Supplementary materials

New Features for Neuron Classification

Neuroinformatics

Leonardo A. Hernández-Pérez[1] · Duniel Delgado-Castillo[1] · Rainer Martín-Pérez[1] · Rubén Orozco-Morales[2] · Juan V. Lorenzo-Ginori[3]

[1] Empresa de Telecomunicaciones de Cuba S.A, Villa Clara, Cuba.

[2] Dpto. Automática y Sistemas Computacionales, Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Villa Clara 54830, Cuba.

[3] Informatics Research Center, Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Villa Clara 54830, Cuba.

Corresponding author:

Leonardo A. Hernández-Pérez
leonardo.hernandez@etecsa.cu

Workflow utilized to reproduce the algorithm described in "New features for neurons classifications" paper.

The examples that are provided in this workflow are obtained using the Alzheimer (local projection) neurons set.

1. Download reconstructed neurons from neuromorpho.org

- http://neuromorpho.org/NeuroMorpho_Linkout.jsp?PMID=12902394 Alzheimer disease (Local and Long projections) set.
- http://neuromorpho.org/NeuroMorpho_Linkout.jsp?PMID=23325800 Ischemia set.
- http://neuromorpho.org/NeuroMorpho_Linkout.jsp?PMID=22628459 Epilepsy set.

2. Obtain time series using Trees Toolbox.
    - Download Trees Toolbox from www.treestoolbox.org
    - Copy the folder Add_Trees_toolbox to Trees Toolbox root directory.
    - Use the function "cellcelloftree.m" to obtain the ".mtr" file with the neurons trees split into two groups. See "AlzheimerLocal.mtr".
    - Run the functions cs_wj.m, cs_woj.m, ss_wj.m and ss_woj.m to obtain the time series. See "Time series to load in MATS"

3. Obtain the dataset's features using the Measures of Analysis of Time Series toolkit (MATS). All MATS operations should be done running the graphical user interface (GUI).
    - Download MATS from http://eeganalysis.web.auth.gr/
    - Copy to the MATS root directory. "MeasureParam" file, available in "Examples files" directory.
    - Run the GUI:
        - Load time series.
        - Select / run measures.
    - Load Measure Parameters (Select "MeasureParam" file copy previously)
    - Run.
        - View measures.
            - Table of measures.
    - Copy the "Table of Measures" result to Microsoft Excel, add the class column and save as ".csv". See example "TableofMeasures(SS-WOJ).csv" in "Examples files" directory.
    - Open ".csv" file with measures obtained in Weka and save it as ".arff". See example "TableofMeasures(SS-WOJ).arff ".
4. Feature selection and classification using the Weka interface of MATLAB, available in: http://www.mathworks.com/matlabcentral/fileexchange/21204-matlab-weka-interface . This interface links MATLAB and Weka to automatize the process of feature selection and classification.
    - Download matlab2weka interface.
    - Rename the functions the functions: "wekaClassification.m" and "wekaFeatureSelection.m". For example "wekaClassification0riginal.m"
    - Copy from "Add-matlab2wekan folder, the functions: "wekaClassification.m" and "wekaFeatureSelection.m" to matlab2weka directory placed inside matlab2weka root directory.

- Copy to matlab2weka root directory the functions: cellcelloftree.m, loaddata.m, eval_c1assif.m and select_class_experimenter.
- Run:
  - loaddata.m to select the ".arff" dataset.
  - eval-c1assif.m to configure the parameter selection.
  - select_class_experimente for the features selection and classification process.
- Saved the result as ".csv" file during the classification process in the matlab2weka root directory. See "Classification result example.csv".
- To implement the alternate method the wekalab toolbox (McCarthy 2017) was used, from which the following functions were employed:
  - 1. "WEKAAPPLYFILTER", used to divide the dataset in training and test subsets with a proportion 2/3 – 1/3.
  - 2. "WEKATRAINMODEL" to create the model using the training subset.
  - 3. "WEKACLASSIFYMODEL" to classify the test set using the previously created model.
  - These three functions were used together with the previous implementation of the procedure published in (Hernandez-Pérez et al, 2017) using a new function called ¨SELEC_CLASS_TRAIN_TEST¨. When using the alternate procedure, the function ¨SELECT_CLASS_EXPERIMENTER¨ must be substituted by this new function.

5. Obtain the morphological features using L-measure.
   - Download L-measure from http://krasnow.gmu.edu/cn3 .
   - Select all morphometric functions from the "Function Panel".
   - Add the neuron file(s) to be analyzed on the "Input Panel".
   - Write the "result.csv" on the "Output Panel" to redirect the results of the analysis to "result.csv" file.
   - The Go Button on the "Go Panel" starts the analysis. The results is saved in the "result.csv" file automatically. The "result.csv" file contains 215 morphometric features. See "result.csv" file example.
   - Convert the "result.csv" file to "result.xlsx" file using Microsoft Excel.
   - Transform the result.xlsx file using the Microsoft Excel macro "MacroMorpho.xlsm". The morphological features dataset is obtained with this macro. "MacroMorpho.xlsm" is available at the "Examples files" directory.
   - Add the column class to the morphological features dataset and save it as ".csv" file. See example "MORPHO_Dataset.csv".
   - Open the ".csv" file with the morphological features in Weka and save it as ".arff", see example "MORPHO_Dataset.arff".
   - Repeat step 4.

Notice: Before run this algorithm, see the comprehensive example script "class_example.m" available in matlab2weka root directory.