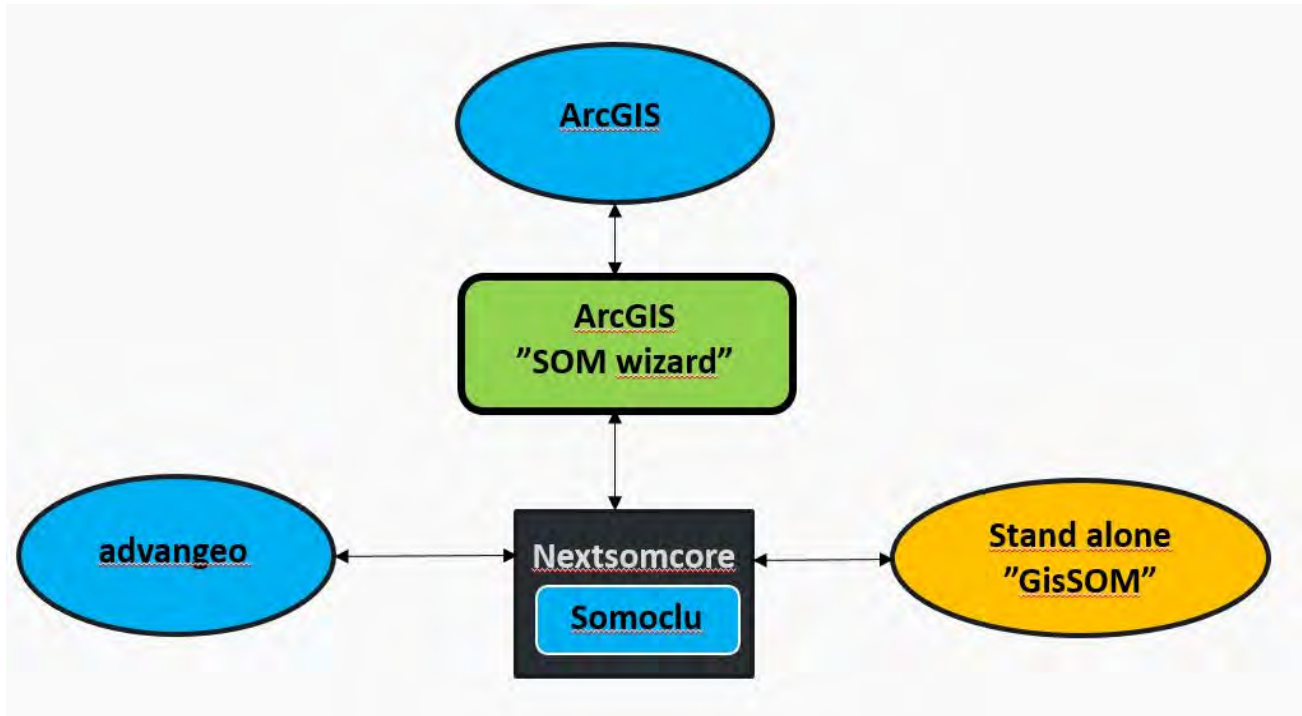
## 1 INTRODUCTION

The purpose of this document is to provide the system concept, software design, class diagram, testing report and user's manual of the open-source stand-alone self-organizing maps (SOM) data analysis software that was developed in the European Union funded H2020 project NEXT.

The following people from the Geological Survey of Finland have contributed to the software development (in alphabetical order): Sakari Hautala, Janne Kallunki, Jaakko Madetoja and Johanna Torppa. The following people from Beak Consultants GmbH have been involved in planning and testing the software (in alphabetical order): Sven Etzold, Peggy Hielscher and Andreas Knobloch.

## 2 SYSTEM CONCEPT

Integrating information of a number of different geoscientific datasets is constantly carried out in geoscientific research and in other research fields making use of spatial distributions of various quantities. Self-organizing maps is a powerful method for integration and visualization of large data sets but, up to date, there has been no properly maintained software to carry out the computations, to visualize the results in geospace and to connect the SOM space to the geospace. CSIRO's commercial Matlab-based SiroSOM software has been useful but, due to the lack of maintainance and documentation, it was considered necessary to implement new software with improved functionality as well as a proper user manual and technical specification documents.



**Figure 1.** *Structure of the self-organizing maps software developed in the NEXT project. Blue colour refers to existing software, while green, orange and black components were implemented in NEXT.*

The software developed in NEXT consists of several components (Figure 1). The open-source stand-alone software, described in this document, consists of the *nextsomcore* (D 4.11 Appendix 1) and the *GisSOM* (D 4.11 Appendices 2 and 3) components. *nextsomcore* performs the SOM and k-means computations using an external SOM computation package Somoclu (Wittek et al., 2013), and is capable of being integrated to other software. *GisSOM* is the graphical user interface that is used for selecting and pre-processing data used by *nextsomcore*, and for post-processing and visualizing the *nextsomcore* results. Interfaces between *nextsomcore* and *advangeo*® (*advangeo* SOM wizard, D 4.12, in development, to be delivered in M18) and between *nextsomcore* and ArcGIS (SOM wizard, D 4.13, in development, to be delivered in M18) are also implemented in NEXT.

### 3 DESCRIPTION OF DELIVERABLE

The software consists of two components (*nextsomcore* and *GisSOM*), each of which are described in a separate document provided as an appendix to this document. In addition, the user's manual is provided as an appendix.

#### *D 4.11 Appendix 1: Technical Specification - nextsomcore*

The document provides technical description such as the software design, class diagram and testing report of *nextsomcore*.

#### *D 4.11 Appendix 2: Technical Specification - GisSOM*

The document provides technical description such as the software design, class diagram and testing report of *GisSOM*.

#### *D 4.11 Appendix 3: User's manual - GisSOM*

The document is the user manual for *GisSOM*.

### 4 REFERENCES

Deliverable 4.12: SOM tool for *advangeo*® (under preparation, due in M18)

Deliverable 4.13: SOM tool for ArcGIS (under preparation, due in M18)

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



**NEXT**

New Exploration Technologies

## DELIVERABLE 4.11

### Appendix 1

## Technical Specification - *nextsomcore*

Horizon 2020 Project: **NEXT**

Author(s): **Johanna Torppa**

Institution: **Geological Survey of Finland**

Date: **30.04.2019**

#### *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

1	Introduction .....	4
2	Self-organizing maps and k-means methods .....	4
3	Software design, architecture and class diagram .....	5
4	Description of nextsomcore.py functions .....	7
4.1	load_data() .....	7
4.1.1	Input file formats .....	8
4.1.1.1	LRN .....	8
4.1.1.2	CSV (TBA) .....	8
4.1.1.3	geoTIFF (TBA) .....	9
4.2	train() .....	9
4.3	cluster()/clusters() .....	10
4.4	save_geospace_result()/save_somspace_result() .....	11
4.4.1	save_geospace_result() output file .....	11
4.4.2	save_somspace_result() output file .....	12
5	Testing report .....	12
6	References .....	13

## LIST OF FIGURES

Figure 1.	Structure of the self-organizing maps software developed in the NEXT project. Blue colour refers to existing software, while green, orange and black components were implemented in NEXT .....	4
Figure 2.	Class diagram of nextsomcore. ....	6

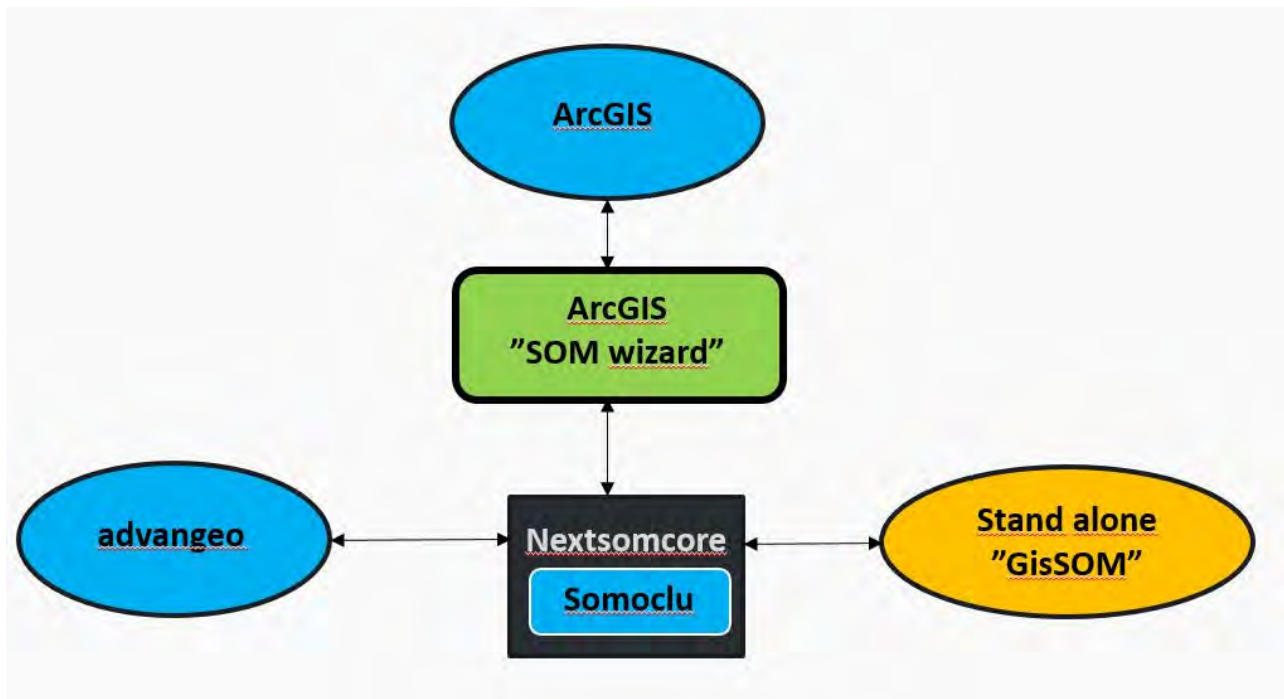
## LIST OF TABLES

Table 1.	Input to the load_data() function.....	7
Table 2.	Output from the load_data() function. ....	7
Table 3.	Contents of the LRN formatted data input file. ....	8
Table 4.	Input to the train() function. ....	9
Table 5.	Output from the train() function.....	10
Table 6.	Input to the cluster()/clusters() function. ....	11
Table 7.	Input to the save_geospace_result() and save_somspace_result() functions.....	11
Table 8.	Contents of the geospace output file.....	12
Table 9.	Contents of the SOM space output file. ....	12
Table 10.	Processing time by nextsomcore. ....	13



## 1 INTRODUCTION

The purpose of this document is to describe the software design, class diagram and the testing procedure of *nextsomcore* that is one component of the software implemented in the European Union funded H2020 project NEXT. The software applies self-organizing maps (SOM) and k-means clustering for analyzing geospatial data, and can be utilized using three different graphical user interfaces: ArcGIS, advangeo® and a freeware user interface *GisSOM*.



**Figure 1.** Structure of the self-organizing maps software developed in the NEXT project. Blue colour refers to existing software, while green, orange and black components were implemented in NEXT.

The software components developed in NEXT are shown in Figure 1, and *nextsomcore* is the one that performs the SOM and k-means computations. *nextsomcore* is open source freeware, and capable of being integrated into other software. In addition to *nextsomcore*, an open-source freeware user interface (*GisSOM*, D 4.11 Appendices 2 and 3) is built to pre-process the data as well as to post-process the SOM and k-means results. Interfaces between *nextsomcore* and *advangeo*® (D 4.12, in development, due in M18) and between *nextsomcore* and ArcGIS (D 4.13, in development, due in M18) are also implemented in NEXT.

## 2 SELF-ORGANIZING MAPS AND K-MEANS METHODS

SOM is an unsupervised artificial neural network that projects a set of n-dimensional vectors, that we here call *data vectors*, to a usually 1-3 dimensional SOM lattice (Kohonen, 2001). The lattice

consists of cells, each of which is represented by an n-dimensional vector that we call a *codebook vector*. Each data vector is assigned to the SOM cell, whose codebook vector is closest to the data vector itself. The SOM cell that a data vector is assigned to, is called the Best Matching Unit (BMU) for that data vector. The usability of SOM comes from its topology preserving nature: similar data vectors are assigned to SOM cells that are close together. This derives from the fact that each time a data vector is assigned to the BMU, not only the BMU codebook vector is changed to better represent the assigned data vector, but also the codebook vectors of its neighbouring cells are changed similarly. The quality of a SOM is divided between data representation accuracy, i.e. how well the SOM describes the original data (quantization error), and topological accuracy, i.e. how well the SOM preserves topology (topological error).

As SOM assigns several data vectors in a single representative SOM cell, it can be considered as a clustering method itself, if each SOM cell is considered as a cluster of the data vectors that are assigned to it. However, as the idea in SOM is to generate a smoothly varying map of codebook vectors, the number of cells in a SOM should be large (one rule of thumb is  $5 * \sqrt{\text{number\_of\_data\_points}}$ ), and the number of clusters is big after one SOM computation round. The number of clusters can be reduced by either hierarchically running SOM again using the codebook vectors as data vectors, or by applying another clustering approach (e.g. k-means) to SOM codebook vectors. As SOM preserves topology, clustering can also be done by visually studying the U-matrix, which is the difference between one SOM cell and its neighbours. By visualizing different codebook vector values, SOM can be used for data characterization and searching for correlation between different input variables.

K-means clustering is a very basic clustering method where each data point is assigned to the cluster that best represents the data point. The representative cluster value is computed as the mean or median of all the data points assigned to it. In k-means, cluster values are not affected by one another, and they are not spatially arranged according to similarity as in SOM. Although one round of k-means computations is faster than one round of SOM, k-means has to be run many times with different initializations for varying numbers of clusters, making it slow for large data sets.

### 3 SOFTWARE DESIGN, ARCHITECTURE AND CLASS DIAGRAM

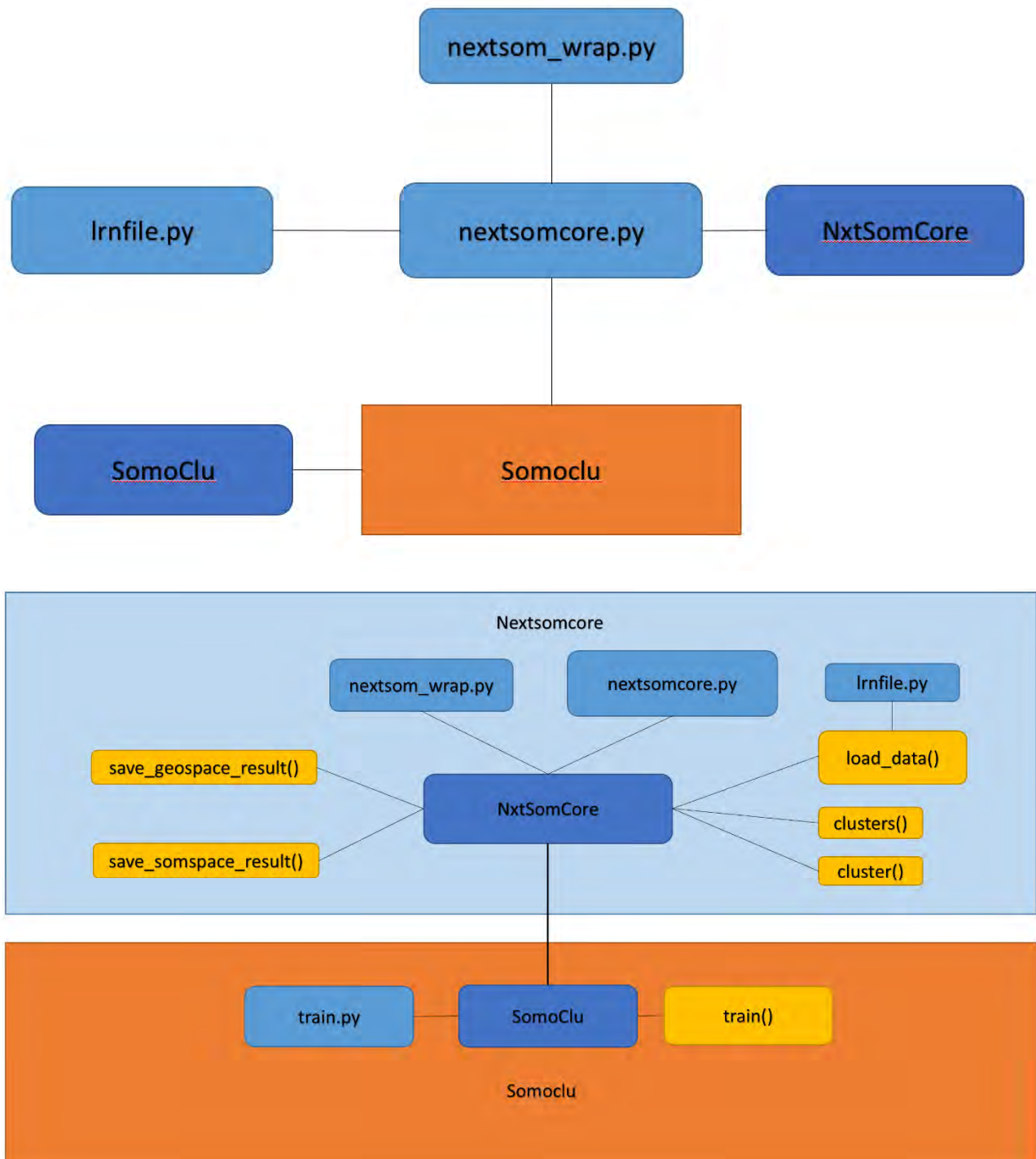
nextsomcore is implemented in Python. The package Somoclu is implemented in C++ with an interface to Python.

The following Python packages/libraries are required:

- somoclu
- matplotlib
- numpy
- sklearn
- scipy
- seaborn
- pandas

nextsomcore package is used by the nextsom\_warp.py wrapper script which creates an instance of NxtSomCore class, the definition of which is contained in the nextsomcore.py -file. NxtSomCore uses

lrnfile.py to read the input LRN file, and train.py from Somoclu to create an instance of the SomoClu –class, and to train the Self Organizing Map using the ‘train()’-method from train.py. NextSomCore uses the clusters() or cluster() –method to run k-means clustering, and uses the methods save\_somspace\_result() and save\_geospace\_result() to save the SOM output into text files. Somoclu is an external package written by Peter Wittek (Wittek et al., 2017).



**Figure 2.** The main packages and classes of nextsomcore.

## 4 DESCRIPTION OF NEXTSOMCORE.PY FUNCTIONS

nextsomcore is a Python package that is used for computing SOM and k-means clustering for a given data set and a set of computation parameters. In addition to the SOM training function (`train()`, Sec 4.2) and the k-means computation function (`cluster()/clusters()`, Sec 4.3) for clustering the SOM codebook vectors, nextsomcore also contains functions for reading in data (`load_data()`, Sec 4.1) and writing out results (`save_geospace_result()/save_somspace_result()`, Sec 4.4).

### 4.1 load\_data()

Currently, input data can only be imported in the LRN format (Sec 4.1.1.1). Other formats to be added (TBA, Sec 0 and 4.1.1.3). The `load_data()` function reads the input data file(s) (uses `lrnfile.py`) and returns the data as a Python numpy array.

Function call:

```
>>load_data(datafile)
```

**Table 1.** Input to the `load_data()` function.

Name	Type	Dim	Description
<b><i>datafile</i></b>	string	1 or $N_{att}$	Input data file(s) containing the $x,y$ (and $z$ ) coordinates and the $N_{att}$ evidence attributes for the $N_{dat}$ data points. If input is in grid format, there is a separate file for each evidence attribute. If input is a text file, there is only one input containing all the evidence attributes.

**Table 2.** Output from the `load_data()` function.

Name/Type	Keys	Type	Dim	Description
<i>LoadData/</i> Dictionary	file	string	1 or $N_{att}$	Input data files = <b><i>datafile</i></b>
	rows	int	1	Number of data points = $N_{dat}$
	cols	int	1	Number of columns in data file = $N_{col}$
	coltypes	int	$N_{col}$	Is the column in the original input file labeled as included data (0=excluded, 1=included). $N_{col}$ = number of columns in the input data file
	colnames	str	$N_{col}$	Names of columns in the input data file
	headerlength	int	1	Number of header rows in input file
	data	float	$N_{dat}, N_{att}$	Input data matrix containing the evidence attributes for the $N_{dat}$ data points (only the attributes that are used for SOM computation)
	filetype	str	1	"lrn" (Others TBA)

### 4.1.1 Input file formats

#### 4.1.1.1 LRN

LRN is a text file format, where the data table columns that are to be used for SOM and k-means computation are defined. The file can, thus, contain extra columns that are not used in SOM and k-means computation.

```
# Comment line
%  $N_{dat}$ 
%  $N_{col}$ 
 $a_1 a_2 \dots a_{N_{col}}$ 
id x y z attr1 attr2 ... attrNattr
1  $x_1 y_1 z_1 val_{11} val_{12} \dots val_{1,N_{attr}}$ 
2  $x_2 y_2 z_2 val_{21} val_{22} \dots val_{2,N_{attr}}$ 
....
 $N_{dat} x_{N_{dat}} y_{N_{dat}} z_{N_{dat}} val_{N_{dat},1} val_{N_{dat},2} \dots val_{N_{dat},N_{attr}}$ 
```

**Table 3.** Contents of the LRN formatted data input file.

Lines 1->N	Optionally, N comment lines starting with a hash sign.
Line N+1	$N_{dat}$ = Number of data points, i.e., number of rows after the header line. The line begins with a percent sign.
Line N+2	$N_{col}$ = 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.
Line N+3	$a_i$ = 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.
Line N+4	Header row indicates the meaning of each column, i.e., data attribute, on the preceding rows containing the data points (separator = tab): id = unique index of data points. This column is required as the first column. x,y,z = optionally there can be one or more spatial coordinates. The name of the corresponding columns must be defined as "x", "y" or "z". attr <sub>i</sub> = data attributes
Lines N+5 ->N+5+N <sub>dat</sub>	Data points according to the header. One data point per line. Column separator = tab.

#### 4.1.1.2 CSV (TBA)

If input is given as a CSV file (described below), only the attributes that will be used in SOM and k-means computation should be included.

```
x,y,z,attr1,attr2,...,attrNattr
 $x_1,y_1,z_1,val_{11},val_{12},\dots,val_{1,N_{attr}}$ 
 $x_2,y_2,z_2,val_{21},val_{22},\dots,val_{2,N_{attr}}$ 
....
 $x_{N_{dat}},y_{N_{dat}},z_{N_{dat}},val_{N_{dat},1},val_{N_{dat},2},\dots,$ 
 $val_{N_{dat},N_{attr}}$ 
```

**Table 4.** Contents of the CSV formatted data input file.

Line 1	Header row indicates the meaning of each column, i.e., data attribute, on the preceding rows containing the data points (separator=comma): x,y,z = optionally there can be one of more spatial coordinates. attr <sub>i</sub> = data attributes
Lines N+1 ->N+1+Ndat	Data points according to the header. One data point per line. Column separator=comma.

#### 4.1.1.3 geoTIFF (TBA)

GeoTIFF is a metadata standard for providing georeferencing information within a TIFF file.

## 4.2 train()

The SOM algorithm is calculated using the Somoclu library (Wittek et. al, 2017) written in C++ with Python, R and MATLAB interfaces. It has multicore capabilities and reduced memory usage, both of which are important as SOM algorithm can take a long time and use a lot of memory when the amount of data points and the size of the SOM increases. The tool also enables the use of graphics processing units (GPUs), but GPU-support is currently untested and nextsomcore defaults to CPU. Somoclu also includes multiple parameters that can be set for the SOM algorithm (Table 4). For more details on how the SOM algorithm, parallelization and workload structure works in Somoclu, see Wittek et al. (2017).

Function call:

```
>>train(dat,SD[1],SD[2],Ni,*kerneltype=ktype, *verbose=v, *neighbourhood=Snh,
*std_coeff=Nc, *radius0=Rnh[1], * radiusN=Rnh[2]=, *radiuscooling=Rcool,
*scale0=Lrate[1], * scaleN=Lrate[2]=, *scalecooling=Lcool)
```

\*=Optional parameter

**Table 4.** Input to the train() function.

Name	Type	Dim	Description	Default value
<b>dat</b>	float	[N <sub>dat</sub> ,N <sub>att</sub> +2]	Input data matrix containing the x,y,z coordinates and N <sub>att</sub> evidence attributes for the N <sub>dat</sub> data points (only the attributes that are used for SOM computation)	LoadData['data']
<b>S<sub>D</sub></b>	int	2	Dimensions of the 2D SOM map.	[sqrt(5*sqrt(N <sub>dat</sub> )),sqrt(5*sqrt(N <sub>dat</sub> ))] (used, e.g., by Vesanto and Alhoniemi (2000))
<b>N<sub>i</sub></b>	int	1	Number of iterations	10
<b>S<sub>G</sub></b>	str	1	Type of SOM grid (hexa (TBA), square).	square
<b>S<sub>s</sub></b>	str	1	Type of SOM (sheet, toroid)	toroid

$S_{nh}$	str	1	Shape of the neighborhood function (Gaussian, bubble)	Gaussian
$R_{nh}$	int	2	Initial and final size of the neighborhood	$[\min(S_D[1], S_D[2])/2, 1]$
$R_{cool}$	str	1	Function that defines the decrease in the neighborhood size as the training proceeds (linear, exponential)	linear
$L_{rate}$	float	2	Initial and final learning rate	$[0.1, 0.001]$
$L_{cool}$	str	1	Function that defines the decrease in the learning scale as the training proceeds (linear, exponential)	linear
$ini$	str	1	Type of SOM initialization (random, pca)	random
$init\_cb$	float	$[S_D[1]*S_D[2], N_{att}]$	Initial codebook vectors	
$N_c$	float	1	Coefficient in the Gaussian neighborhood function ( $\exp(-  x-y  ^2/(2*(coeff*radius)^2))$ )	0.5
$k_{type}$	int	0	Kernel type. 0=dense CPU kernel, 1=dense GPU kernel (if compiled using GPU)	0
$v$	int	1	Level of verbosity. 0, 1 or 2. Needed?	2

**Table 5. Output from the train() function.**

Name/Type	Key	Type	Dim	Description
som/ Dictionary	<b>codebook</b>	float	$[S_D[1]*S_D[2], N_{att}]$	SOM codebook vectors
	<b>bmus</b>	int	$[N_{Dat}, 2]$	Best matching units ( $som\_x, som\_y$ ) on SOM for each data point
	<b>umatrix</b>	float	$[[S_D[1], S_D[2]]]$	The SOM U-matrix
	n_columns	int	1	Number of columns in SOM
	n_rows	int	1	Number of rows in SOM
	n_dim	int	1	Number of attributes ( $=N_{att}$ )

### 4.3 cluster()/clusters()

k-means clustering function from the external Python library sklearn is used for clustering the SOM codebook vectors. The cluster() function runs k-means for a given number of clusters ( $N_c$ ), while the clusters() function runs clustering for a defined range of clusters  $[N_{cr}[1], N_{cr}[2]]$  and a defined number of random initializations, and returns the best clustering result based on the Davies-Boulding index.

Function call:

```
>>cluster(som, Nc)
>>clusters(som, Ncr[1], Ncr[2], Ni)
```

**Table 6.** *Input to the cluster()/clusters() function.*

Name	Type	Dim	Description / Note
<i>som</i>	Dictionary	-	train() function's output dictionary (Sec 4.2)
<i>N<sub>c</sub></i>	int	1	Number of clusters
<i>N<sub>cr</sub></i>	int	2	Low and high ends of the range of the number of clusters
<i>N<sub>i</sub></i>	int	1	Number of random initializations

**Table 7.** *Output from the cluster()/clusters() function.*

Name	Type	Dim	Description / Note
<i>cl</i>	float	[ <i>S<sub>D</sub></i> [1], <i>S<sub>D</sub></i> [2]]	Cluster indices for each SOM cell

## 4.4 save\_geospace\_result()/save\_somspace\_result()

Results are written into two separate text files: one for geospace and one for SOM space.

Function call:

```
>>save_geospace_result(outf,LoadData,som)
>>save_somspace_result(outf,LoadData,som)
```

**Table 7.** *Input to the save\_geospace\_result() and save\_somspace\_result() functions.*

Name	Type	Dim	Description / Note
<i>outf</i>	str	1	Name of the geospace output file
<i>LoadData</i>	Dictionary	-	load_data() function's output dictionary (Sec 4.1)
<i>som</i>	Dictionary	-	train() function's output dictionary (Sec 4.2)

### 4.4.1 save\_geospace\_result() output file

The geospace output file is a space separated text file, organized as follows:

```
X Y (Z) som_x som_y cluster b_attr1 battr2 ... b_attrNatt attr1 attr2 ... attrNatt qerror
x1 y1 z1 sx1 sy1 cl1 cb11 cb12 ... cb1,Natt val11 val12 ... val1,Natt qe1
x2 y2 z2 sx2 sy2 cl2 cb21 cb22 ... cb2,Natt val21 val22 ... val2,Natt qe2
....
xNdat yNdat zNdat sxNdat syNdat clNdat cbNdat,1 cbNdat,2 ... cbNdat,Natt valNdat,1 valNdat,2 ... valNdat,Natt
```



**Table 8. Contents of the geospace output file.**

Line 1	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_attr <sub>i</sub> = Best matching codebook vector attr <sub>i</sub> = input data data qerror = quantization error for the data point
Lines 2 ->1+N <sub>dat</sub>	Output for each x <sub>j</sub> ,y <sub>j</sub> (z <sub>j</sub> ) combination; one point per line. Columns according to the header. Column separator = space.

#### 4.4.2 save\_somspace\_result() output file

The SOM space output file is a space separated text file, organized as follows:

<pre> som_x som_y b_attr<sub>1</sub> battr<sub>2</sub> ... b_attr<sub>Natt</sub> umatrix cluster sx<sub>1</sub> sy<sub>1</sub> cb<sub>11</sub> cb<sub>12</sub> ... cb<sub>1,Natt</sub> um<sub>1</sub> cl<sub>1</sub> sx<sub>2</sub> sy<sub>2</sub> cb<sub>21</sub> cb<sub>22</sub> ... cb<sub>2,Natt</sub> um<sub>2</sub> cl<sub>2</sub> .... sx<sub>Ndat</sub> sy<sub>Ndat</sub> cb<sub>Ndat,1</sub> cb<sub>Ndat,2</sub> ... cb<sub>Ndat,Natt</sub> um<sub>Ndat</sub> cl<sub>Ndat</sub> </pre>
---

**Table 9. Contents of the SOM space output file.**

Line 1	Header row indicates the meaning of each column (separator = space):  som_x, som_y = SOM coordinates  b_attr <sub>i</sub> = Best matching codebook vector  umatrix = Umatrix value  cluster = SOM codebook vector cluster (from the cluster() or clusters() function)
Lines 2 -> 1+S <sub>D</sub> [1]*S <sub>D</sub> [2]	Output for each s <sub>xj</sub> ,s <sub>yj</sub> pair; one point per line. Columns according to the header. Column separator = space.

## 5 TESTING REPORT

For *nextsomore*, a graphical user interface for pre- and postprocessing tasks as well as the start of the SOM calculation was developed and *nextsomore* performance in computation speed was tested.

The *nextsomore* was applied with different parameter values to check computation times and the influence of the parameters to the calculation time (Table 10).

**Table 10.** 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

## 6 REFERENCES

Deliverable 4.11 Appendix 2: Technical Specification – *GisSOM*

Deliverable 4.12: SOM tool for advangeo® (under preparation, due in M18)

Deliverable 4.13: SOM tool for ArcGIS (under preparation, due in M18)

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

Vesanto J. and Alhoniemi E., 2000. Clustering of the self-organizing map, *IEEE Transactions on neural networks*, 11 (3), pp. 586-600.

Wittek P., Gao S. C., Lim I. S., and Zhao L., 2017. Somoclu: An efficient parallel library for self-organizing maps. *arXiv preprint arXiv:1305.1422*.



**NEXT**

New Exploration Technologies

## DELIVERABLE 4.11

### Appendix 2

### Technical Specification: *GisSOM*

Horizon 2020 Project: **NEXT**

Author(s): **Johanna Torppa**

Institution: **Geological Survey of Finland**

Date: **30.04.2019**

#### *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

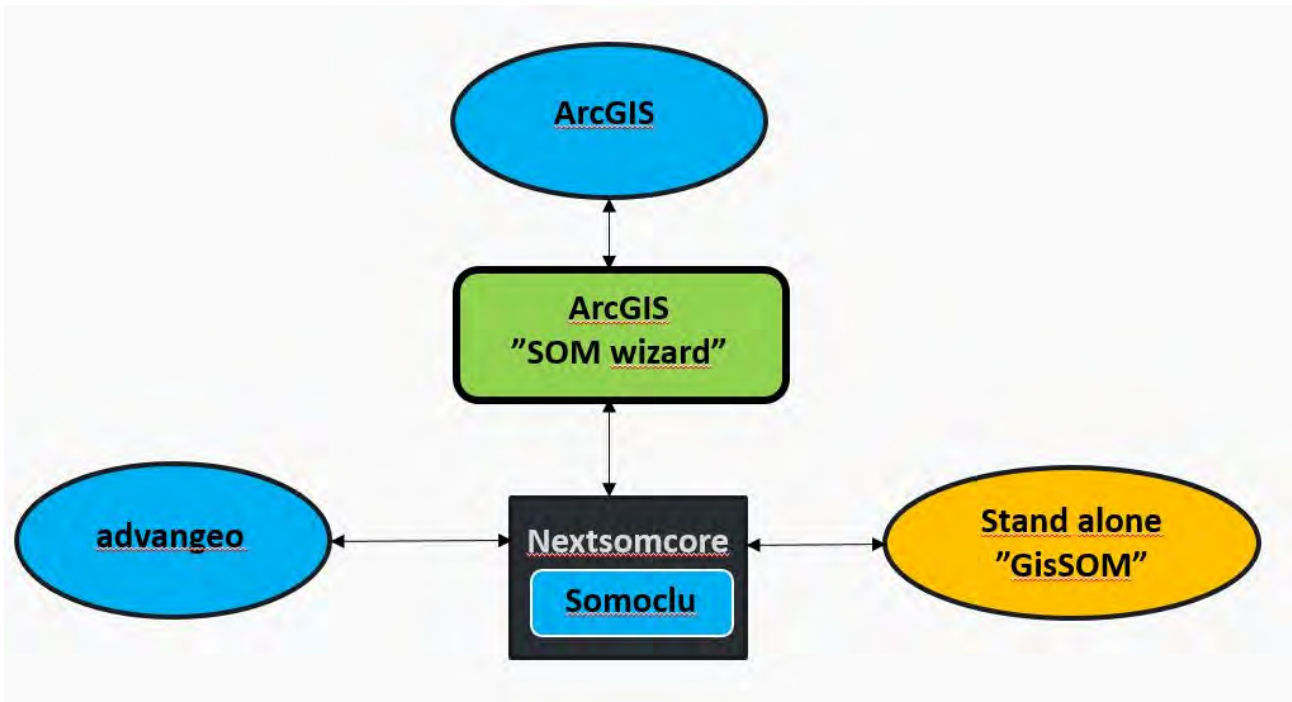
1	Introduction .....	3
2	Software Design and Class Diagram .....	4
3	References.....	6

## LIST OF FIGURES

Figure 1.	Structure of the self-organizing maps software developed in the NEXT project. Blue color refers to existing software, while green, orange and black components were implemented in NEXT.....	3
Figure 2.	Model, View and ViewModel classes. ....	5
Figure 3.	App, MainWindow and smaller service classes.....	5
Figure 4.	Python scripts for computational tasks related to data preparation and plotting. ....	6

# 1 INTRODUCTION

The purpose of this document is to describe the software design, class diagram and the testing procedure of *GisSOM* that is one component of the software implemented in the European Union funded H2020 project NEXT. The software applies self-organizing maps (SOM) and k-means clustering for analyzing geospatial data, and can be utilized using three different graphical user interfaces: ArcGIS, advangeo® and the freeware user interface GisSOM.



**Figure 1.** *Structure of the self-organizing maps software developed in the NEXT project. Blue color refers to existing software, while green, orange and black components were implemented in NEXT.*

The software components developed in NEXT are shown in Figure 1. *nextsomcore* (D 4.11 Appendix 1) is the component that performs the SOM and k-means computations. *GisSOM* is a graphical interface that provides tools for performing data pre-processing, interface to *nextsomcore*, as well as tools for post processing and visualization of the SOM and k-means results. *GisSOM* is open source freeware. In addition to *nextsomcore* and *GisSOM*, interfaces between the *nextsomcore* and *advangeo*® (D 4.12, in development, due in M18) and *ArcGIS* (D 4.13, in development, due in M18) are implemented in NEXT.

## 2 SOFTWARE DESIGN AND CLASS DIAGRAM

GisSOM is implemented in C# using the Windows Presentation Foundation (WPF) framework, according to the the Model-View-ViewModel (MVVM) design model. Computational tasks related to data preprocessing and visualization of the results are implemented as Python scripts.

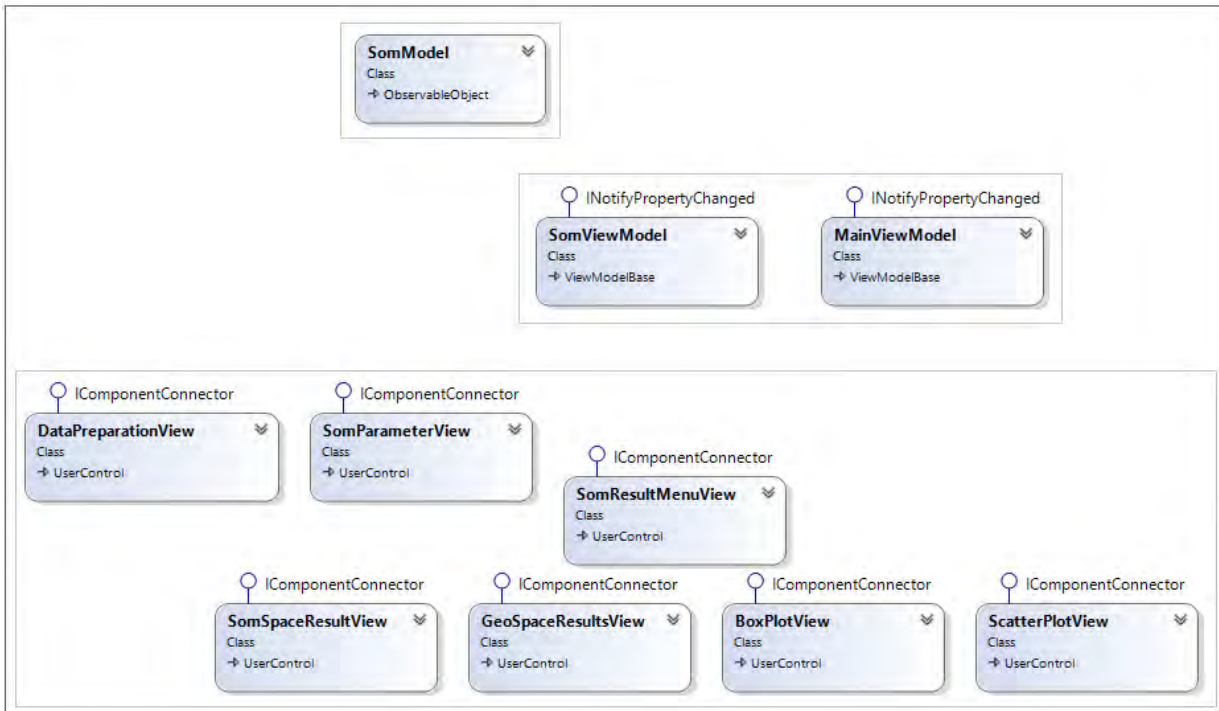
The following Python packages/libraries are required

- nextsomcore
- matplotlib
- numpy
- seaborn
- pandas

Figure 2 presents the Model, View and ViewModel classes. The Model class handles all the data, the View classes handle the user interface and the ViewModel classes act as an interface between these two.

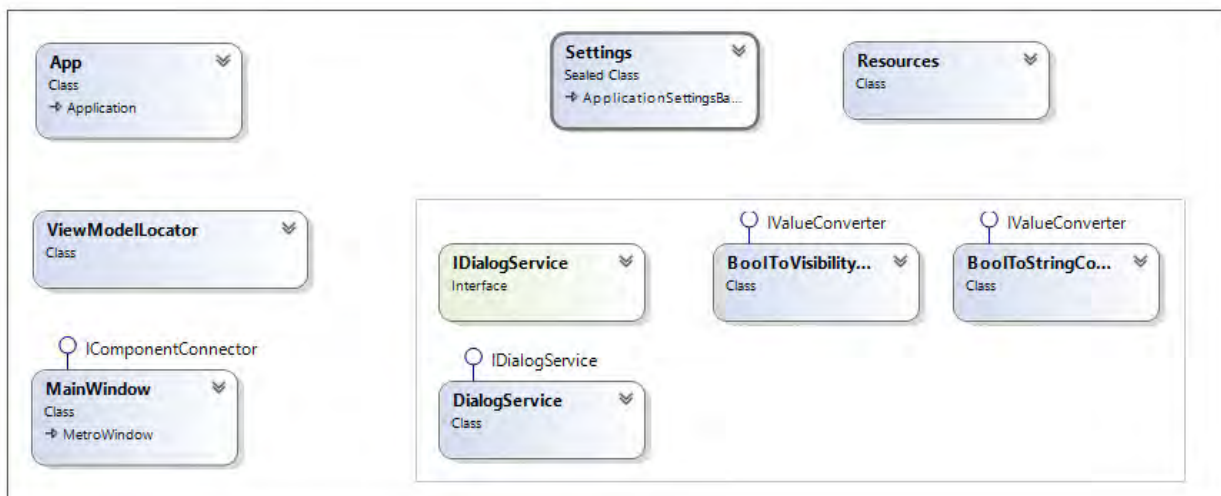
SomModel contains all the parameters that are used in data preprocessing as well as in SOM and k-means computations. It also contains links to the input data files and output files. MainViewModel handles the UI logic and SomViewModel handles all the rest and acts as a link between the View and Model classes.

All Views are user controls that are hosted in the same window (MainWindow, Figure 3). Input data is selected and prepared in the DataPreparationView (log transform, winsorizing, attribute selection). SOM computation parameters are selected in the SomParameterView. Results are shown in SomResultMenuView, which hosts four tabs: SomSpaceResultView (attribute maps and U-matrix in SOM space), GeoSpaceResultView (input data attribute, q-error and k-means cluster maps in geospace), BoxPlotView (boxplots of data distributions within the k-means clusters) and ScatterPlotView (attribute correlation plots in 2D).



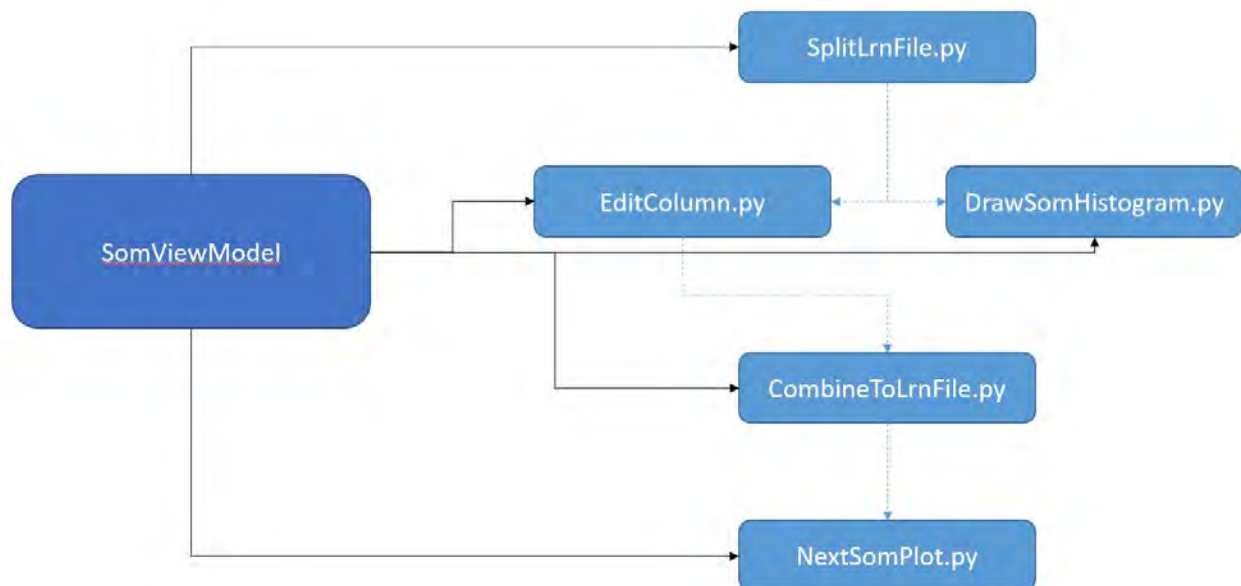
**Figure 2. Model, View and ViewModel classes.**

Figure 3 presents the App, MainWindow and smaller service classes. Class App serves as the launching point for the application. Views use the ViewModelLocator to access the ViewModels. MainWindow is the window where all the visible content is shown. The DialogService, using the IDialogService interface, is used to open file browser dialogs. The ValueConverters are simply value converter service classes. Settings contains general application settings, and Resources external resources.



**Figure 3. App, MainWindow and smaller service classes.**

Figure 4 presents the Python scripts that perform the computational tasks related to data preparation, and visualization of SOM and k-means results. SomViewModel calls all the Python scripts. The workflow and order of execution for the Python scripts is illustrated by the blue lines, but the line is dashed because there is no actual direct connection between the scripts (this is handled by SomViewModel). SplitLrnFile splits the input data file to individual columns that are saved as binary 2D numpy arrays. These individual columns are used by EditDataColumn and DrawSomHistogram scripts. EditDataColumn is used to do the data preparation procedures (winsorize, log transform, etc.), and DrawSomHistogram draws a histogram of the selected data column. After editing the data, CombineToLrnFile script is used to combine the individual columns back to a LRN file. NextSomPlot is used after SOM calculation, to draw the maps, scatterplots and boxplots.



**Figure 4.** Python scripts for computational tasks related to data preparation and plotting.

### 3 REFERENCES

Deliverable 4.11 Appendix 1: Technical Specification – *nextsomcore*

Deliverable 4.12: SOM tool for advangeo® (under preparation, due in M18)

Deliverable 4.13: SOM tool for ArcGIS (under preparation, due in M18)





**NEXT**

New Exploration Technologies

## DELIVERABLE 4.11

### Appendix 3

### User manual - GisSOM

Horizon 2020 Project: **NEXT**

Author(s): **Jaakko Madetoja**

Institution: **Geological Survey of Finland**

Date: **30.04.2019**

#### *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

1	Introduction .....	3
1.1	Self-organizing maps and k-means method .....	3
2	Installation.....	3
2.1	Installation requirements .....	3
3	Using the software .....	3
3.1	Load the data .....	4
3.2	Study the data and perform transformations .....	5
3.3	Choose SOM and k-means parameters .....	6
3.4	Results .....	7
4	References.....	11

## LIST OF FIGURES

Figure 1.	Selecting the data format.....	4
Figure 2.	Studying the data. ....	5
Figure 3.	Choosing parameters for SOM and k-means. ....	6
Figure 4.	SOM space results. ....	7
Figure 5.	Geospace results. ....	8
Figure 6.	Boxplot results.....	9
Figure 7.	Scatterplot results. ....	10

## 1 INTRODUCTION

The purpose of this document is to explain how to install, use and read the results of the *GisSOM* software developed in the European Union funded H2020 project NEXT. Detailed information on the theory and software design is provided in D 4.11 Appendix 2.

*GisSOM* is one component of the software package implemented in NEXT that utilizes self-organizing maps (SOM) and k-means clustering for analyzing geospatial data. The other components of the package are *nextsomcore* (D 4.11 Appendix 1), which performs the SOM and k-means computations, and interfaces between *nextsomcore* and *advangeo*<sup>®</sup> (D4.12, in development, due in M18) as well as *nextsomcore* and ArcGIS (D4.13, in development, due in M18) software.

### 1.1 Self-organizing maps and k-means method

Self-organizing maps (SOM) is an unsupervised artificial neural network that projects a set of n-dimensional vectors to a usually 2 dimensional SOM lattice (Kohonen, 2001). The usability of SOM comes from its topology preserving nature: similar data vectors are assigned to SOM cells that are close together.

Although SOM can be considered as a clustering method itself, the number of clusters is generally too large. Thus, *GisSOM* applies k-means clustering to the results of SOM. K-means clustering is a very basic clustering method where each data point is assigned to the cluster that best represents the data point.

## 2 INSTALLATION

The software comes with an installer. Double-click the installer to start the installation wizard and install the software.

### 2.1 Installation requirements

*GisSOM* requires Windows operating system (7, 8 or 10).

## 3 USING THE SOFTWARE

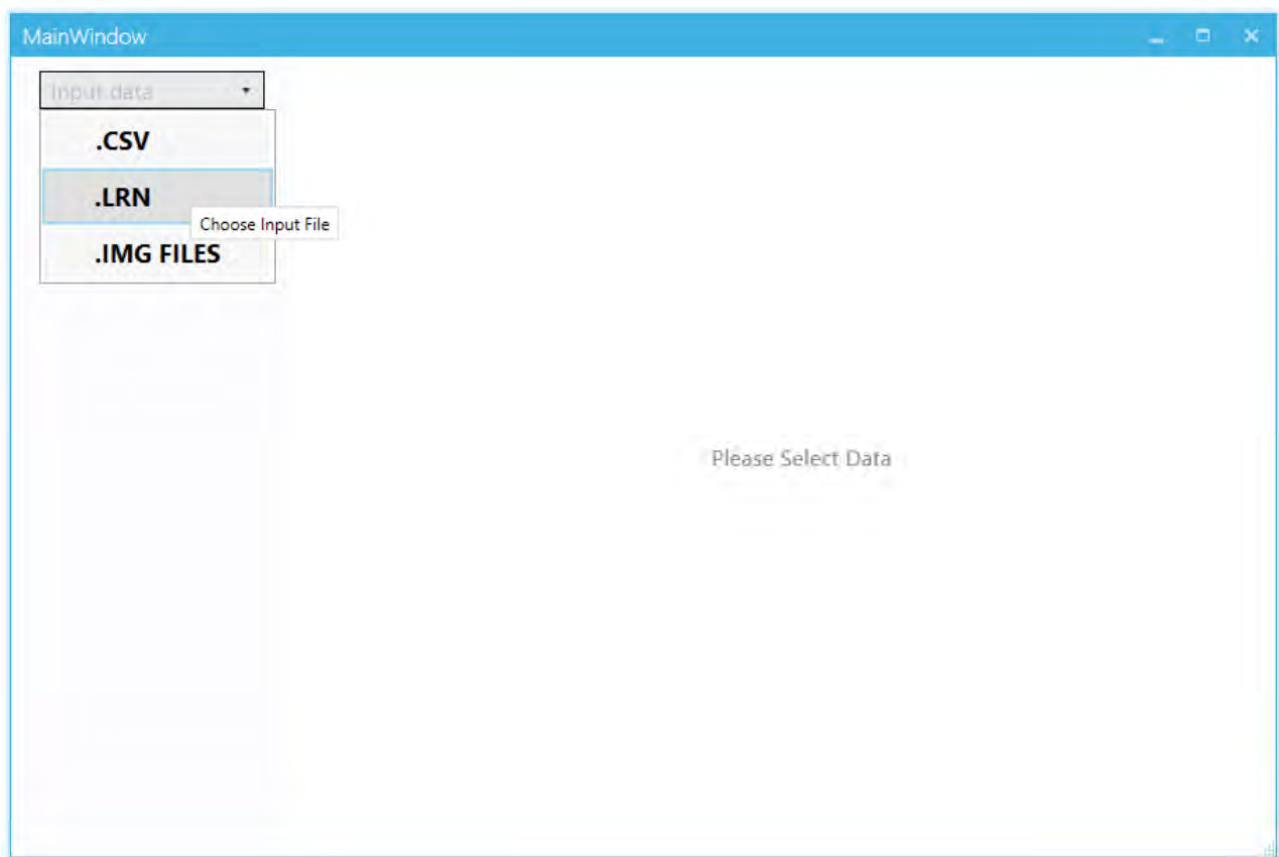
Open the software using *SomUI.exe*, which is in the root of the installation directory. This will open a wizard-style window where you can load and study input data, perform simple transformations, decide SOM and k-means parameters, and study the results in SOM space, geospace, boxplots and scatterplots.

### 3.1 Load the data

In the first step of the wizard, you need to select the input data format and locate the data file from your computer.

Options are

- (currently unavailable) **.CSV**: A comma-delimited text file with comma (,) as the column separator and point as the decimal separator.
- **.LRN**: A text file with tab as the column separator and header lines (see D 4.11 Appendix 1).
- (currently unavailable) **.GeoTIFF FILES**: A raster data format.



**Figure 1.** *Selecting the data format.*

### 3.2 Study the data and perform transformations

In the next step, you need to study the data before it can be used in SOM. You need to select the correct North and East coordinates, exclude any data you don't want to use, and possibly transform data.

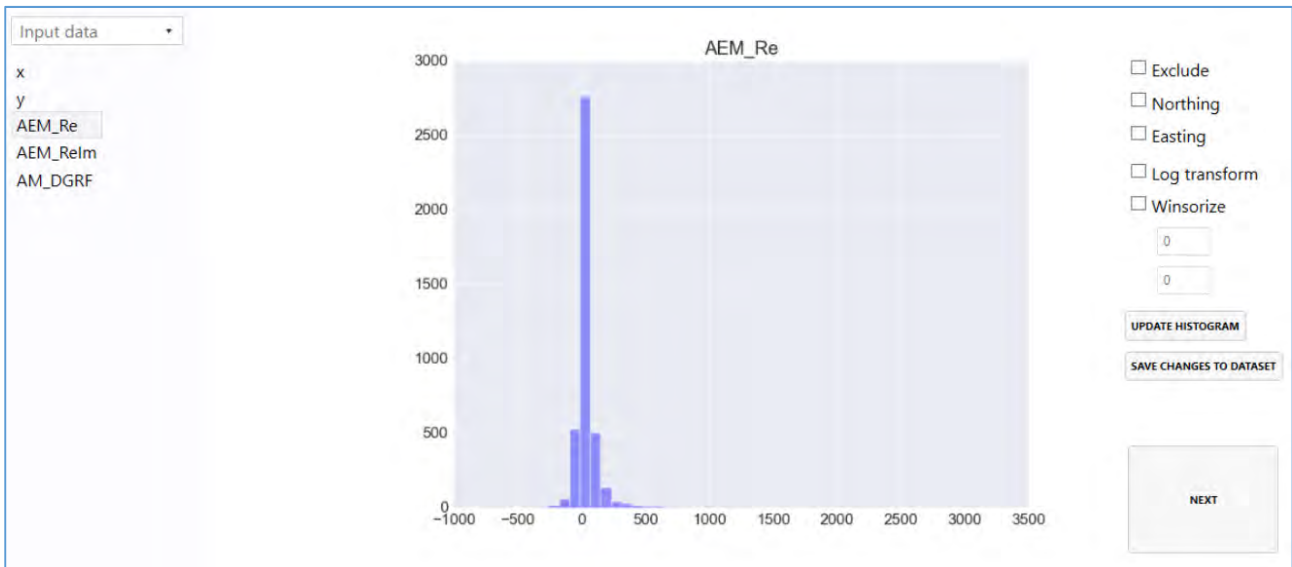


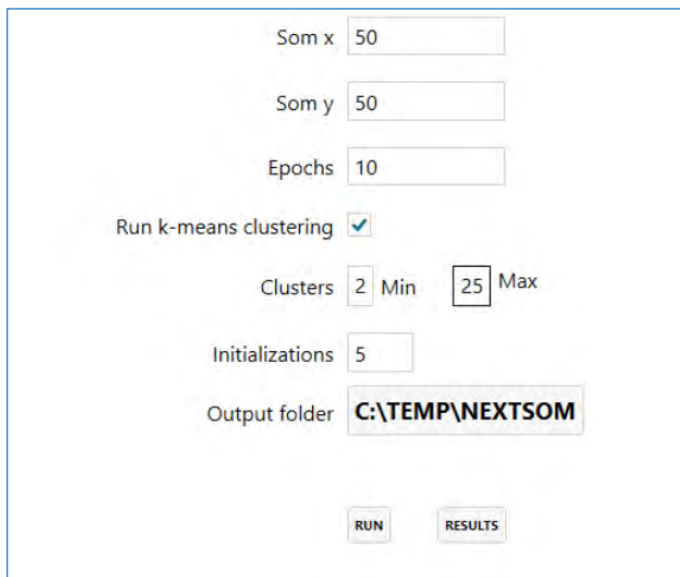
Figure 2. Studying the data.

The following steps are required:

- Select the North and East coordinates by selecting the correct column from the left side and choosing “Northing” or “Easting” from the right side.
- Exclude the data columns that you don't want to be used in SOM by selecting the correct column and choosing “Exclude”.
- Study the columns that you want to use as attributes in SOM using the histogram in the middle. The attributes should be normally distributed. If they are not, you can apply logarithmic transformation to the attribute values (“Log transform”) or limit extreme values (“Winsorize”) to make the histogram closer to normal distribution. Remember to click “Update histogram” to see changes in the histogram and “Save changes to dataset” to apply them to the dataset.
- Click “Next” to proceed to the next step

### 3.3 Choose SOM and k-means parameters

In the next step, you need to choose the parameters used in SOM and k-means clustering.



**Figure 3.** Choosing parameters for SOM and k-means.

#### Parameters for SOM

- Choose the size of the SOM using “Som x” and “Som y”. These refer to the number of SOM cells in x and y direction. The default values are calculated using one rule of thumb: the total number of cells is  $5 * \sqrt{\text{number of data points}}$  so both “Som x” and “Som y” are square root of that value. The larger the values, the more detailed the SOM, but it also takes more time to compute.
- “Epochs” is the number of times that the data set will be used when training the SOM. The default is 10. Small values result in faster computation, but possibly also in an inaccurate SOM. Larger values increase computation time, but might also improve the quality of the SOM. Usually the quality will not increase after certain amount of epochs.

#### Parameters for k-means

- You can choose to skip k-means clustering and run only SOM by removing the tick in “Run k-means clustering”.
- You need to select the minimum and maximum number of clusters. The default for minimum is 2 and maximum is 25. K-means requires a known number of clusters. This software applies k-means to the results of SOM using multiple values for the number of clusters and the most optimal number is chosen based on the smallest the Davies-Bouldin index.
- Choose the number of random “Initializations”. The default is 5. K-means utilizes random number generator in the algorithm and is sensitive to the initialization. Thus, this software runs k-means using different initializations and chooses the most optimal based on the smallest the Davies-Bouldin index.

- In addition, select a folder where all the results will be saved as “Output folder”.
- Click “Run” to run the software and after it, click “Results” to study the results.

More parameters will be available in the next software update.

### 3.4 Results

In the last step, you can see the results. These are divided between “Somspace results”, “Geospace results”, “Boxplots” and “Scatterplots”. You can access these using the buttons on top of the results.

#### Results in SOM space

These images show the resulting SOM using the attributes (also known as codebook vectors) and k-means clusters.

The k-means clusters are visualized in the last image with different colours for different clusters. All other images show the different attributes using a rainbow colour scheme, where blue is the lowest and red the highest value.

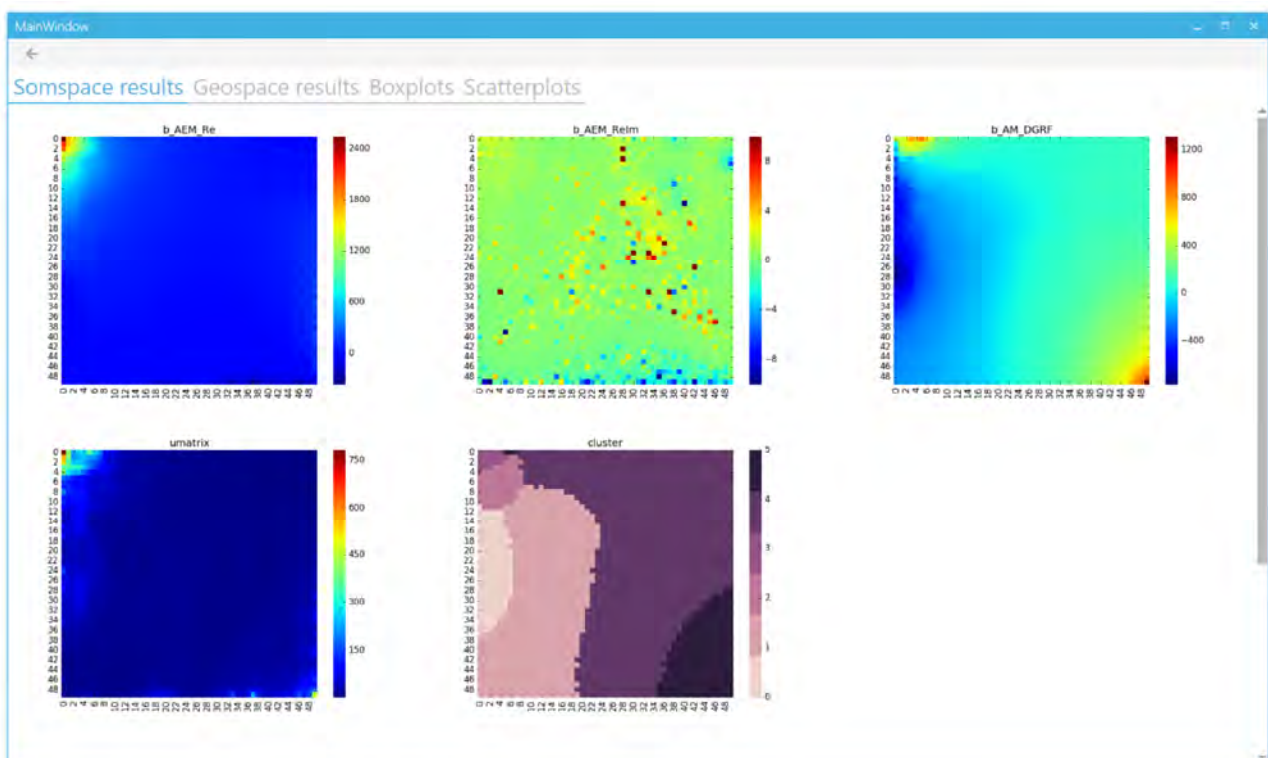
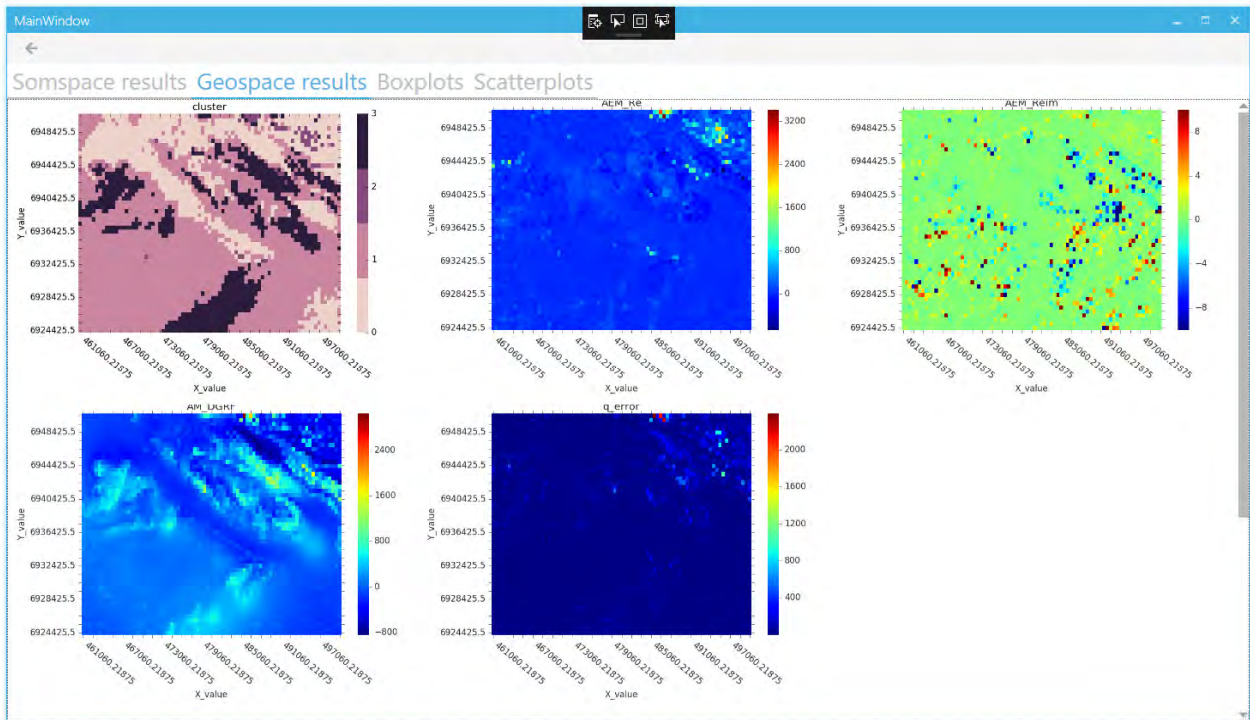


Figure 4. SOM space results.

### Results in geospace

These images show the k-means clustering results, original attributes and quantization errors in geographical space.



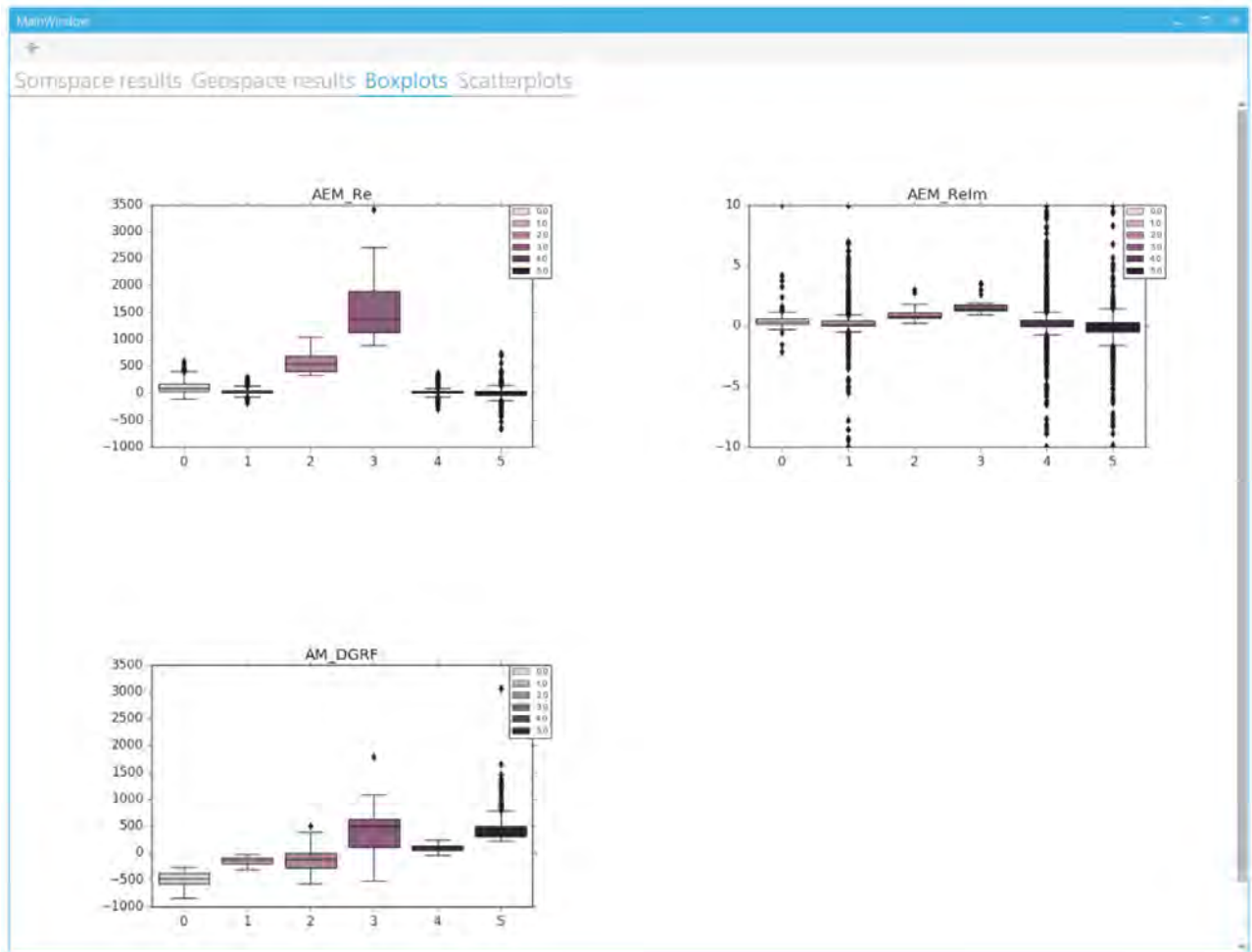
**Figure 5. Geospace results.**

The k-means clusters are visualized in the first image. Next images show the attributes of the original data set. The last image shows the quantization error in each location: It is the difference between the original data attributes and the SOM attributes of the cell where the data point has been projected to. High quantization error values show outliers in the data.



### Boxplots

These images show different attributes of the k-means clustering results as boxplots.

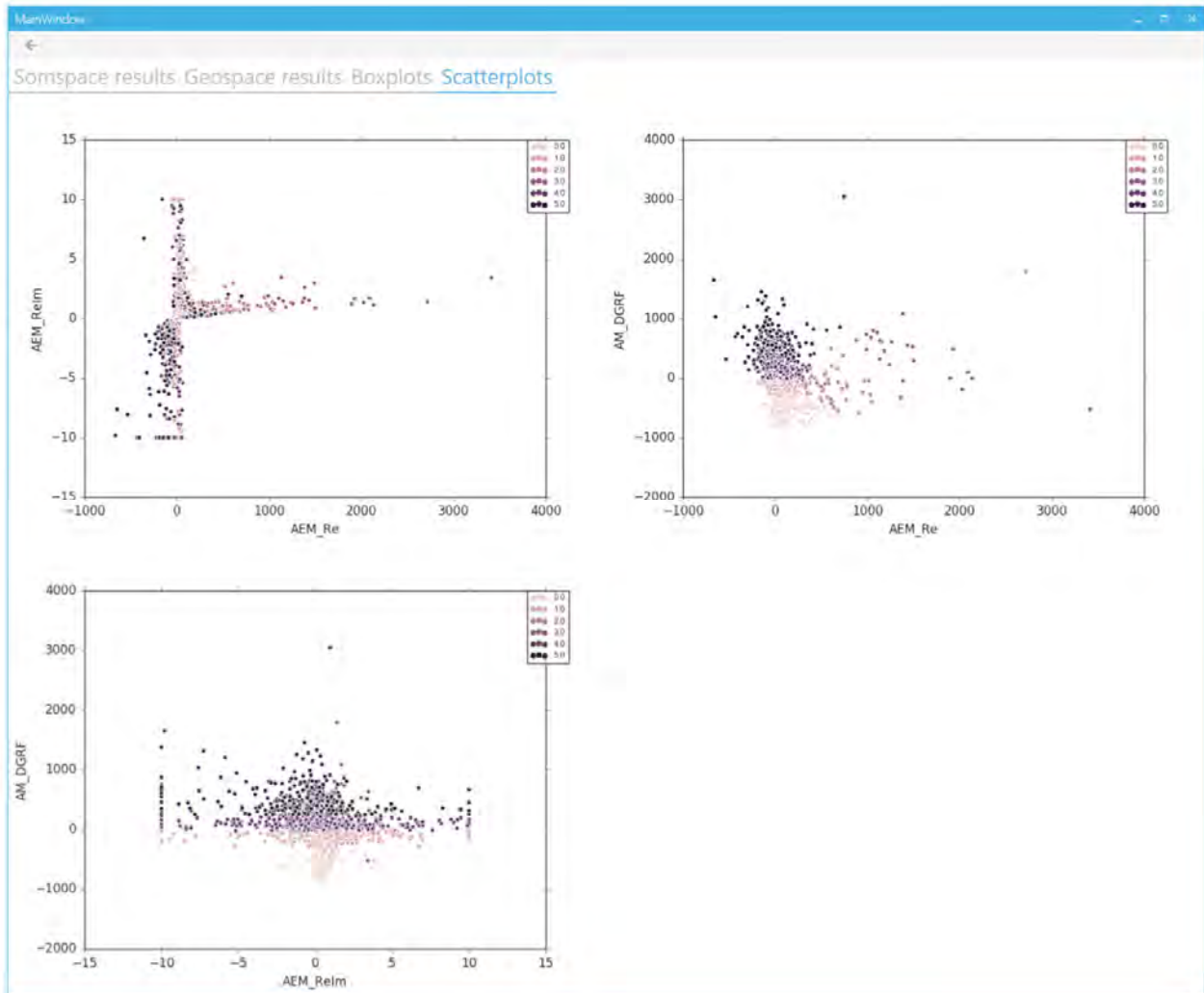


**Figure 6. Boxplot results.**

The window includes one image for each data attribute. In the image, there is one boxplot for each cluster. The boxplot describes the distribution of original data values: the line in the middle of the box is the mean value and the borders of the box are the first (25 %) and third (75 %) quartiles. The lines extend from the box borders to reach the minimum and maximum values, but no more than 1.5 times the size of the box. If there are any points outside the box and lines, they are visualized using points.

### Scatterplots

These images show different attributes of the clustering results as scatterplots.



**Figure 7. Scatterplot results.**

The window includes one image for each data attribute pair. The scatterplot shows each original data object as a dot in x,y-coordinates with one attribute as x and another as y. The dots are coloured based on the cluster that the object belongs to.

## 4 REFERENCES

Deliverable 4.11 Appendix 1: Technical Specification – *nextsomcore*

Deliverable 4.11 Appendix 2: Technical Specification – *GisSOM*.

Deliverable 4.12: SOM tool for advangeo® (under preparation, due in M18)

Deliverable 4.13: SOM tool for ArcGIS (under preparation, due in M18)

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