**GCRF GUI TOOL**

# 1. Introduction

Structured regression models are designed to use relationships between objects for predicting output variables. In other words, structured regression models consider the attributes of objects and dependencies between the objects to make predictions. General objective of regression is to predict the output variable Y as accurately as possible given an input vector of attributes X. Traditional supervised learning models, like neural networks, use only information contained in X to predict Y, while structured regression models use dependencies among outputs to improve final predictions. These problems can be seen as a graphs, where the nodes correspond to objects with attributes X and outputs Y, while the links of this graph contain weighs that are given by the similarity measures. We usually have some prior knowledge about relationships among the outputs Y. Mostly, those relationships are application-specific where the dependencies are defined in advance, either by domain knowledge or by assumptions, and represented by statistical models. For example relationships between hospitals can be based on similarity of their specialization, relationships between pairs of scientific papers can be presented as the similarity of sequences of citation, relationships between documents can be quantified based on similarity of their contents, etc.

The Gaussian Conditional Random Fields (GCRF) model is one type of structured regression models that incorporate the outputs of unstructured predictors (based on the given attributes values) and the correlation between output variables in order to achieve a higher prediction accuracy. This model was first applied in computer vision, but since then it has been used in different applications, and extended for various purposes.

GCRF GUI TOOL integrates various GCRF methods and supports training and testing those methods on real-world data from different domains.

To calculate the regression accuracy of all methods, we used $R^2$ coefficient of determination that measures how closely the output of the model matches the actual value of the data. A score of 0 indicates a very poor matching, while a score of 1 indicates a perfect match.

## 2. Installation

Download latest zip file from following link:

https://drive.google.com/folderview?id=0B_vOEFyds9xYdXlHbGswY3NuSjA&usp=sharing

Extract zip to the desired location. The structure of extracted folder *GCRF_GUI* is presented at Figure 1. The executable file *gui.jar* is located in the folder *GCRF_GUI*.

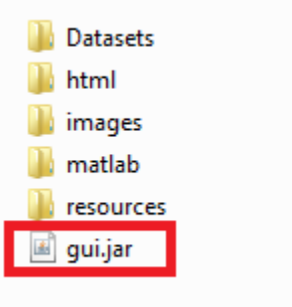


Figure 1. Folder structure

The *matlab* folder contains MATLAB source code for all methods that are implemented in MATLAB. The *html* folder contains files for Help. The *Datasets* folder contains dataset samples that can be used to test specific methods. Samples are provided in .txt files, in the format that is required by this tool. In these samples files are denoted as follows:

- x.txt - attributes
- y. txt - desired output
- s.txt - graph that presents relationships between objects

Each time when new dataset is added it will be stored in this folder.

## 3. Configuration

When you run *gui.jar* at the first time Configuration panel will be displayed and main menu will be disabled (Figure 2). Most of the parameters have default values and those values can be changed. In order to successfully save configuration you should insert path to *matlab.exe* file. If you do not want to use MATLAB you can uncheck "*Use methods implemented in MATLAB*" check box. In that case in GUI you will see only methods that are implemented in Java.

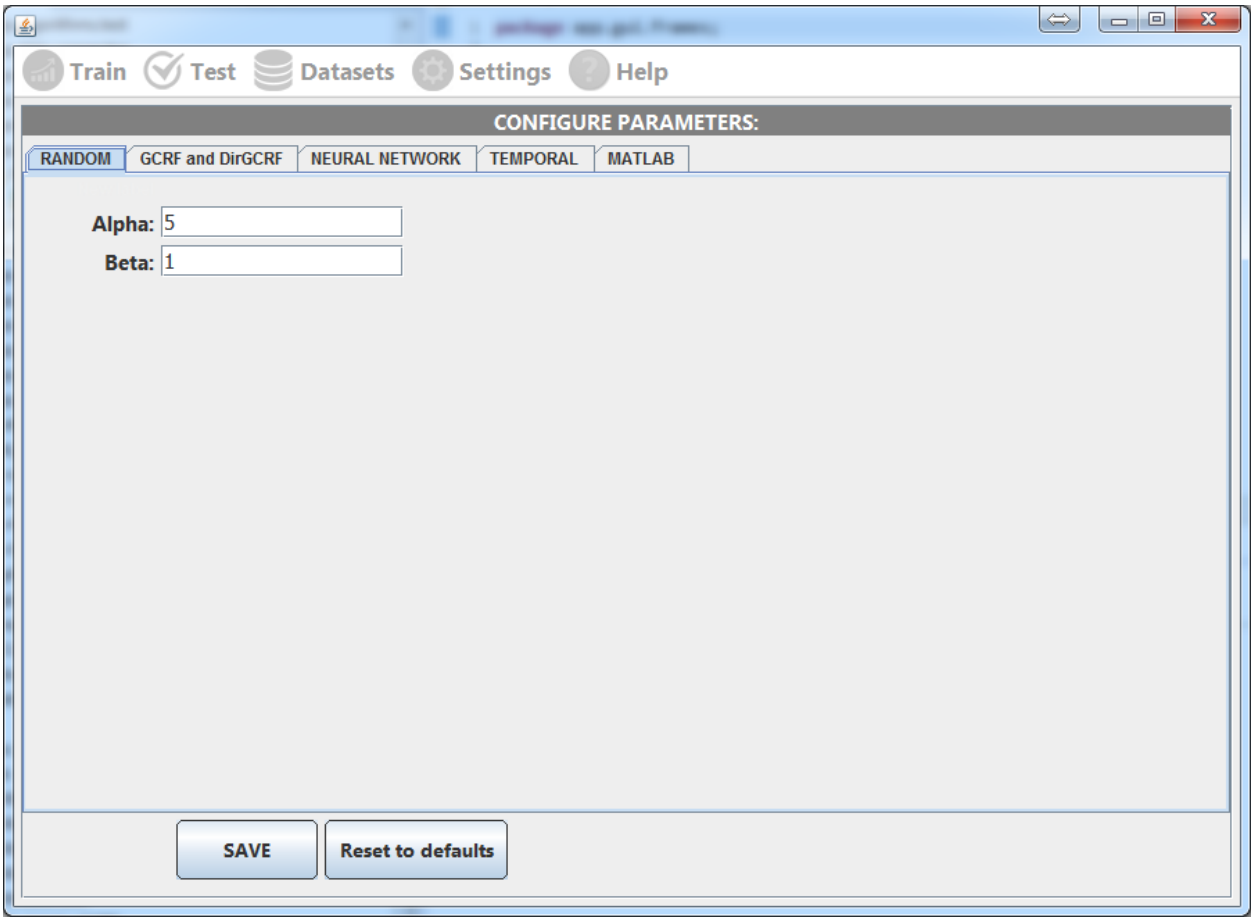


Figure 2. Configuration panel

When you click *Save* button Configuration panel will disappear and main menu will be enabled. If you want to change default values of the parameters Configuration panel can be opened from main menu: Settings->Configuration.

## 4. Train on networks

*Train on networks* menu item (Figure 3) is used to train following GCRF methods:

- Standard GCRF

- Directed GCRF (DirGCRF) – method that extends the GCRF to allow modeling asymmetric relationships (directed graphs)
- Unimodal GCRF (UmGCRF) – method that extends the GCRF parameter space to include negative values

Note: UmGCRF is visible only if you checked use MATLAB in configuration panel.

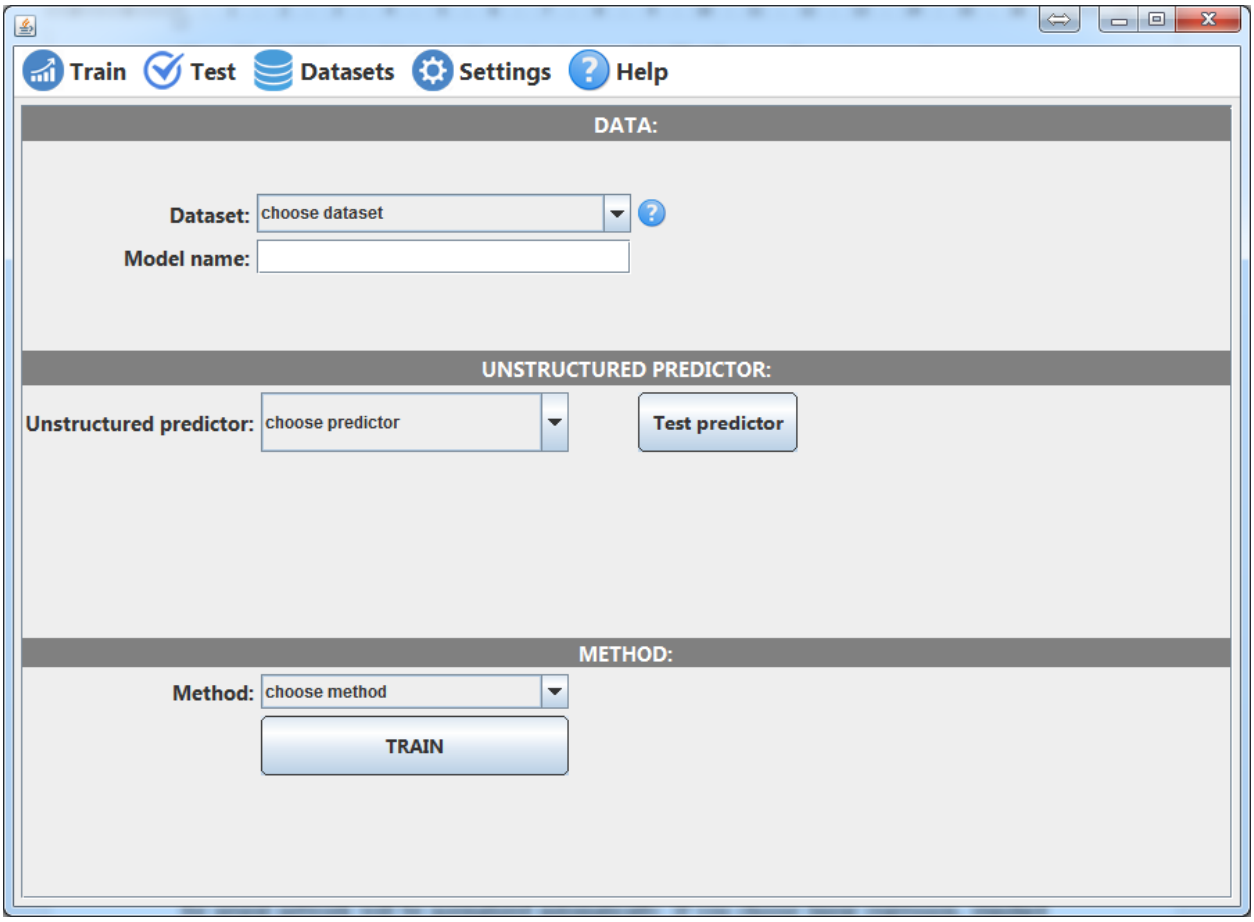


Figure 3.Train on networks

Two unstructured predictors can be selected: neural networks and linear regression. If you choose neural network you should insert the number of hidden neurons and the number of iterations. Data for neural network will be normalized automatically. If you choose linear regression, standard linear regression or multivariate linear regression will be applied, depending on number of attributes. When you select the desired method from *Method* combo box fields for that method's parameters will automatically show bellow the combo box. When you click *Train* button, the training process will start. Example of training results is presented at Figure 5.

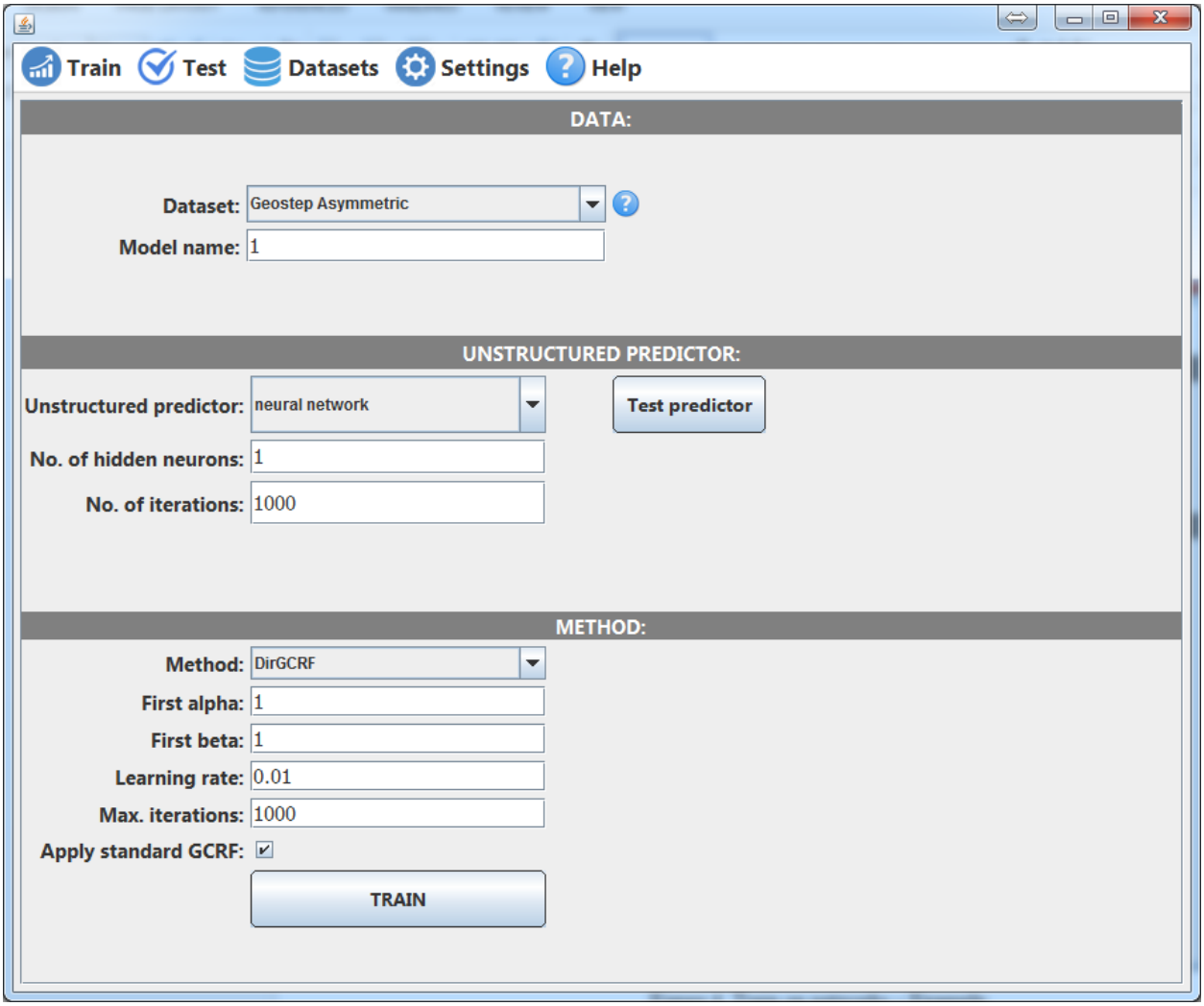


Figure 4. Train on networks – Example

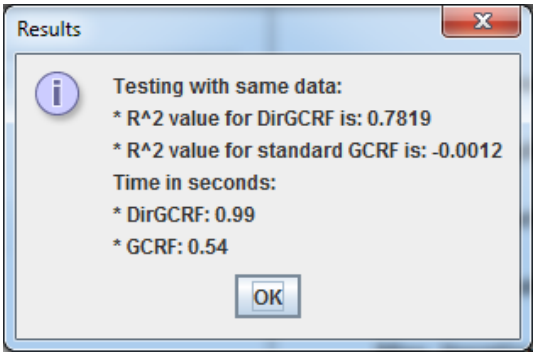


Figure 5. Train on networks – Example of the training results

Each time when you train new model, folder for that model will be created in *GCRF_GUI* folder. Models will be grouped by methods. After training process, folder for each model will contain following folders (example is presented at Figure 6):

1. data - where .txt files with data are copied
2. nn, lr or mlr – contains files with parameters for unstructured predictor
3. parameters – contains files with parameters for used method

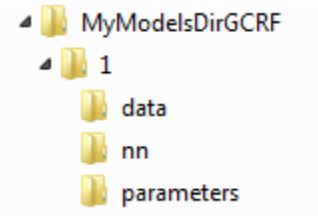


Figure 6. Train on networks – Example of the folder for model “1”

## 5. Test on networks

When model is trained we can test it by using menu item *Test->Test on networks* (Figure 7). When testing process is finished, $R^2$ value will be shown and the predicted values will be exported to .txt file and saved into model's folder, into subfolder *test* (Figure 8).

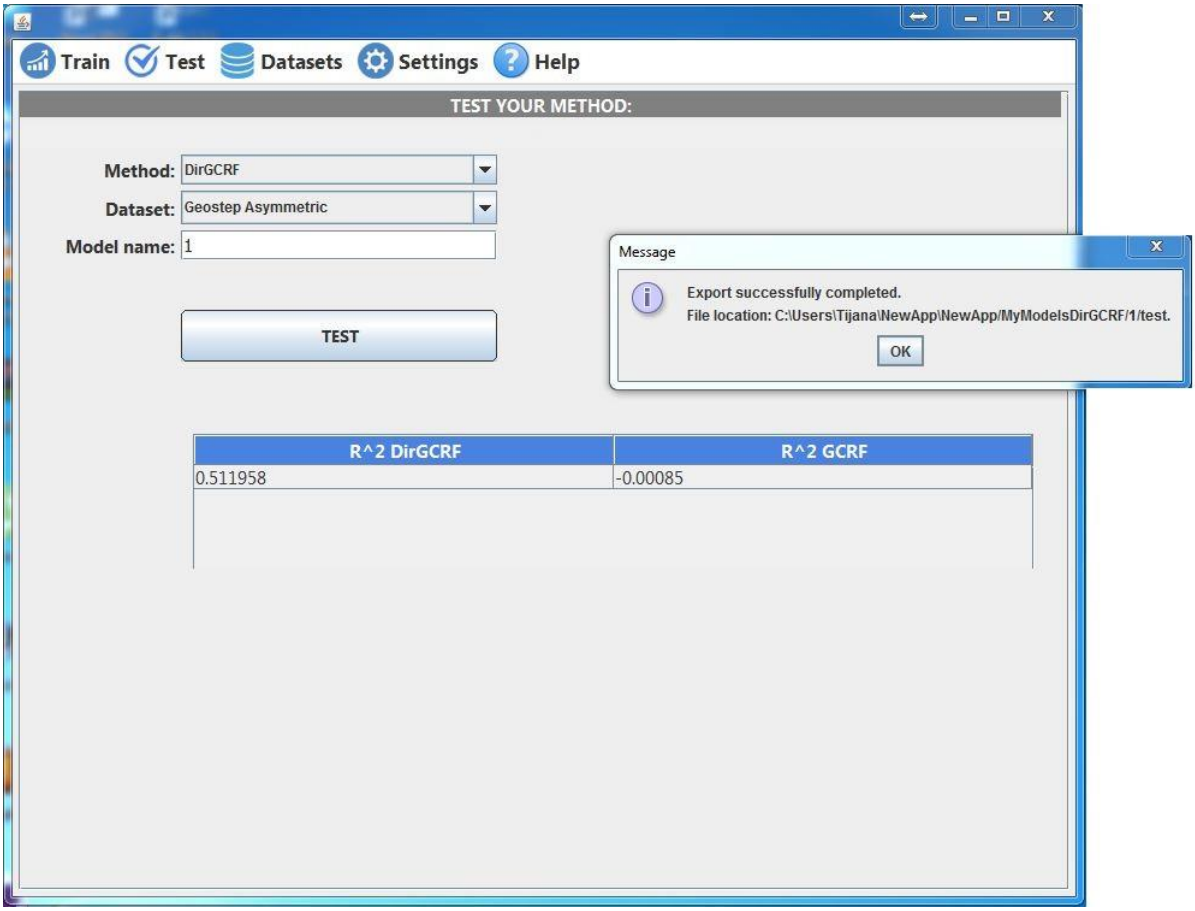


Figure 7. Test on networks – Example

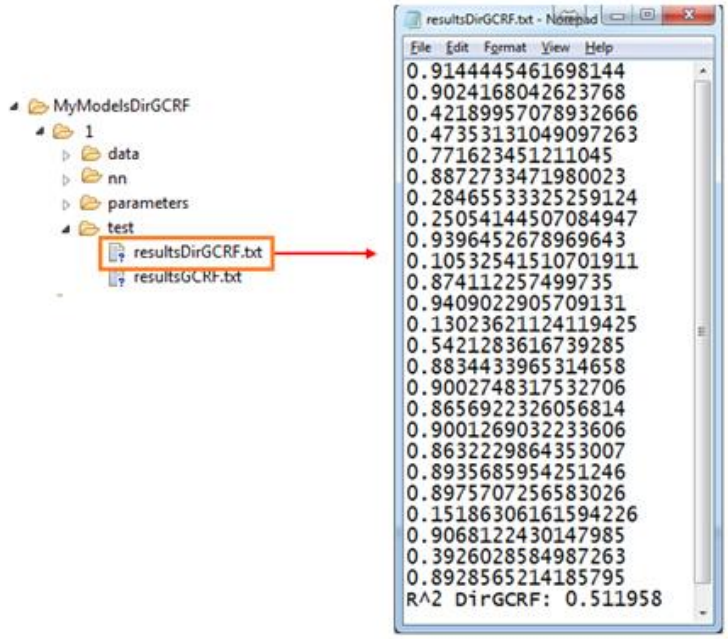


Figure 8. Example of the file with test results for model "1"

## 6. Train on temporal networks

*Train on temporal networks* menu item (Figure 9) is used to train following GCRF methods:

- Representation Learning based Structured Regression (RLSR) – method that is able to learn hidden representation of inputs, and structure among outputs simultaneously
- Uncertainty propagation GCRF (up-GCRF) – GCRF method for propagating uncertainty in temporal graphs by modeling noisy inputs
- Marginalized Gaussian Conditional Random Fields (m-GCRF) – GCRF method for dealing with missing labels in partially observed temporal attributed graphs.

Note: All methods are visible only if you checked use MATLAB in configuration panel.

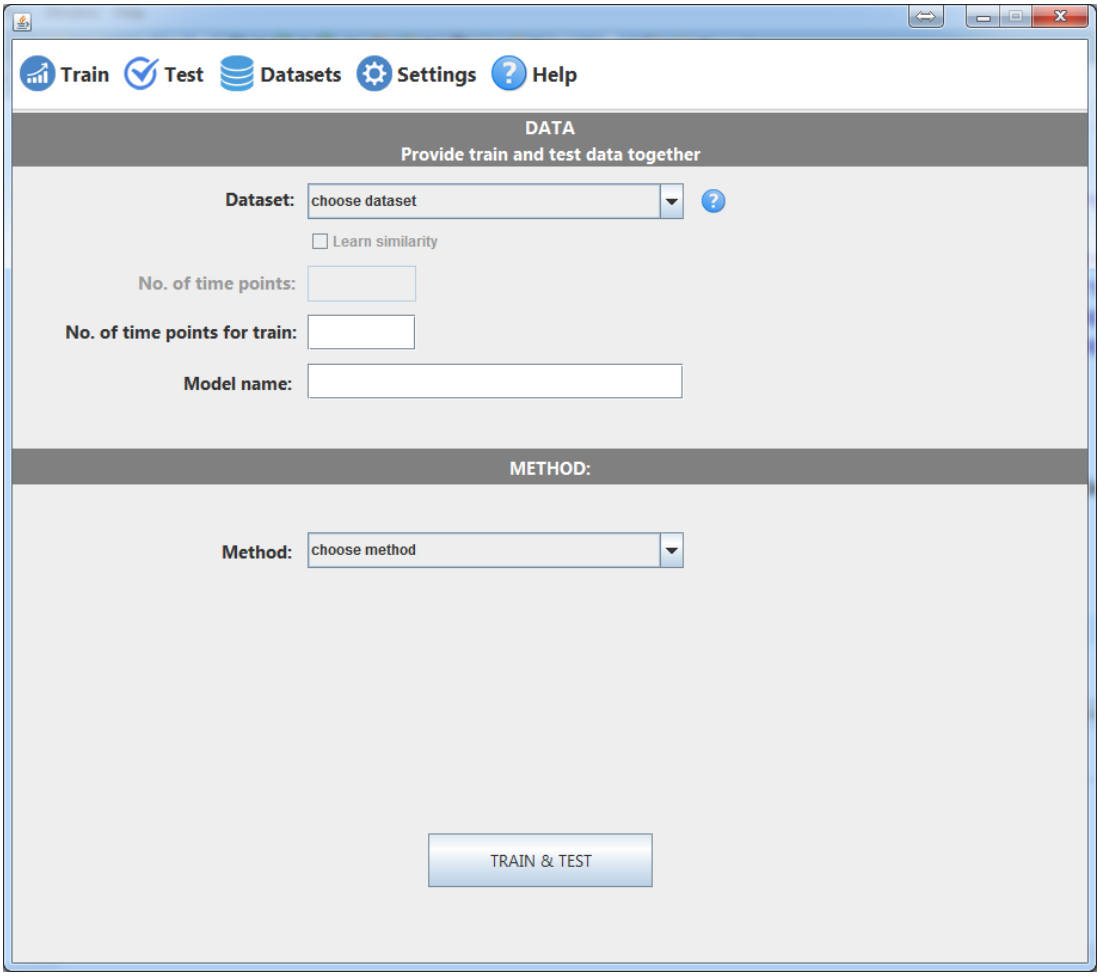


Figure 9. Train on temporal networks

For these methods data for training and data for testing should be provided together, and both processes will be completed when you click *Train & Test* button. RLSR and up-GCRF methods do not require similarity information, as they can learn similarity. When you select the desired method from *Method* combo box fields for that method's parameters will automatically show bellow the combo box. Since all methods are implemented in MATLAB they may require more time to provide results.

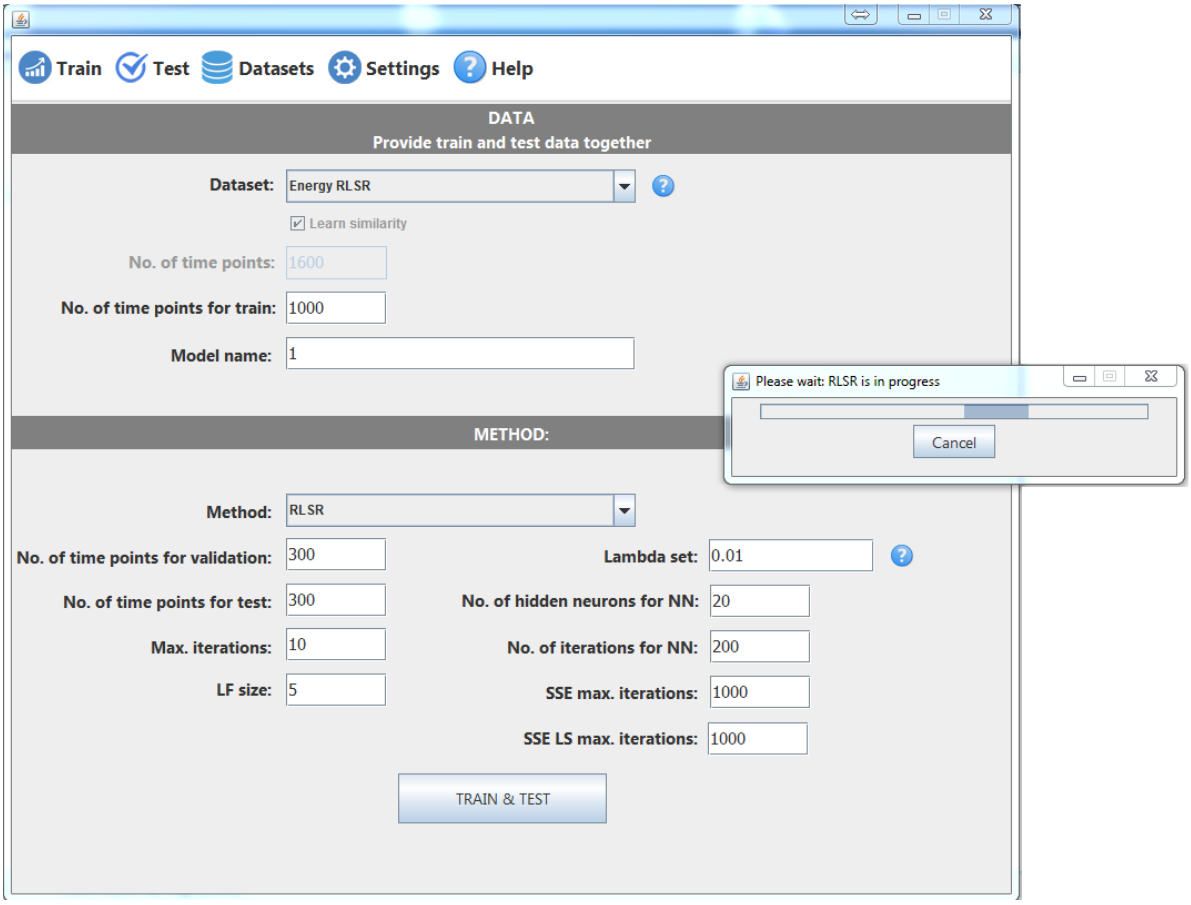


Figure 10. Train on temporal networks – Example

## 7. Train and test on random networks

The purpose of train and test on random networks options is to test the accuracy of DirGCRF method under controlled conditions on different types of directed graphs (Figure 11) and to compare it with the accuracy of standard GCRF.

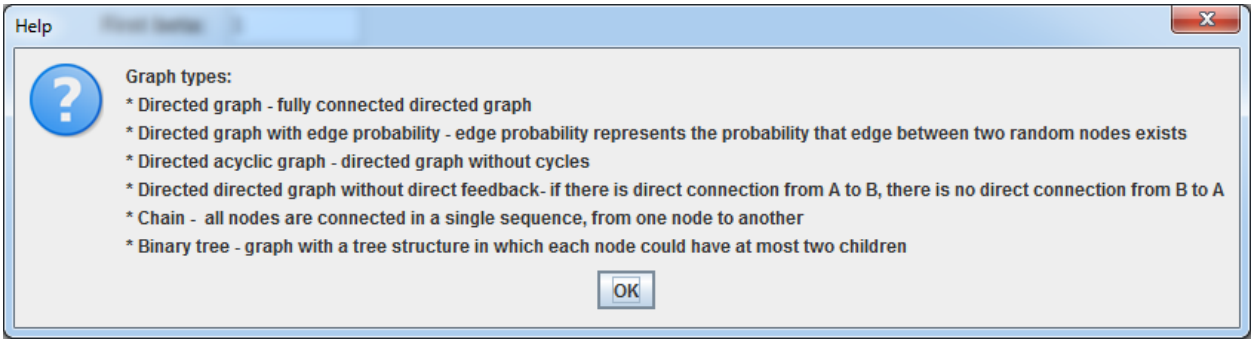


Figure 11. Different types of directed graphs

*Train on random networks* menu item (Figure 12) is used to train DirGCRF method (and GCRF if *Train symmetric* check box is checked) on randomly generated graph.

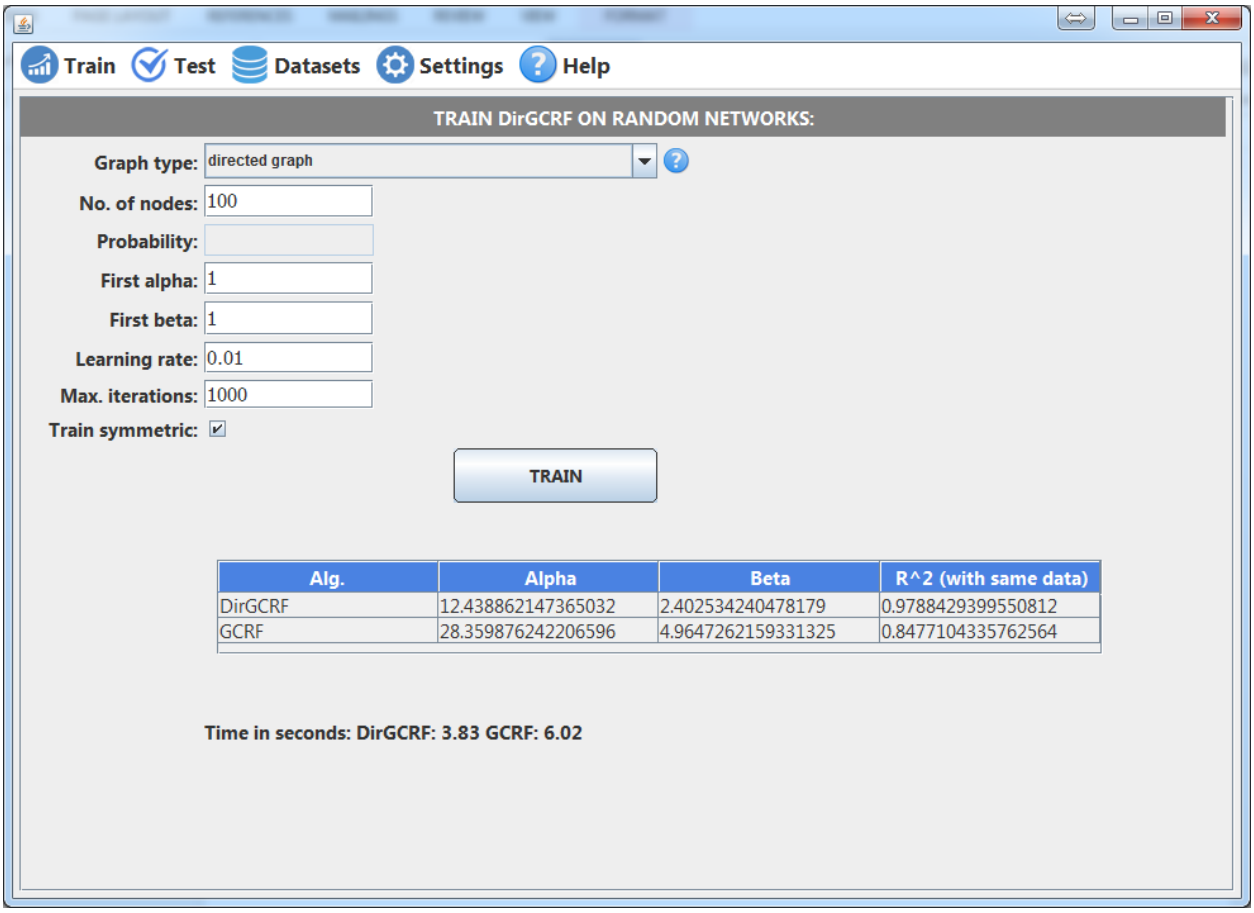


Figure 12. Train on random network – Example

*Test on random networks* menu item (Figure 13) is used to test trained models on randomly generated graphs. Number of graphs for testing process should be inserted in *No. of graphs* filed.

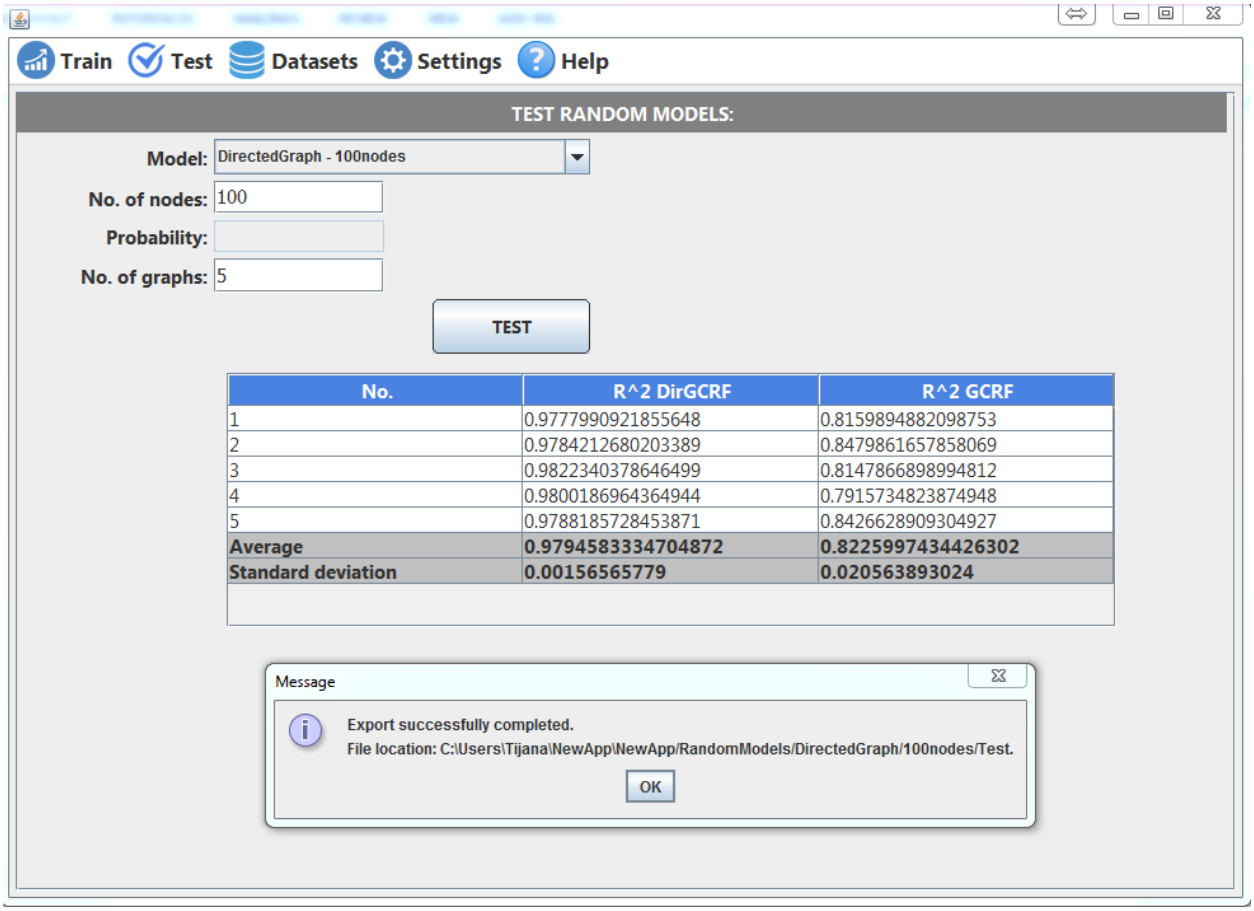


Figure 13.Test on random networks – example

For all models with randomly generated data new folder will be created in GCRF_GUI folder (example is presented at Figure 14).

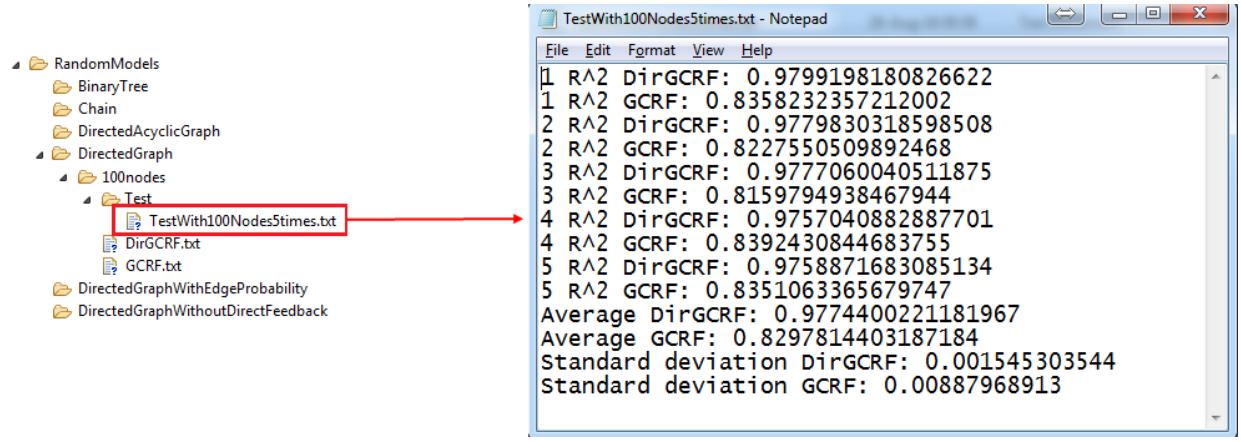


Figure 14. Example of the file with test results for directed graph with 100 nodes

## 8. Datasets

In Datasets -> Add dataset menu item new dataset can be added.

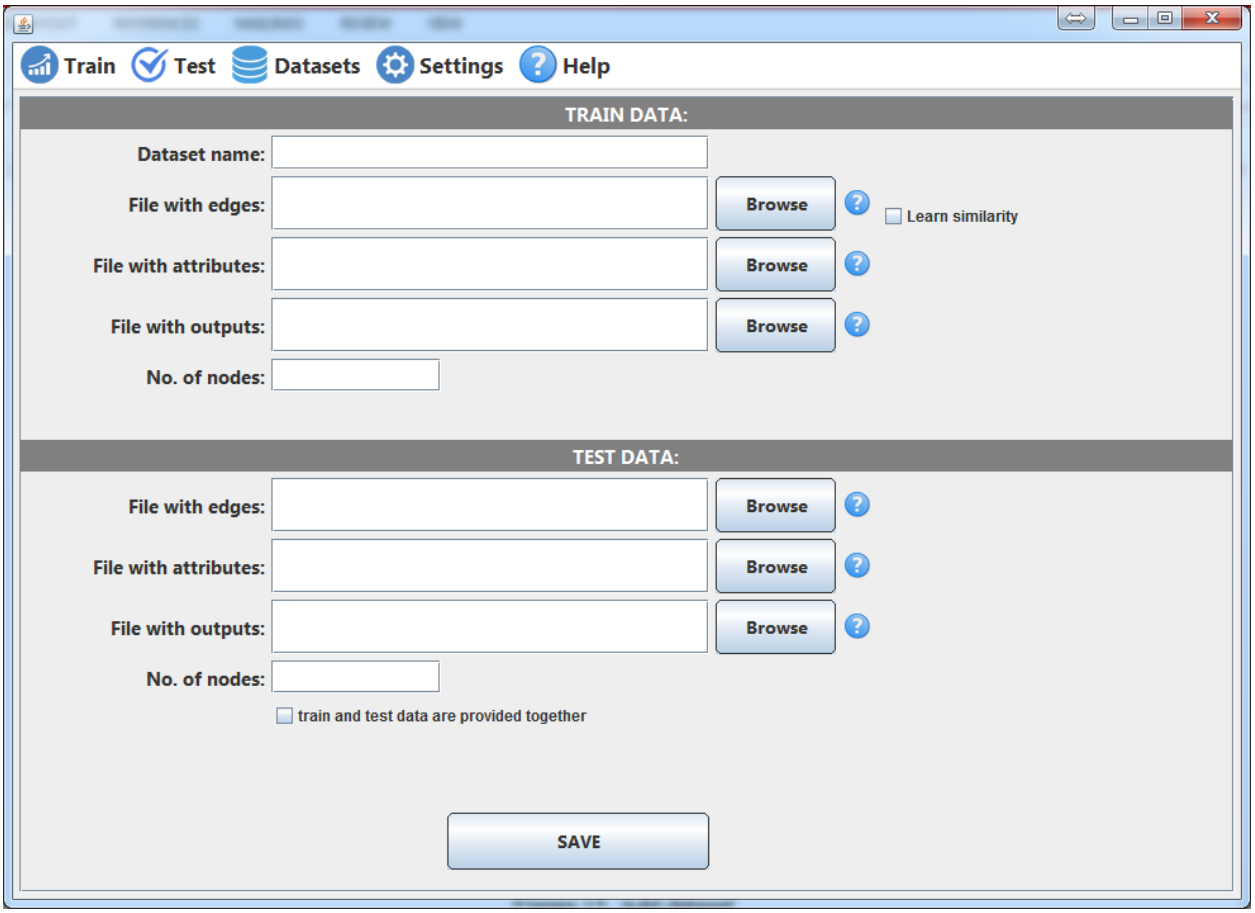


Figure 15. Add dataset

All datasets are listed in Datasets -> Manage datasets menu item. Each dataset can be deleted or renamed.

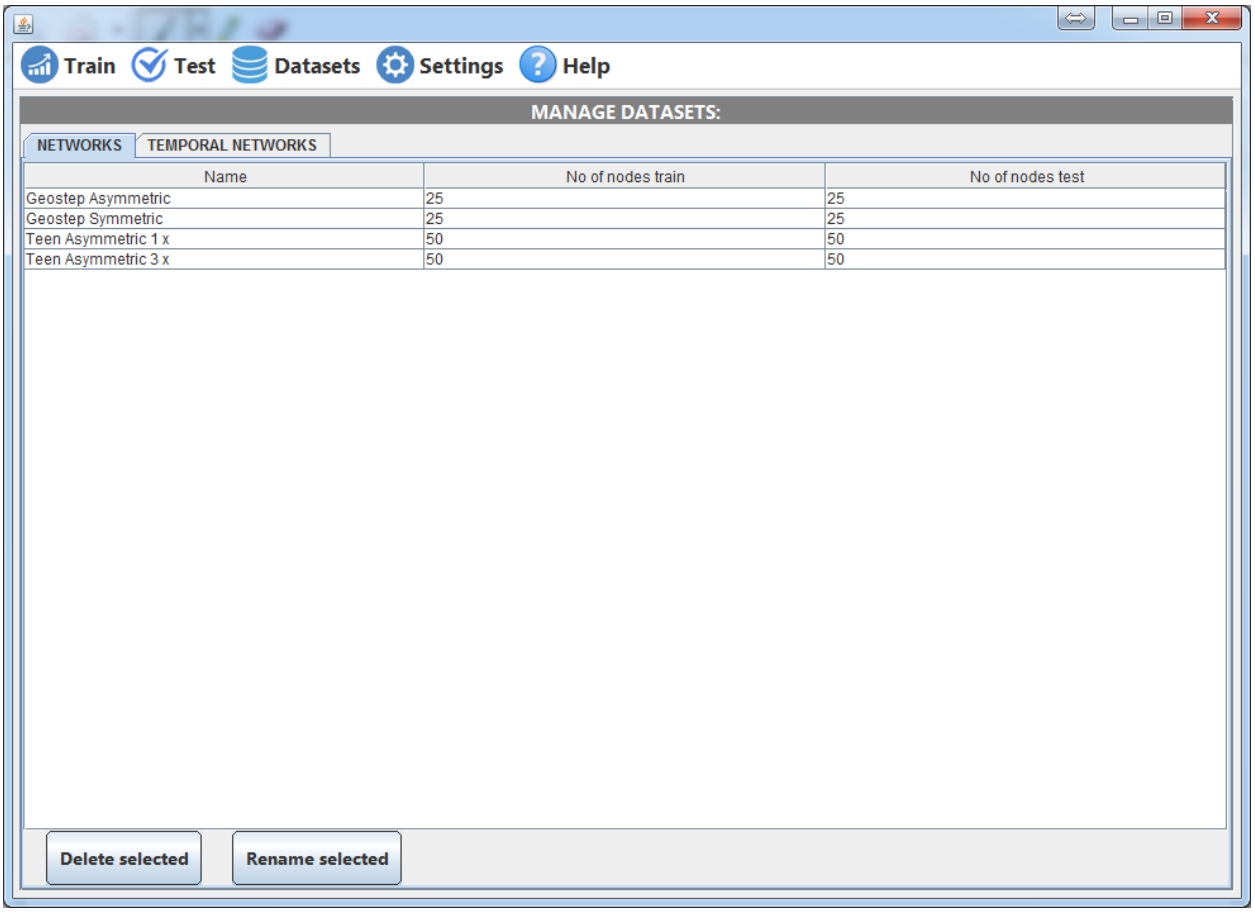


Figure 16. Manage datasets