Adaptive Rewiring on Logistic Maps with Heterogeneous Parameters

Author Name(s), First M. Last, Omit Titles and Degrees

Institutional Affiliation(s)

Author Note

Include any grant/funding information and a complete correspondence address.

Abstract

The abstract should be one paragraph of between 150 and 250 words. It is not indented. Section titles, such as the word Abstract above, are not considered headings so they don’t use bold heading format. Instead, use the Section Title style. This style automatically starts your section on a new page, so you don’t have to add page breaks. (To see your document with pagination, on the View tab, click Reading View.) Note that all text styles for this template are available on the Home tab of the ribbon, in the Styles gallery.

Keywords: Add keywords here. To replace this (or any) tip text with your own, just select it and then start typing. Don’t include space to the right or left of the characters in your selection.

Adaptive Rewiring on Logistic Maps with Heterogeneous Parameters

# Introduction

# Method

To answer the research question, we model binary networks of chaotic coupled oscillators with different parameterizations and let them evolve by means of adaptive rewiring algorithm. In this model, every node is characterized by two parameters, coupling strength and chaos, and a global rewiring algorithm lets the network structure to evolve. There have been five different sets of parameters, for which a total of 50 models (10 model per parameter set) are analyzed. In order to ease the discussion, we use metaphors to describe our models: each model is assumed to be a “brain” where the oscillators at the nodes are called “neurons”. Moreover, the brains have been assigned a randomly generated name, and each parameter set is called a “family”. Brains within a family have identical parameters and the only difference between family members is the random seed used to assign the initial values at their birth. In what follows, we go through the elements of our models: birth of the brains; characteristics of the oscillators; the rewiring algorithm; making of families; and the qualitative and quantitative measures of network structures. All the simulations and analyses are conducted in R programming language version 3.5.0 (R Core Team, 2018).

## Building the graph

In this study, building on previous literature (citation), we modeled an undirected graph with 300 nodes and 5200 unweighted edges. In mathematics, a graph is a set of 3-tuples that define the relation (called edge or connection) between pairs of objects and (called vertices or nodes). The set of edges and vertices are represented by E and V, respectively. The value of characterizes the strength of association between vertices and and hence is often called edge weight. Conventionally, the edges with zero values (i.e., no connection between the vertices) are omitted from the set E. If , the graph is called unweighted or binary. In the case where , the network is called an undirected network. The graph can be depicted graphically by circles and association between nodes can be represented graphically by circles as nodes and (directed) lines connecting them. A graph can be formally described by a set of 3-tuples, an edge list, or an adjacency matrix. Adjacency matrix is a square matrix of the size |V| where each element is the edge weight of the connection between the nodes i and j. If the graph is undirected, the matrix is symmetrical around its main diagonal, and if it is unweighted all the elements take values of zero or one.

The models (“brains”) simulated in this study are undirected graphs of |V| = 300 “neurons” and |E| = 5200 unweighted connections. The initial graph was constructed by randomly imputing matrices of zeros with ones, as such the matrix becomes symmetrical (undirected) and has zeros on diagonal (without any loops). The random numbers were generated based on a seed which depended on |V|, |E|, and a generation parameter `round`.

## Neurons and the coupled oscillators

The activities of the node (i.e., “neurons”) of each brain was modeled using coupled logistic maps. Logistic maps are a family of functions that can produce wherein the output only depends on a chaos parameter and the value the function at previous time step. The logistic maps used in this study is of the form shown in Equation 1.

EQ1

For certain regimes of the chaos parameter, the behavior of the logistic map converges to one or more attractors, but otherwise it exhibits chaotic behavior. In these regimes, logistic maps produce deterministic stochastic bounded time series which has been suggested as an approximation of electrical activities of spiking neurons (citation). The values of the 200 draws of logistic maps after a burn-in period of 4000 iterations can be seen in Feigenbaum plots shown in Figure 1. The parameters of the logistic maps in the current study were chosen in the range of 1.7-1.9, as it grants stochastic signals bounded in the range of [-1, 1] with properties suitable for the adaptive rewiring algorithm[[1]](#footnote-1). Each neuron in each brain was assigned a fixed value for \alpha, and the initial x\_1 values of activations were drawn uniformly from the range of [0, 1]. This random draw was based on the same seeds used in constructing the random graphs. Since the models are mathematically described (and simulated) by matrix algebra, the same notation will be used from now on.

The logistic maps at the core of the activities of neurons are coupled according to Equation 2 as such the activity of each neuron is moderated by the activity of its neighbors. In a matrix notation, for a network of |V| = N, the activity of neurons at time t is calculated via

EQ2

In the right-hand side of Equation 2, the vector form of logistic maps is Hadamard-multiplied[[2]](#footnote-2) by a coupling term. In the coupling term, is the vector of coupling strengths and is the connectivity matrix at time t. in the denominator is a vertical unit vector of size N. Division of by normalizes the former (i.e., the effect of of node i) by sum of the weights of the edges connected to each node. Since our graphs are binary, the term in the denominator counts the number of connections for each node.

## Adaptive rewiring of the network

At each rewiring attempt, the connections of a random node is optimized locally as such the node is disconnected from its most dissimilar neighbor and is connected to the most similar nodes to which it was not connected. The dissimilarity of two neurons at a given time is defined as the absolute value of the difference in the value of their activity. To do so, at time t, a node i is selected randomly from V, and a vector[[3]](#footnote-3) of its distance from other nodes is calculated as , and another vector of similarities is defined as .

The most dissimilar neighbor and the most similar non-neighbor of the neuron, denoted as and , respectively, are found by finding the index of the maxima of the following vectors:

The matrix multiplication of M and (1-M) with , respectively, ensures that the search for the edges subject to rewiring happens in the right subset of edges. The rewiring is then changing the corresponding elements of the adjacency matrix:

In this study, a rewiring attempt takes place after every 20 updates of the activities of the logistic maps. In this one million rewirings were done, meaning that each neuron fired 20 million times.

## Setting the parameters and making families of brains

As we have seen, each brain can be characterized by three constant parameters: `round`, which determines the seed used to generate the initial random graph and set the starting values of the logistic maps; chaos parameter for each node; and , the vector of coupling strengths of each neuron. Previous research (citation) and preliminary exploratory simulations suggest that adaptive rewiring on coupled logistic maps realize best for and in the ranges of [1.7-1.9] and [0.3-05], respectively. In this study, the midpoints of the ranges are assumed the baseline values for the parameters. The nodes with lower and upper values of the ranges of and are assumed to be hypo- and hyper-chaotic and hypo- and hyper-coupled.

To answer the research question, i.e., to what extend … (phrasing?), the neurons were assigned to two partitions of minority (the first 50 nodes) and majority (the remaining 250 nodes). While keeping the parameters of the majority at the baseline () five different combinations of parameters were assigned to the minorities, each combination called a “family”: The homogeneous family (), and the families with hypo-chaotic minority (), hyper-chaotic family (), hypo-coupled minorities (), and hyper-coupled minority ().

In each family, ten generation of brains are simulated to increase the robustness of the outcomes. The only difference between the generations are in the `round` parameters used to initialize the brains. The generations across families are matched in that parameter, i.e., the initial network structure and activation values of generation k are identical in all families. With this choice, it is possible to match brains starting with the same conditions to see the effect of different parameterizations more clearly.

## Qualitative and quantitative description of network structure

As mentioned earlier, each brain is characterized by the set of aforementioned parameters, which remain constant over the course of simulation. The state of the network at any given time t can be described by the connectivity matrix and the neuron activities, i.e., , respectively. Since this study seeks to answer questions about the structure of the networks under certain conditions (i.e., different parameterizations of and ), we focus here on describing the connectivities of the brains, and the evolution of these measures over time as the brains mature.

### Qualitative assessment of the structure.

The network structure can be qualitatively assessed by means of visual investigation of the graph diagram or by visualizing the adjacency matrix. Since the neurons belonging to a given partition are not distinguishable, the raw adjacency matrix can only give an estimate of the edge density within and between partitions but fails to provide intuitions into the structure of the network. Hence, the adjacency matrix is serialized using … algorithm, implemented in the package `seriation` (citation), which orders the rows and columns of the matrix to maximize visibility of modules within the network (better/more precise phrasing?). The raw (unserialized) and ordered (serialized) adjacency matrices, and the graph diagrams are plotted using `seriation` and `igraph` packages. In the plots, the minority and majority neurons are colored sky blue and pink, respectively. In both matrix visualizations and graph diagrams the within-minority and within-majority edges are colored blue and red, respectively, while the inter-partition edges (those connecting neurons of minority subset to neurons of majority) are colored green.

### Quantitative measures of the structure.

In network science, a wide range of structural measures of connectivity, also known as network (summary (?)) statistics, have been proposed. (citation) After each rewiring attempt, we calculate eight network connectivity measures for the whole network and three subsets of edges, namely, intra-minority, intra-majority, inter-partition. These measures are explained in what follows.

#### Clustering coefficient.

This measure gives an indication of the tendency of nodes to form clusters and can be defined either locally or globally. We use global clustering coefficient, which is defined as the number of closed triplets of nodes (the triplets of that are all connected) divided by the number of connected triplets, either open (paths of length 2) or closed (triangles). The numerator is equal to three times the number of triangles in the graph, and can be calculated formally from the adjacency matrix as follows

EQ 3

#### Modularity.

Modularity of a graph, as proposed by Newman (2006) and denoted by Q, is a measure of how (and to what degree), for a certain labeling of nodes, the nodes tend to form communities with the nodes of the same label and tend to not connect to other nodes of the graph. This measure requires a priori labeling of nodes that defines the communities to which they are believed to belong. The labeling can be done manually based on theoretical knowledge or arbitrary decisions.

There has been a line of research on algorithmically discovery of optimal modules (also known as communities) within graphs such that the measures of modularity is maximized. (for a review of the proposed methods, cf. Zhang, Ma, Zhang, Sun, & Yan, 2018). The communities discovered by these algorithms can be used as labels for calculating modularity of the network. In this study, in line with (Clauset, Newman, & Moore, 2004), using `igraph` package, the fast greedy algorithm is used to optimally detect communities and calculate the modularity based on the community membership of nodes.

#### Average path length.

Average path length is the mean value of lengths of shortest path between all pairs of nodes. This measure, calculated using `igraph` package, gives an indication of how closely the nodes of a network are located from each other.

#### Efficiency.

The efficiency of a graph quantifies the efficiency of information exchange within the network, and is defined as sum of inverses of the distances between nodes. In order to make the measure comparable across graphs of different size, the average efficiency is often used which is defined in Equation 4.

EQ 4

Where N is the size of the network and are the distances between unidentical nodes I and j. The denominator is twice the number of all possible edges in a graph of size N.

#### Small-worldness.

Small-worldness is a measure of the degree to which the graph shows properties similar to the structures known as small world (citation). It is defined as the multiplication of normalized clustering coefficient and efficiency of the network, i.e.. are the expected clustering coefficient and efficiency of a random graph of the same as the graph in question. For computational reasons, in this study, a non-normalized version of small-worldness coefficient is calculated and reported.

#### Assortativity.

Assortativity coefficient indicates the preferences of nodes to connect to “similar” nodes by summarizes the probability of connections between the similar nodes. The similarity is can be imposed externally, e.g., by assigning categories to the nodes using labels (known as nominal assortativity), or by internal criteria like the node degrees (degree assortativity). Degree assortativity, calculated using `igraph` package in this study, measures the probability that nodes of similar degree (i.e., number of connections) are connected. (citation?)

#### Rich Club coefficient.

This coefficient quantifies the tendency of nodes with higher than certain degrees to connect to each other, and is calculated using `brainGraph` package (citation). More formally, it is equivalent to the edge density (see below) of the subgraph of the main graph where the number of edges in graph where the nodes with lower degrees than the cut-off value k are removed:

EQ 5

#### Edge density.

For a subset of edges, this coefficient is the proportion of existing edges to the maximum number of edges possible in that subset. It is used to quantify the normalized density of edges labeled as within-minority, within-majority, and inter-partition, as well as the whole network. Since the total number of edges remains the same during the adaptive rewiring, this coefficient gives an indication of how strongly each partition has attracted the new nodes into itself at every rewiring step. One can visually estimate the value of this coefficient by the density of each color in the subsets of the unserialized adjacency matrix.

# Results

# Discussion

References

Last Name, F. M. (Year). Article Title. *Journal Title*, Pages From - To.

Last Name, F. M. (Year). *Book Title.* City Name: Publisher Name

Footnotes

1Add footnotes, if any, on their own page following references. The body of a footnote, such as this example, uses the Normal text style. (Note: If you delete this sample footnote, don’t forget to delete its in-text reference as well. That’s at the end of the sample Heading 2 paragraph on the first page of body content in this template.)

Tables

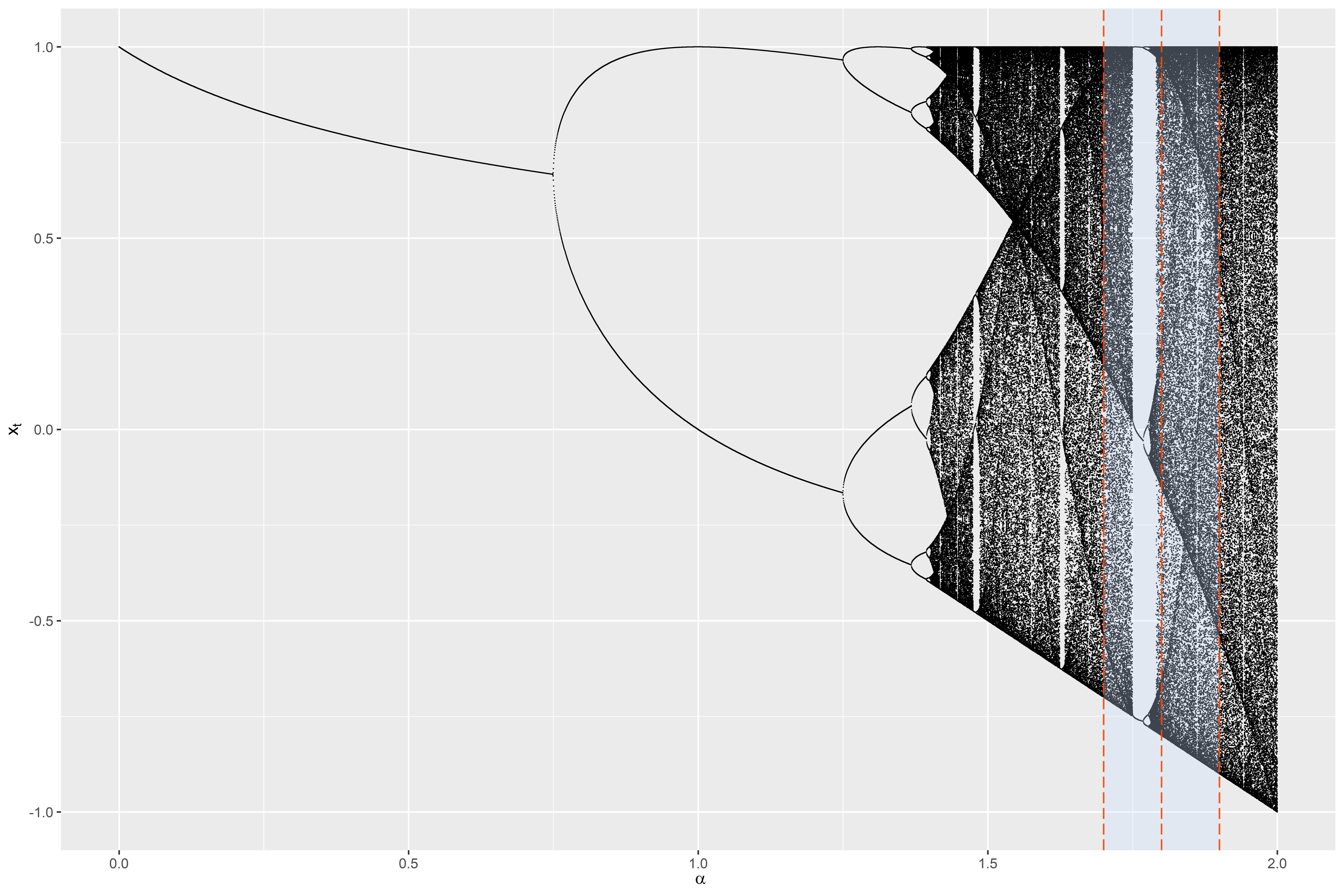
Table 1

Table Title

| Column Head | Column Head | Column Head | Column Head | Column Head |
| --- | --- | --- | --- | --- |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |

Note: Place all tables for your paper in a tables section, following references (and, if applicable, footnotes). Start a new page for each table, include a table number and table title for each, as shown on this page. All explanatory text appears in a table note that follows the table, such as this one. Use the Table/Figure style, available on the Home tab, in the Styles gallery, to get the spacing between table and note. Tables in APA format can use single or 1.5-line spacing. Include a heading for every row and column, even if the content seems obvious. A table style has been setup for this template that fits APA guidelines. To insert a table, on the Insert tab, click Table.

Figures



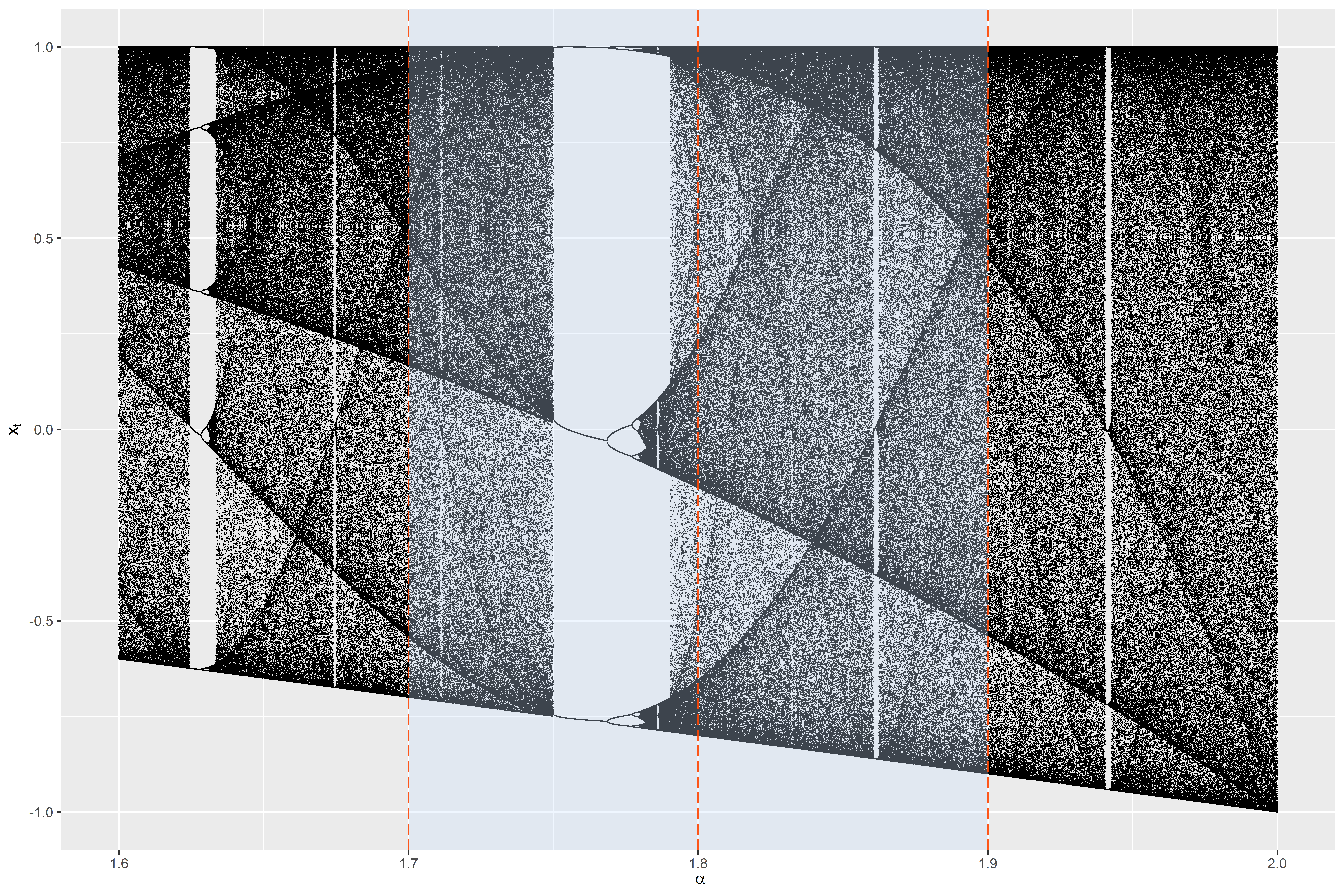


Figure 1. Feigenbaum diagrams.

1. The values were chosen in line with (citation), and the range has been narrowed down by exploratory inspection of preliminary outcomes of the simulations. [↑](#footnote-ref-1)
2. Also known as elementwise multiplication of matrices where the corresponding elements of matrices are multiplied. [↑](#footnote-ref-2)
3. While programming the analyses, a matrix of distances was calculated to increase the versatility of the code for other rewiring algorithms. [↑](#footnote-ref-3)