



MOLUSCE

MODULES FOR LAND USE CHANGE EVALUATION

QUICK HELP

LICENSE

EXCEPT WHERE OTHERWISE NOTED, CONTENT OF THIS WORK IS LICENSED
UNDER A CREATIVE COMMONS ATTRIBUTION-SHAREALIKE 3.0
UNPORTED LICENSE.

Contents

1	Plug-in overview	2
1.1	Introduction	2
1.2	What is MOLUSCE?	2
1.3	Functions	3
2	How to use MOLUSCE	4
2.1	Inputs	4
2.2	Evaluating correlation	5
2.3	Area Changes	6
2.4	Transition potential modelling	6
2.4.1	Artificial Neural Network (ANN)	7
2.4.2	Multi Criteria Evaluation (MCE)	10
2.4.3	Weights of Evidence (WoE)	12
2.4.4	Logistic Regression (LR)	13
2.5	Cellular Automata Simulation	14
2.6	Validation	15

Chapter 1

Plug-in overview

1.1 Introduction

Open source software platforms are progressively becoming widely used in the public and private sectors. In Geographical Information System (GIS), open source software packages such as QGIS are actively being developed. More importantly, customization and further development is possible since developers create specific plug-ins with flexibility.

Asia Air Survey Co., Ltd. (AAS) started to move towards open source software since 2012 becoming the first QGIS gold sponsor worldwide. Furthermore, open source software started to be used more extensively for internal use and in international project. Alongside with these recent changes AAS also started to develop open source solutions aiming to further extend its market.

1.2 What is MOLUSCE?

AAS released MOLUSCE (Modules for Land Use Change Evaluation) at FOSS4G 2013. MOLUSCE is a user-friendly plug-in for QGIS 2.0 and above. MOLUSE is designed to analyse, model and simulate land use/cover changes. The plug-in incorporates well-known algorithms, which can be used in land use/cover change analysis, urban analysis as well as forestry applications and projects.

MOLUSCE is well suited to:

- analyse land use and forest cover changes between different time periods;
- model land use/cover transition potential or areas at risk of deforestation;
- and simulate future land use and forest cover changes

1.3 Functions

MOLUSCE user interface offers an easy-to-use interface with specific modules and functions. Following is a brief description of basic modules in MOLUSCE.

Input module Land use/cover maps from different epochs, biophysical and socio-economic driving factor data such as road network, rivers, topography, population etc., are loaded in the input module.

Area change analysis Computes land use/cover changes between two time periods (T1 and T2). Land use/cover change transition matrices as well as land use change maps are produced.

Modelling methods Four methods, namely Artificial Neural Networks (ANN), Logistic Regression (LR), Multi-Criteria Evaluation (MCE) and Weights of Evidence (WoE) are used for modelling land use/cover change transition potential.

Simulation Displays transition potential maps, certainty function (experimental) and simulation results. A simulated (projected) land use/cover map is produced based on a Monte Carlo Cellular-automata modelling approach.

Validation This sub-module incorporates Kappa statistics (standard Kappa, Kappa histogram and Kappa location), which will be used to validate the accuracy of the simulated land use/cover maps.

Chapter 2

How to use MOLUSCE

2.1 Inputs

A screenshot of Inputs tab is shown in Figure 2.1.

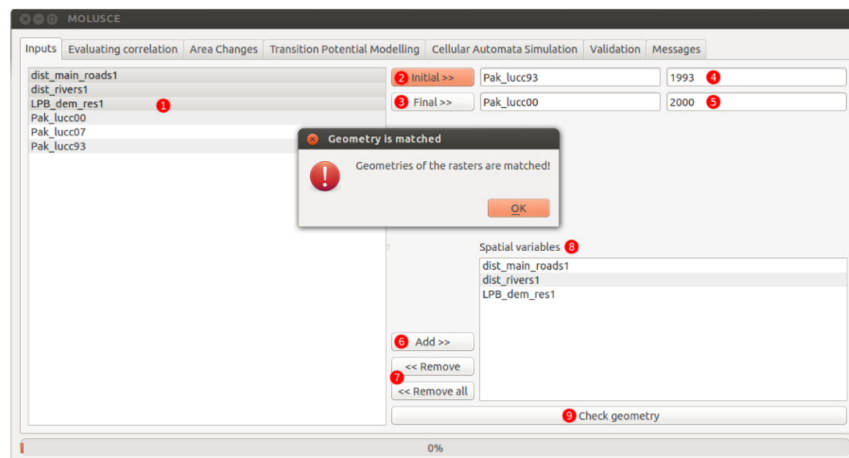


Figure 2.1: Inputs tab

Data can be loaded using the inputs tab. To load your data follow the next steps (Figure 2.1).

- Step 1. Load all rasters that you need in the input tab.
- Load the initial and final land use/cover maps as shown in steps 2 and 3. The base map determines the geometry of all the output files, pixel size, scaling and projection.
- Steps 4 and 5 show the corresponding years. (The user can type here If the corresponding years do not appear automatically).

- Steps 6 and 7 show how the add/remove buttons used to add or remove spatial variables.
- The selected spatial variables are shown in step 8.
- The check geometry button is a mandatory step to check if the geometry of the selected raster is matched (step 9).

Note: It is important when layers are added to QGIS that the No Data Value (NDV) is set. If this is not done, MOLUSCE will process NDV areas as land use/cover classes, increasing processing time and confusing the model calibration. MOLUSCE picks up the NDV of the input (base) layer and propagates it to any output maps generated, along with the geometry of the base layer.

2.2 Evaluating correlation

The **evaluating correlation** module contains three techniques for performing correlation analysis:

1. Pearson's correlation.
2. Cramer's coefficient.
3. Joint information uncertainty.

The user can choose between a two-way raster comparison by selecting **first raster** and **second raster** or **check all rasters** loaded into MOLUSCE.

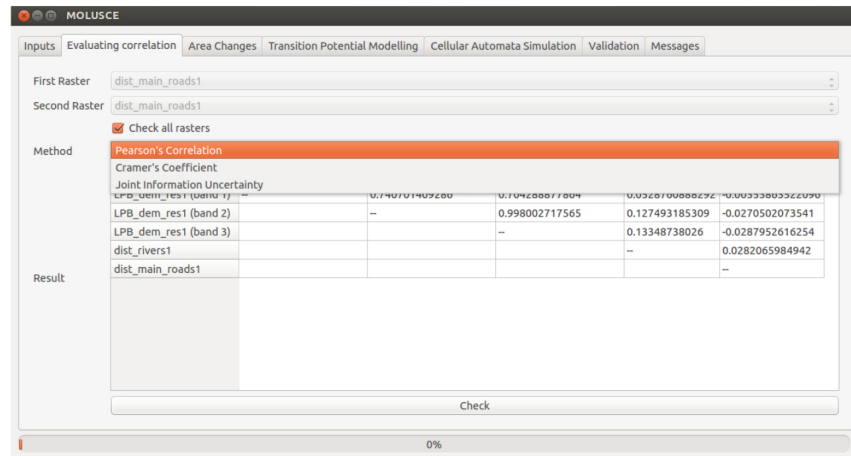


Figure 2.2: Evaluation correlation tab

The user can run correlation by pressing the **check** button located at the bottom of the window (see Figure 2.2).

Note: The **Cramer's coefficient** and **joint information uncertainty** work only with categorical data. The data should be converted to categorical data (eg. using GRASS).

2.3 Area Changes

A screenshot of **Area Changes** tab is shown in Figure 2.3.

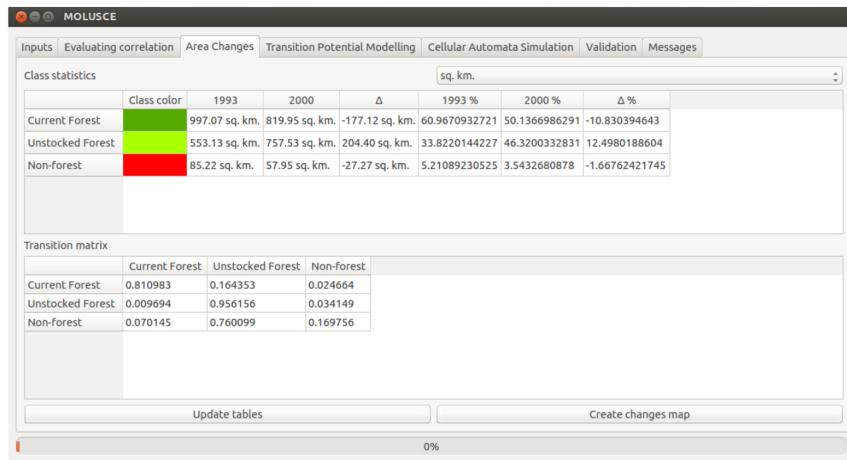


Figure 2.3: Area Changes tab

The **update tables** button produces **class statistics** and **transition matrix** tables.

The **class statistics** table shows the initial and final land use/cover change (LUC) areas.

The **transition matrix** shows the proportions of pixels changing from one land use/cover to another.

The **create change map** button will generate a map of change classes. This will be added automatically to QGIS and saved as a GeoTiff.

Note: Data from tables can be copied and pasted directly into spreadsheet programs, simply by selecting the desired rows/columns and by pressing the “Ctrl + C” keyboard combination.

2.4 Transition potential modelling

MOLUSCE uses Artificial Neural Network (ANN), Multi Criteria Evaluation (MCE), Weights of Evidence (WoE) and Logistic Regression (LR) methods to model land use/cover transition potential. The user can select a **method** from the drop down menu.

ANN, WoE and LR are machine learning models; they "look" at the training samples and try to find patterns that are "hidden" in the samples. However MCE is different: it is a model in which an expert (the user) describes the importance of factors according to his/her knowledge of the domain area. After that, the expert's knowledge is transformed into a computational model that estimates transition potentials.

2.4.1 Artificial Neural Network (ANN)

MOLUSCE uses full-connected Multi-Layer Perceptron for modelling. A screenshot of an example of ANN model settings is shown in Figure 2.4.

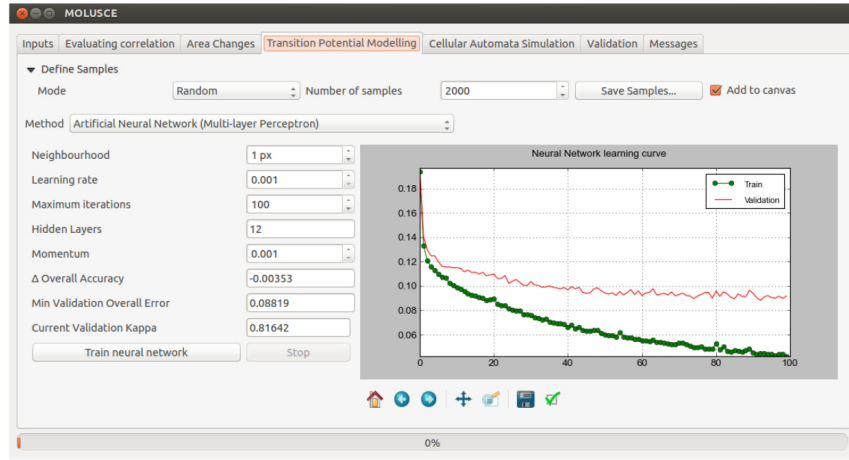


Figure 2.4: Transition potential modelling: ANN model

The **define samples** function specify number of samples and sampling mode. In addition, the sampling points created can be saved and displayed.

Inputs Five inputs are used to customize the ANN modelling:

- The **neighbourhood** defines count of neighbour pixels around current pixel. Size=1 means 9 pixels (3x3 region), size=2 means 25 pixels (5x5), etc.
- **Learning rate**, **momentum** and **max iterations number** define parameters of learning. Big learning rate and momentum allow fast learning, but the learning process can be unstable (spikes on the graph). Small learning rate and momentum means stable but slow learning.
- **Hidden layers** input string takes a list of numbers: N_1, N_2, \dots, N_k , where N_1 is number of neurons in 1st hidden layer, N_2 is number of neurons in 2nd hidden layer and so on, N_k is the number of neurons

of the last hidden layer (k-th layer). For example if the user types in the input string "2" then a network with 1 hidden layer and 2 neurons will be created. In order to create a network with 2 hidden layers the user should insert 2 numbers, such as "10 2" which will create a network with 10 neurons in the first hidden layer and 2 neurons in the second.

Outputs The following outputs are proposed (for the current learning iteration):

- The graph area. Contains errors of training and validation sets. It is the main information about learning process. The graph can be edited and saved as image.
- The **min validation overall error** contains information about min reached error on validation set of samples.
- The **delta overall accuracy** contains difference between min reached error and current error.
- The **current validation Kappa** shows the Kappa value.

The training process can be started by pressing on the **train neural network** button and stopped at any time using the **stop** button.

Learning algorithm analyses the reached accuracy on training and validation sets of samples and stores the best neural net in memory. The training process finishes when the **maximum iteration number** is reached, and the best neural net will be used to produce transition potentials.

How to train the model

There are few parameters that can dramatically change quality of trained model. The main of them are:

- ANN complexity (number of layers and neurons): complex models are more powerfull, but training them properly is hard process, complex models require more samples to train etc.
- learning rate and momentum: large values of learning rate allow to train quickly, but the training process may be unstable in this case; momentum helps stabilize the training.
- sample count: large sample count allows to train complex models, but the training becomes slow and requires more computer memory (which can be issue for large rasters).

So process of ANN training is a trial and error method: during the training the uses searches "good" parameters that lead to "good" model. The training graph is the main tool that provide information about the training process and model quality.

Bellow we provie an example of ANN training, discuss some potential caveats and how to fix them.

Unstable learning Figure 2.5 shows an example of unstable learning process: there are a lot of spikes on the graph. Note that the learning rate is 0.1.

A stable training process is one of the necessary conditions for good ANN model: we can't expect to learn something usefull if the values of its parameters show dramatic changes with every iteration.

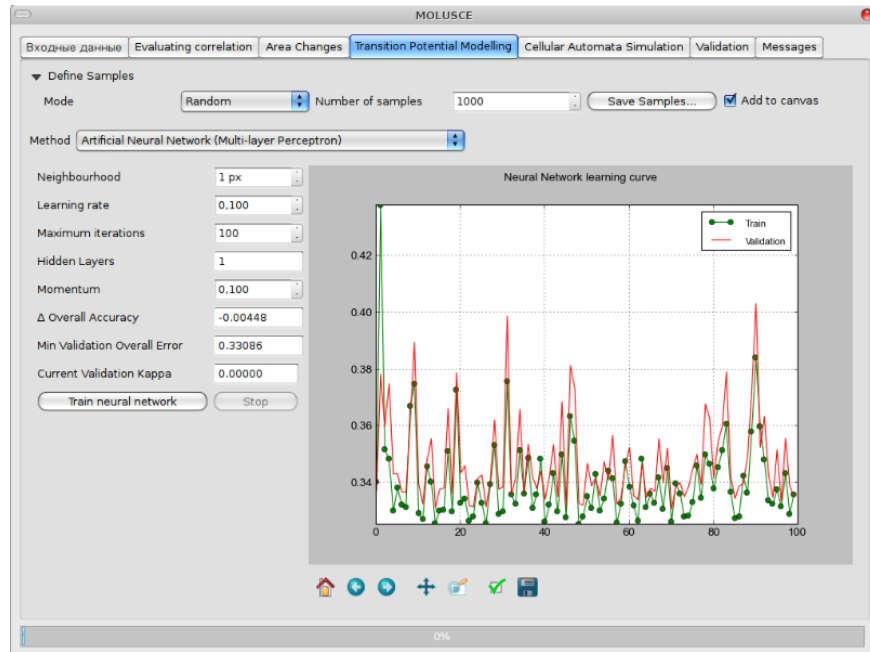


Figure 2.5: ANN model: spikes on the graph (unstable training)

We can fix the issue if decrease the value of this parameter. Figure 2.6 shows that if we set learning rate to 0.001 the learning curves become smooth. The smooth form of lines for training and validation errors indicates a stable training process. This means that the internal values of the parameters of ANN model do not change suddenly with every iteration.

But we can see that the model isn't perfect (look at the Kappa and validation error).

Improving the quality Validation error and Kappa show that the model is not perfect (see Figure 2.6).

We can add a neuron for the hidden layer, it makes the net more complex. The learning curves are shown in Figure 2.7. We can see that Kappa is not zero and validation error decreases. It means that the model is getting better.

What about the overfitting? The model shows some overfitting (there is some difference between validation and training errors) but it is not significant.

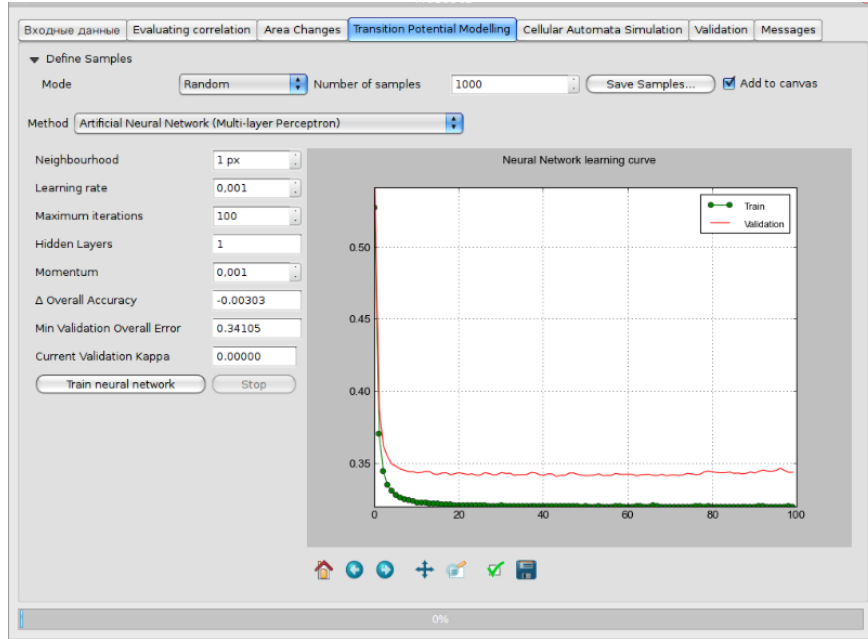


Figure 2.6: ANN model: smooth learning curves

To get an example of clearer overfitting, set 12 neurons in the hidden layer. The result presented in Figure 2.8.

We can see that the training error is decreasing but the validation error is still nearly constant. It is clear signal about overfitting. To avoid it we have two choices:

- decrease number of neurons;
- increase sample count.

Figure 2.9 shows training on 20,000 samples.

We can see that the training and validation errors are close. Overfitting is not present.

2.4.2 Multi Criteria Evaluation (MCE)

The MCE model [2] takes into account the expert point of view regarding the importance of the predictors. The user makes pairwise comparisons of the factors and describes their importance for the transition potentials. The comparisons are made using Saaty's scale (see Table 2.1).

For example, coefficient "1" between factors A and B means that the factors have equal importance for the transition process. Coefficient "9" between factors A and B means that factor A absolutely exceeds factor B in importance.

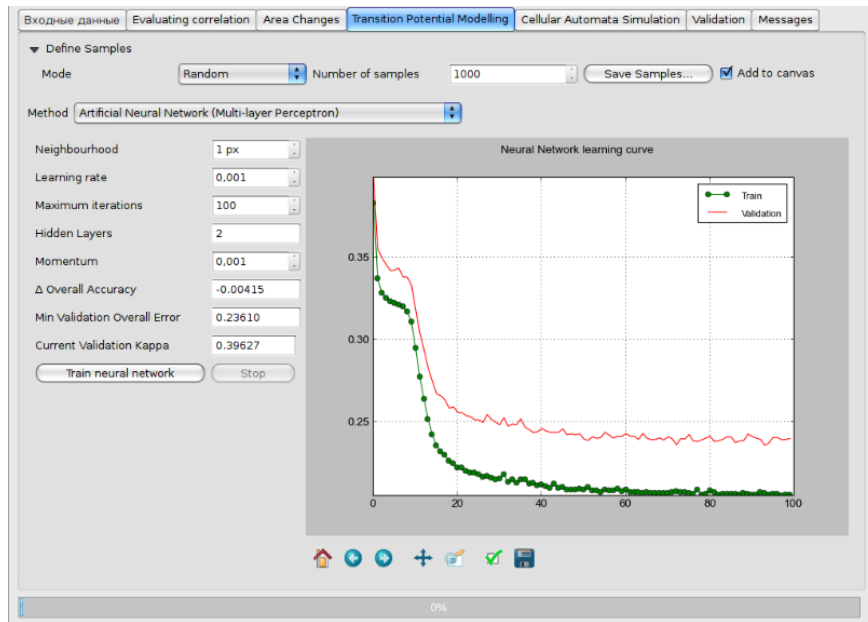


Figure 2.7: ANN model: better model with slight overfitting

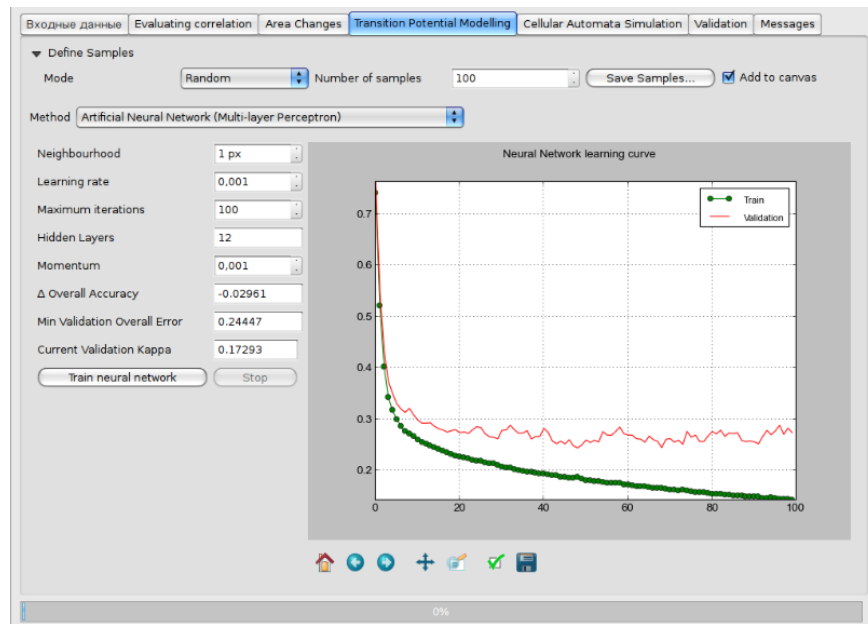


Figure 2.8: ANN model: overfitting example

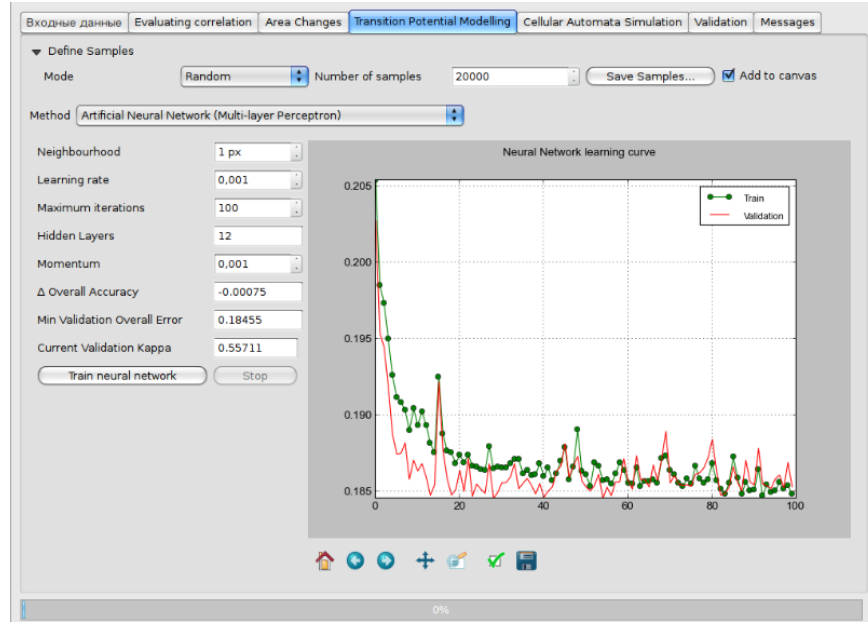


Figure 2.9: ANN model: final example

Table 2.1: Saaty's scale

Rating	Preference
1	Equal importance
3	Moderate importance
5	Strong importance
7	Very strong importance
9	Extreme importance
2, 4, 6, 8	Intermediate values

A screenshot of an example of MCE model settings is shown in Figure 2.10.

The user can set the values inside the pairwise comparison matrix.

The user can also select which classes to use to train the model by changing the from class and to class values located at the bottom of the window.

The model can be started by pressing the train model button.

The weights of the spatial variable will then appear in the weights matrix.

2.4.3 Weights of Evidence (WoE)

The WoE [1] method proposes two ways of defining the range breaks. The user can define a number of intervals or specify the range breaks values.

When the calculate range breaks button is pressed the weights information for each transition are produced.

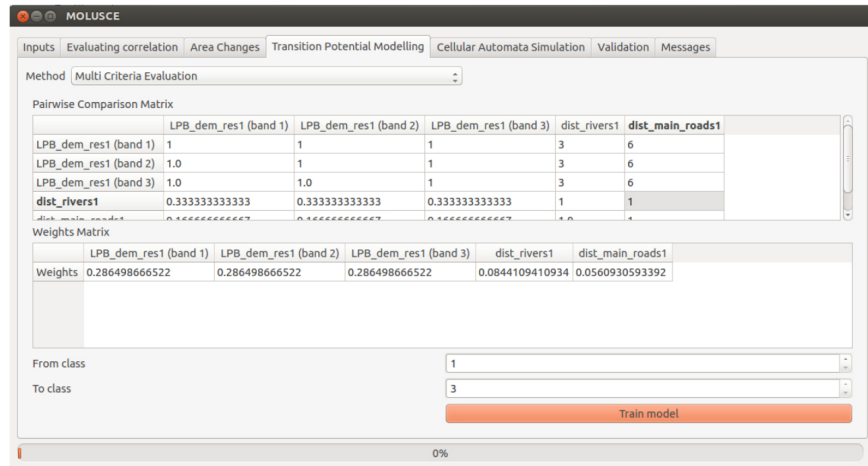


Figure 2.10: Transition potential modelling: MCE model

Once the user is satisfied with the weights the **train model** button can then be pressed.

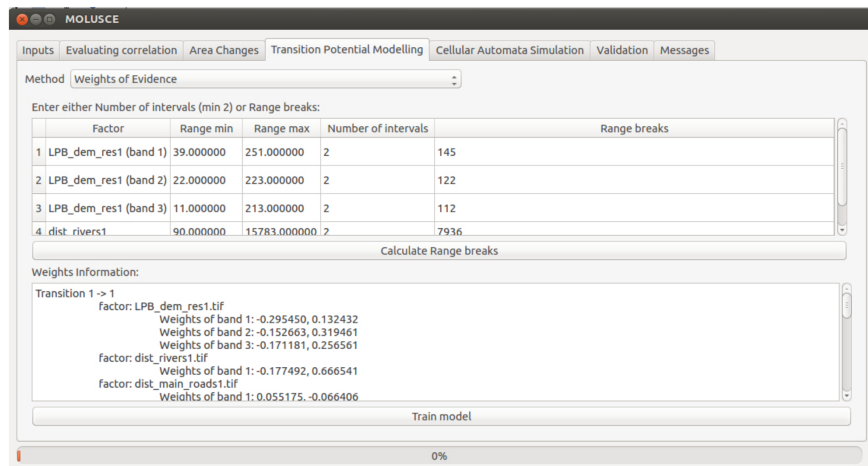


Figure 2.11: Transition potential modelling: WOE model

An example of WoE model settings is shown in Figure 2.11.

2.4.4 Logistic Regression (LR)

An example of LR model settings is shown in Figure 2.12.

The LR method offers the possibility to **define samples** (number of samples and sampling mode) as well as save and display the sampling points created.

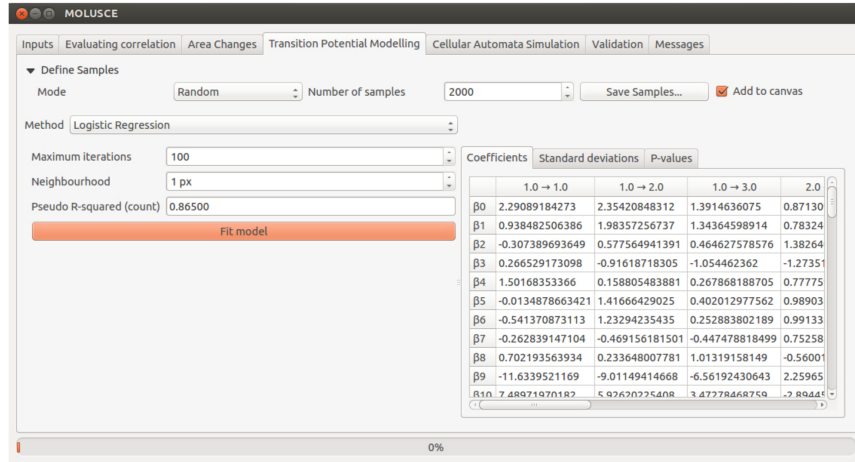


Figure 2.12: Transition potential modelling: LR model

Inputs Two inputs are used to customize the LR modelling:

- The **maximum iteration** defines the total number of iterations.
- The **neighbourhood** defines count of neighbour pixels around current pixel. Size=1 means 9 pixels (3x3 region), size=2 means 25 pixels (5x5), etc.

Outputs The following outputs are proposed (for the current learning iteration):

- The **pseudo R-squared** shows the goodness-of-fit.
- The **coefficients** tab.
- The **standard deviations** tab.
- The **p-values** tab.

The user can run the model by pressing on the **fit model** button.

Note: For additional information on the LR outputs please consult the “Technical information: Methods and Algorithms” manual.

2.5 Cellular Automata Simulation

Once one method has been chosen and trained from the **transition potential** modelling tab, the user can then access to the **cellular automata** simulation tab.

Be aware that MOLUSCE will keep in memory the latest method processed, if for example the user runs first the ANN and then the LR methods, the cellular automata simulation tab will retain the results from the LR.

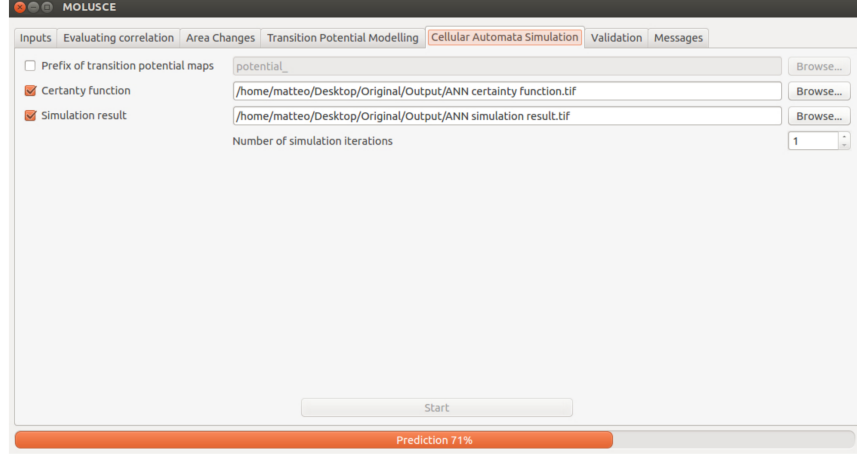


Figure 2.13: Simulation tab

Three type of output maps are produced. A check box at the beginning of each output is provided to allow the user to enable only what it is necessary. A **browse...** button located at the end of each output allows to save each map.

- The **prefix of transition potential maps** button allows to select prefix of the names of transition potential maps. Transition potential map shows the probability or potential to change from one land use/cover class to another. Transition potential values range from 0 (low transition potential of change) 100 (high transition potential). Transition potential maps will be produced from the corresponding land use/cover changes (e.g., “forest to unstocked forest” transition potential, “forest to non-forest transition potential).
- The **certainty function**.
- The **simulation result** produces a simulated land use/cover map.

An example of simulation tab is shown in Figure 2.13.

2.6 Validation

The **validation** tab allows the user to check, validate and compare the simulation results. **Reference** and **simulated land use/cover maps** must be loaded in order to start the validation process. The former indicates a land use/cover map (T_3). T_1 refers to the initial land use/cover, while T_2 refers to the final

land use/cover used in the model). A **browse** button located at the end of each output allows to load the desired map. A two way map comparison is performed from reference land use/cover (T_3) and simulated land use/cover maps.

In order to perform a three way map comparison, the user can check the **risk class validation map** check-box. Although it is not shown explicitly, the three-way map comparison uses the initial land use/cover map (T_1), the reference land use/cover map (T_3) and the simulated land use/cover map.

The **multiple-resolution budget** method [3, Chapter 17] is also used in MOLUSCE. The user can decide the **number of validation iterations** and can start the process by pressing the **start validation** button. The graph can be edited and saved as image.

The overall accuracy (**% of correctness**, **Kappa (overall)**, **Kappa (histo)** and **Kappa (loc)**) can be executed by clicking the **calculate Kappa** button.

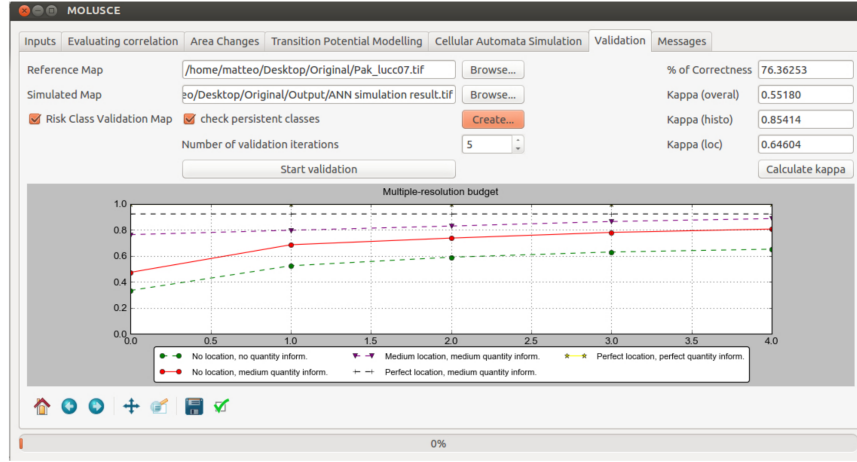


Figure 2.14: Validation tab

An example of validation tab is shown in Figure 2.14. We discuss here some parts of the graph, see full details in [3, Chapter 17]. The chapter propose 15 types of statistic, but MOLUSCE calculates and plots 5 of the most important of them:

- No location information, no quantity information;
- No location information, medium quantity information;
- Medium location information, medium quantity information;
- Perfect location information, medium quantity information;
- Perfect location information, perfect quantity information.

These lines agreement metrics. They are measured for maps at different scales. The zero point on the abscissa axis corresponds to the original scale of the maps, the point one corresponds the maps scaled down by half (2^1), the point two corresponds the maps scaled down by quarter (2^2), etc.

Medium location information, medium quantity information is "the main" line. It represents sequence of measures of agreement between the reference map and the simulated map. Higher values of the line show higher agreement.

Other lines show measures of agreement between the reference map and those that describe different hypothetical situations. For example the line "No location information, no quantity information" is the agreement due to chance. Usually it lies below the "Medium-Medium" line, in this case the simulated map has better agreement with the reference than random generated map.

The upper line, "Perfect-Perfect," represents the highest possible agreement. It corresponds to the case when we have complete information about the reference map in advance and can place all changes in their correct locations during the simulation. Thus, this line shows the case when no errors are committed. Usually the "Medium-Medium" line lies below this line.

Other lines represent other hypothetical situations when we know more or less information about the reference map. This information consists of two parts: information about the quantity of pixels (quantity of changes in our case) and information about the locations of changes.

Bibliography

- [1] Graeme F Bonham-Carter. Weights of evidence modelling: a new approach to mapping mineral potential. *Statistical applications in the earth sciences*, pages 171–183, 1989.
- [2] J Ronald Eastman. Multi-criteria evaluation and gis. *Geographical information systems*, 1(1):493–502, 1999.
- [3] Robert Pontius and Beth Suedmeyer. *Components of Agreement between Categorical Maps at Multiple Resolutions. Chapter 17 in Ross S Lunetta and John G Lyon (eds). Remote Sensing and GIS Accuracy Assessment.*, pages 233–251. CRC Press: Boca Raton FL.