LSEx.m User Manual

version: 1.0

January 2017

LSEx.m was designed by Michael Lindner and Doug Saddy and programmed by Michael Lindner

l-s-ex@gmx.co.uk

University of Reading, United Kingdom

Center for Integrative Neuroscience and Neurodynamics

https://www.reading.ac.uk/cinn/cinn-home.aspx

Content

1. Installation

- Copy the files onto a folder on your hard drive. e.g. in [C:\LSEx](C:\LSEx)
- Add the path to the files to the Matlab path by either using the „Set Path" button in the GUI or using the „addpath" function in the command line. e.g. `addpath('`[`C:\LSEx`](C:\LSEx)`')`

2. Dependencies

LSEx.m is a Matlab based toolbox. Therefore, for the usage of LSEx you need to have Matlab installed:

- Matlab 8.5.1 (R2015a) or newer (www.mathworks.com)

3. License

LSEx.m is licensed under a creative commons Attribution 4.0 international license (**CC BY 4.0: https://creativecommons.org/licenses/by/4.0**/)

4. Lindenmayer System Explorer

This tool is designed to generate and/or modify Lindenmayer systems (grammars).

First steps:

- Start the Lindenmayer System Explorer either by
  - typing in `run LSEx.m` in the Matlab command window, or
  - select the LSEx.m in the Matlabs current folder window and press F9, or
  - open LSEx.m file in the editor and press the start button.
- From here you will have two choices (via two buttons):
  - Lindenmayer System Generator
    - Using the Generator you can generate predefined Lindenmayer systems (e.g. Fibonacci) or you can define your own rules to create grammar. See detailed description in section 5.
  - Lindenmayer System Modifier
    - Using the Modifier you can change and modify existing L-systems, by building rules to iteratively replace chains of n characters of the system with chains of m other ones. See detailed description in section 6.

Enjoy playing around with this tool!

5. Lindenmayer System Generator

This tool is designed to generate predefined Lindenmayer systems (e.g. Fibonacci). But also user defined rules can be specified to create new L-systems.

Usage:

Select L-system

You can select a predefined system (e.g. Fibonacci), specify new
rule or load user defined rules:

- predefined rules.
  - The rules for the predefined system will be presented in the fields under the drop-down menu after making your choice.
- User defined rules
  - You can also define a new grammar by selecting User defined. Here you have two different options:
    1. The *classical version* where you can specify rule for replacing one character with a specific amount of other characters.
    2. An *extended version* where you can specify rule for replacing m number of characters with a specific amount of other n number of characters.

    You can make your choice by pressing the appropriate button.

    In both cases a new window will open to specify the number of rules (replacements). A window, where you can specify the rules will then open. You can save the defined rules by pressing the "Save rules" button. Then you can specify a name for the rule and it will be saved in a file called generator_user_rules.mat. The saved rules will appear in the drop down menu with the prefix USER. The rules can be deleted by deleting the generator_user_rules.mat file. BUT BE AWARE that all rules will be deleted by doing this!'

    In case of the *extended* replacement you can further select the type of replacement which should be performed:

    You have three options for types of replacement (how the rules should be applied to the system):

    a. segmentwise

        In the segmentwise replacement all vocabulars need to have the same length.

(e.g. 101, 111, 000, etc,). The L-system is then cut into segments of the same length and the replacement is then performed for each segment.

b. continuously

In the continuous replacement the algorithm goes through the whole grammar elementwise and replaces the n elements from the vocabulary with the m elements that you defined! Each element will be checked and can be used more than once for a replacement!

e.g. system: 00111011001001110010011110000

    rule: 111-->2 ; 110-->3 ; 100-->4

    results: 0023103400400223400400223400000

c. continuously (skip last n)

This type of replacement follows the same protocol as the one above, but when identifying a match with your rule the next n-1 characters (length of the vocabular) of the system are skipped. Here, each element will only be used once for a replacement!

e.g. system: 00111011001001110010011110000

    rule: 111-->2 ; 110-->3 ; 100-->4

    results: 0020304230423000

Start value:

The start value for the L-system generation is by default: 0 for the predefined systems and for the user defined system it is the value (lefthand side) of the first rule!

But the start vlaue can be changes changed individually.

IMPORTANT: The values in the "start values" field need to be included within the rules you define!

Number of recursions:

With the number of recursions you can specify how often the rules should be used iteratively.

Output folder:

You need to specify an output folder by typing in a path or by using the browse button.

Output file prefix:

You can specify an output file prefix.

Output file type:

You can choose the type of output which should be stored. Either a Matlab .mat file, a text file, or both are valid:

    In the case of a:

- .mat file, three variables are stored: a cell with the grammar of each iteration, the rule and a vector with the length of each iteration.
- .txt file, three output files are stored. One with the grammar, one including the length of each iteration and one containing the rule.

Generate L-system:

After setting up all parameters, you simply need to press the generate L-system button.

The L-system will be generated and saved in the output files. Additionally, the L-system will be shown in the command window of Matlab as well.

6. Lindenmayer System Modifier

This tool is designed to modify Lindenmayer systems (grammar). With this tool, rules can be defined to replace n with m number of elements in the system.

 e.g. 111 with 1

    10 with 2

    110 with 10

    1   with 1011 etc.

Usage:

Load L-System:

First you need to load a system by using the Load L-system button. You can easily load a Matlab output file from the Lindenmayer System Generator. You can also load other systems, as long as the system is stored in a matlab .mat file as a cell!

Define rules:

You can define any rules for replacement by pressing the "Define rules" button. After pressing the button a new window opens where you can specify the number of rules (replacements). Afterwards another window opens, where you can specify the rules

 e.g. 101 ---> 1

    010 ---> 0 etc.

You can save the defined rules by pressing the "Save rules" button. Then you can specify a name for the rule and it will be saved in a file called modifier_user_rules.mat. After creating this file by saving one or more rules, you will get the option to load the rules after pressing the "Define Rules" button. The rules can be deleted by deleting the modifier_user_rules.mat file. BUT BE AWARE that all rules will be deleted by doing this!

You have three options for types of replacement (how the rules should be applied to the system):
1. segmentwise

    In the segmentwise replacement all vocabulars (left side of the rules, which should be replaced) need to have the same length. (e.g. 101, 111, 000, etc,). The grammar is then cut into segments of the same length and the replacement is then performed for each segment.

2. continuously

   In the continuous replacement the algorithm goes through the whole grammar elementwise and replaces the n elements from the vocabulary with the m elements that you defined! Each element will be checked and can be used more than once for a replacement!

   e.g. system: 001110110010011110010011110000

      rule: 111-->2 ; 110-->3 ; 100-->4

      results: 002310340040022340040022340000

3. continuously (skip last n)

   This type of replacement follows the same protocol as the one above, but when identifying a match with your rule the next n-1 characters (length of the vocabular) of the system are skipped. Here, each element will only be used once for a replacement!

   e.g. system: 001110110010011110010011110000

      rule: 111-->2 ; 110-->3 ; 100-->4

      results: 0020304230423000

Number of recursions:

With the number of recursions you can specify how often the rules should be used iteratively.

Output folder:

You need to specify an output folder by typing in a path or by using the browse button.

Output file prefix:

You can specify an output file prefix

Output file type:

You can choose the type of output which should be stored. Either a Matlab .mat file, a text file, or both are valid:

   In the case of a:

   • .mat file, three variables are stored: a cell with the grammar of each iteration, the rule and a vector with the length of each iteration.

   • .txt file, three output files are stored. One with the grammar, one including the length of each iteration and one containing the rule.

Modify L-system:

After setting up all parameters, you simply need to press the Modify L-system button.

The loaded L-system will be modified and saved in the output files. Additionally, the L-system will be shown in the command window of Matlab as well.