Human Brain Project

SynapsesSA

User Reference Manual

Version 1.0

Luis Rodriguez-Lujan luis.rodriguezl@alumnos.upm.es
Laura Antón-Sánchez l.anton-sanchez@upm.es
Pedro Larrañaga pedro.larranaga@fi.upm.es
Concha Bielza mcbielza@fi.upm.es

last update 26th March 2015

Contents

1 Introduction

This guide contains all essential information to make full use of the tool *synapsesSA* and its functionality, and a highly detailed walkthrough about the user interface. For further information about spatial statistics, its application to neuroscience and deeper insight into *synapsesSA*, please refer to *Anton-Sanchez et al.* [2014].

The document is organized as follows:

- This first section defines the basic concepts used in *synapsesSA*, the tool itself and its technical requirements
- The getting started section explains how to install the R package and run it for first time, describing all fundamental commands needed to complete the process
- In the third section, required cortical synapses data is defined, specifying the structure and content of every accepted data source
- Graphical user interface (GUI), the fourth section, reviews the web interface supplied with the tool section by section, from how to load data to the generation of simulated samples
- Common user cases, the last section, provide step-by-step instructions to use the GUI in several scenarios

1.1 What is *synapsesSA*?

SynapsesSA is a tool designed to process and analyze patterns in the three-dimensional spatial distribution of cortical synapses. It brings a variety of both innovative and well-known techniques from the spatial statistics field and a web-based graphical interface compatible with most common browsers.

You should read this document if you are a neuroscientist or data modeller who wants to:

- Process and visualize data from cortical synapses for error checking
- Model the spatial distribution of the synapses
- Replicate, via simulation, samples of cortical synapses
- Compare several indicators obtained from data of different layers

1.2 Requirements

In any case you will need:

- An installed R environment, version 3.1.0 or later (3.1.3 is recommended). Earlier versions are not guaranteed to work properly
- Internet access: Source code and auxiliary libraries are downloaded from internet repositories
- For *Windows* users: An installed R-tools environment (available on the official web page: `http://cran.r-project.org/bin/windows/Rtools/`)
- For *Linux* and *MAC* users: build-essential (or functionally equivalent) package installed, and the cURL client

1.3 Basic concepts

The aim of this tool is to explore the three-dimensional distribution of synapses in the cerebral cortex and find out any possible distribution pattern of synapses for the provided data, and to create simulations based on real data. This also includes identifying similarities and differences between samples from different cortical layers.

Assuming some neuroscience background of the reader, this section focuses on the definition of some basic spatial statistics concepts. Once again, for more information please refer to *Anton-Sanchez et al.* [2014].

A complete spatial randomness (CSR) or homogeneous spatial Poisson point process is a spatial distribution where every point is equally likely to occur at any location within the domain space, regardless of the locations of other points (that is, without any restriction).

Similarly, a Random Sequential Adsorption (RSA) process is also a spatial distribution where points are distributed in the domain space almost randomly, with the only constraint that each point has an associated surrounding volume that cannot overlap with others.

Finally, summary functions in spatial statistics are a set of widely used indicators to characterize spatial distributions. *synapsesSA* uses four of the summary characteristics most commonly applied in the analysis of spatial point processes:

- The ***F*** function measures the distribution of all inter-point distances from an arbitrary point; in other words, it measures the empty space between points
- The ***G*** function examine the cumulative frequency distribution of nearest neighbor distances in the sample
- The ***Ripley's K*** function for a distance d is defined as the expected number of other points within a distance d of a typical point of the process divided by the intensity/density *Ripley* [1977]
- ***Besag's L*** function is a transformation of *Ripley's K* function. The 3D expression is:

  $$L(d) = \sqrt[3]{\frac{3}{4\pi}K(d)}$$

  As a particular case, the L function for the CSR process is a straight line , making the plot interpretation easier

2 Getting started

In the first step, *synapsesSA* will be downloaded and installed in your machine. After that no configuration is needed and the initialization process is straightforward.

2.1 Installing from the repository

The following installation process relies on a R repository maintained by the CIG.In the future the package will be hosted at the HBP Unified Portal and the installation process may vary.

First, open an R terminal and then execute the command below to add the CIG-R repository to the current repository list. A prompt will ask you to select which repositories should be added; select the options that correspond to CRAN and CRAN(extras) repositories.

```
setRepositories(addURLs=
  c(cigRepo="http://boadilla.dia.fi.upm.es/R"))
```

Then, you just need to install *synapsesSA* as any other regular R package that comes from the CRAN repository.

```
install.packages("synapsesSA")
```

Windows users might experience some problems during the installation. In that case, execute the next command instead of the previous one.

```
install.packages("synapsesSA",type="source")
```

If still having problems with missing dependencies (i.e. *car* or *shinyRGL* packages) install those packages independently executing the following command.

```
install.packages("PACKAGENAME")
```

Once the install process has finished without errors, *synapsesSA* is installed in your R environment. Then, you can proceed to the next section to run the GUI.

2.2 First run

The first step to run the tool is to load the library in the current R workspace and launch the user interface. To do so, just execute the commands below.

```
library(synapsesSA)
run_SynapsesSA_UI()
```

System's default web browser should be launched automatically after the app is started. If this is not the case, look for a line like *Listening on http://127.0.0.1:5811* in the R terminal and copy the URL to your browser.

synapsesSA is structured as a regular R library with a graphical user, based on *shiny*, an embedded web-interface. The package is intended to be used through the web-interface, but every single function included in the package has been documented and additional technical information is available. For example, the command below loads the help page for the *nRSAsimulations* function.

```
?nRSAsimulations
```

3 Input data

synapsesSA has been designed to work with spatial data from cortical synapses. For each synapse, synapse centroid 3D coordinates (x,y,z) and its Feret's diameter must be provided. Every input measure should be given in nanometers.

In addition, in order to identify the sample and layer of each synapse unambiguously, this information sample number and layer) has also to be included. Please note that one sample can contain synapses from several layers, but each layer from a single sample will be processed independently as if it come from different samples.

Currently, only CSV files are supported as input format, but in the near future *synapsesSA* will integrate other HBP scientific databases. For example, the ESPINA software output file is easily loadable.

3.1 CSV file

A character-separated values (CSV) file stores plain-text data in rows and columns, like a table. *synapsesSA* accepts CSV files with commas, semicolons or tabulators as field separators and dots or commas as decimal points. A header is required but column order is not fixed. These CSV files can be created in any spreadsheet editor such as Excel or LibreOffice Calc just by saving the current spreadsheet (fig. 1) in CSV format (fig. 2).

	A	B	C	D	E	F	G	H
1	**Sample**	**Sample no.**	**Type**	**Feret**	**X**	**Y**	**Z**	**Layer**
2	syn1	4	Asymmetric	501,909245	1745,74042	1,4956	3971,6586	III
3	syn2	4	Asymmetric	748,049725	192,23	3842,43106	3835,9586	IV
4	syn3	4	Asymmetric	12,8282	2009,49928	3458,14404	3971,87872	V
5	syn4	4	Symmetric	38,82827	510,20251	1136,47607	3893,44394	III

Figure 1: Sample file in a spreadsheet editor

The header is mandatory and must be in the first line of the CSV file, assigning the following names to each column:

- **Layer**: Roman number that identifies the cortical layer of the sample *(Required)*
- **Sample.no.**: Number that identifies the sample. Each sample should have a unique ID *(Required)*
- **Type**: Type of synapse; Symmetric or Asymmetric *(Optional)*
- **Feret**: Synapse Feret's diameter in nanometers *(Required)*
- **X**: Synapse centroid X coordinate in nanometers *(Required)*
- **Y**: Synapse centroid Y coordinate in nanometers *(Required)*
- **Z**: Synapse centroid Z coordinate in nanometers *(Required)*

Additional columns can be included, but will be ignored during the loading process.

```
1 Sample;Sample no.;Type;Feret;X;Y;Z;Layer
2 syn1;4;Asymmetric;501,9092446;1745,740422;1,4956;3971,6586;III
3 syn2;4;Asymmetric;748,0497245;192,23;3842,431064;3835,958597;IV
4 syn3;4;Asymmetric;12,8282;2009,499279;3458,144044;3971,878721;V
5 syn4;4;Symmetric;38,82827;510,2025095;1136,476071;3893,443939;III
```

Figure 2: CSV file

4 Graphical user interface

A reactive web-based user interface is implemented in *synapsesSA* in order to ease the usage of the package for non-expert users. The interface is divided in four sections:

- Data loader
- Model builder
- Model simulator
- Layers comparison

The application layout is similar across all sections (fig. 3):

- A header with the application name and official logos (**1**)
- The Computational Intelligence group logo that links to the contact page **(2)**
- The navigation bar, with buttons to access each section of the web page **(3)**
- The work space *(4)*, divided in two subsections:
  - A side control panel, with selectors and buttons varying from tab to tab **(4.a)**
  - The main space, where plots, tables, etc. are drawn. Any error or warning message will be printed on the top of this section **(4.b)**

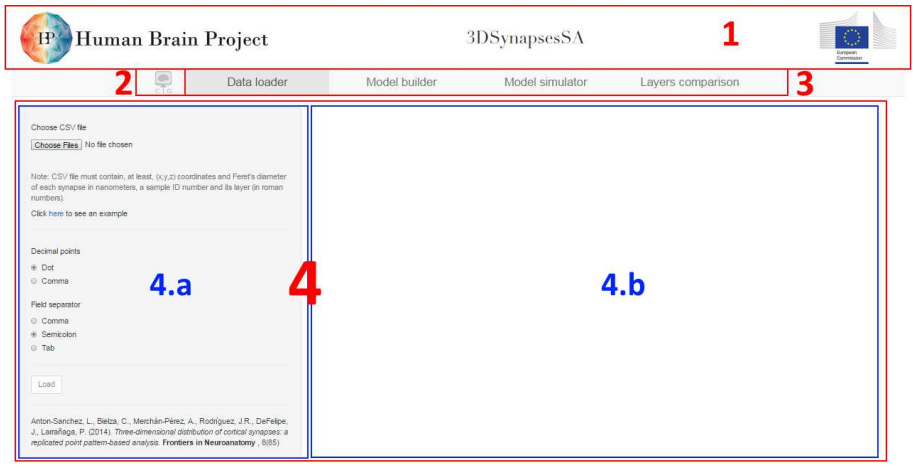


Figure 3: GUI structure

4.1 File load tab

The file load tab allows the user to upload one or more CSV files to the application and preview them. Several sets of files can be uploaded sequentially, attaching their content to the samples previously loaded in the same session. Once a file is uploaded, it cannot be removed. To start a new session, just refresh the web page.

The control panel (fig. 4) in the load tab includes:

- A multiple file selector that accepts CSV files
- A decimal point selector
- A field separator selector
- The load button that activates once a file is uploaded through the file selector

In the main panel, once valid data have been loaded, two tabs are displayed:

Figure 4: Load tab control panel

- The first one is a data table that supports column ordering, pagination and field search. By reviewing this table, the user can check if loaded data matches to the original file (fig. 5)

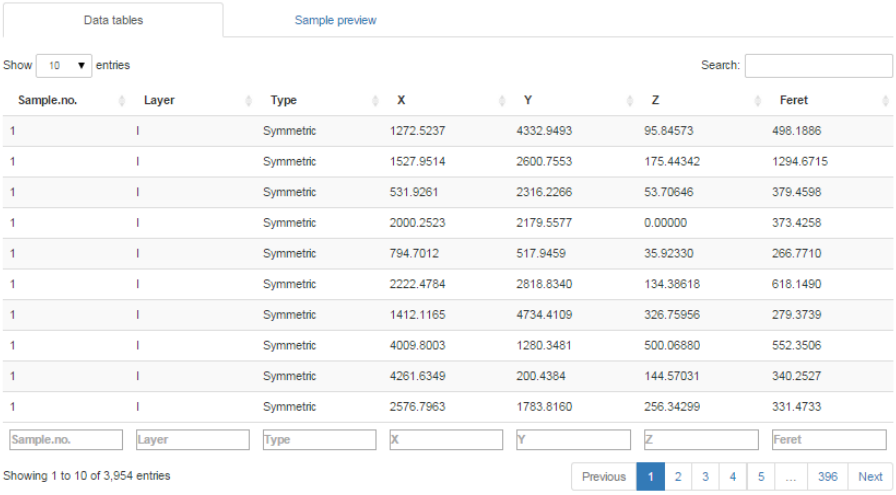


Figure 5: Loaded data table

- Next tab shows a 3D preview, that supports zoom and rotation, of each loaded sample. Each synapse is depicted as a sphere, using the Feret's diameter as the spherical diameter. Green spheres represent symmetric synapses, while red ones are asymmetric synapses. By using this preview, the user can visualize the original data and detect errors, for example looking for overlapping synapses. (fig. 6)

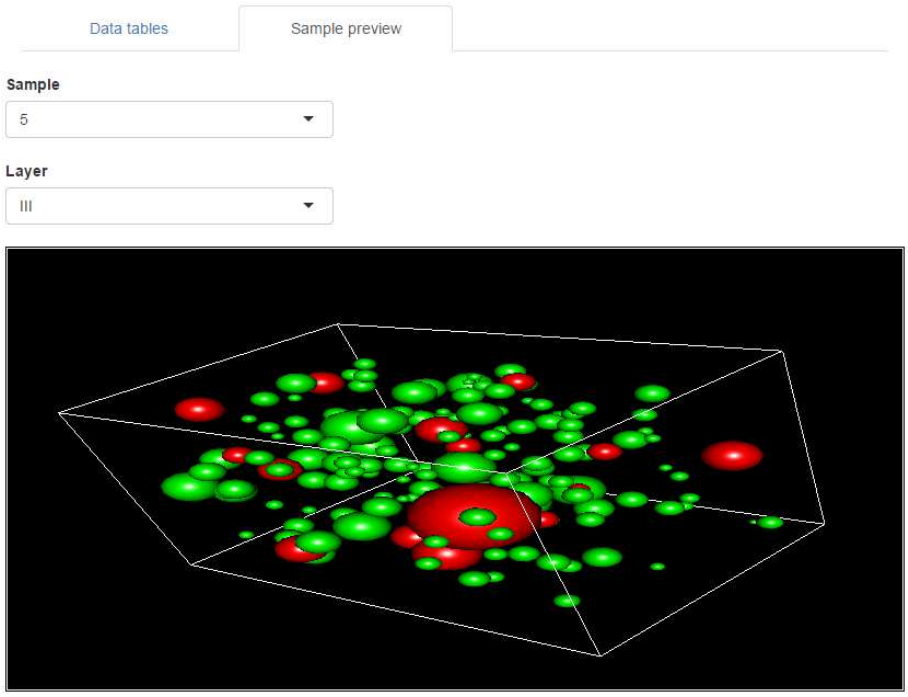


Figure 6: Loaded data preview

4.2 Modeling tab

In the modeling tab a sequential process is followed to create a model for a selected sample or a whole layer. The process starts modeling the Feret's diameter distribution from the selected data by fitting a log-normal distribution with parameters μ and σ. Then, a plot that compares the empirical distribution of values and the fitted model is created as well as an informative message with the distribution parameters and the $p - value$ for the goodness-of-fit test (fig. 7).

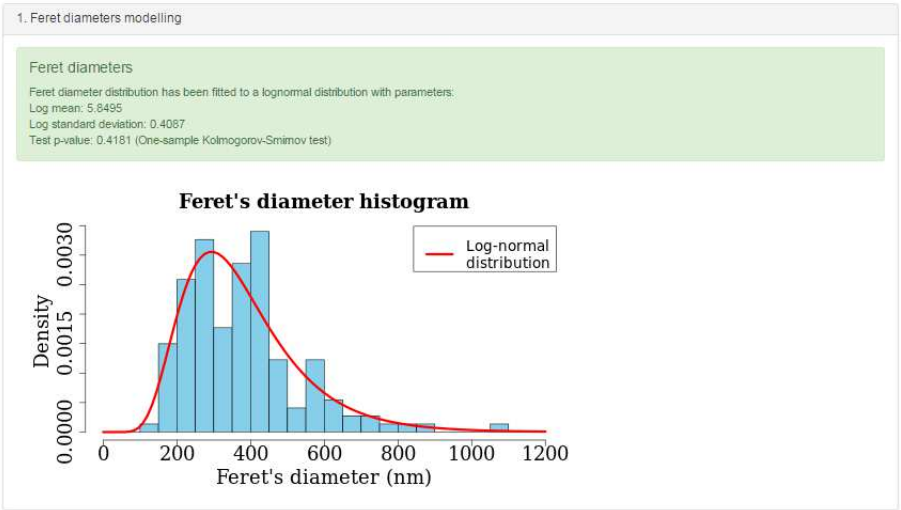


Figure 7: Feret's diameter fit plot

If the Feret's diameters from selected data do not follow a log-normal distribution, and therefore the goodness-of-fit test rejects the null hypothesis, a warning message will be displayed stating that results based on this hypothesis might be inaccurate (fig. 8).

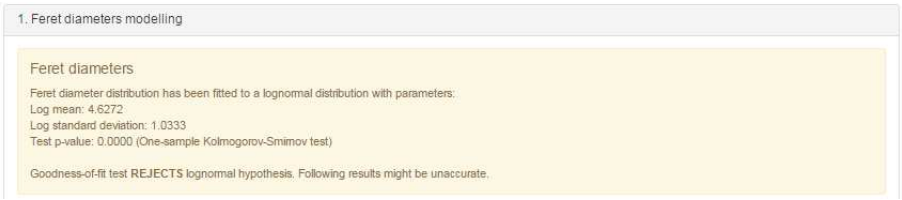


Figure 8: Fit fail message

The next step is to verify that selected data is distributed in the spatial domain following a random sequential adsorption process. This is done by creating a large number of RSA simulations that replicate original data and then computing their average and maximum deviation independently. The result of this process is a plot tat compares the original data, the CSR model, the RSA model and its envelope with respect to the L summary function (fig. 9). The envelope (grey) is defined as a region of constant width calculated as the furthest deviation between the RSA average (green) and and any of the separate set of 99 RSA simulations with the same parameters. If the Feret's diameters of the selected synapses do not follow a log-normal distribution, results obtained from this plot are not conclusive.

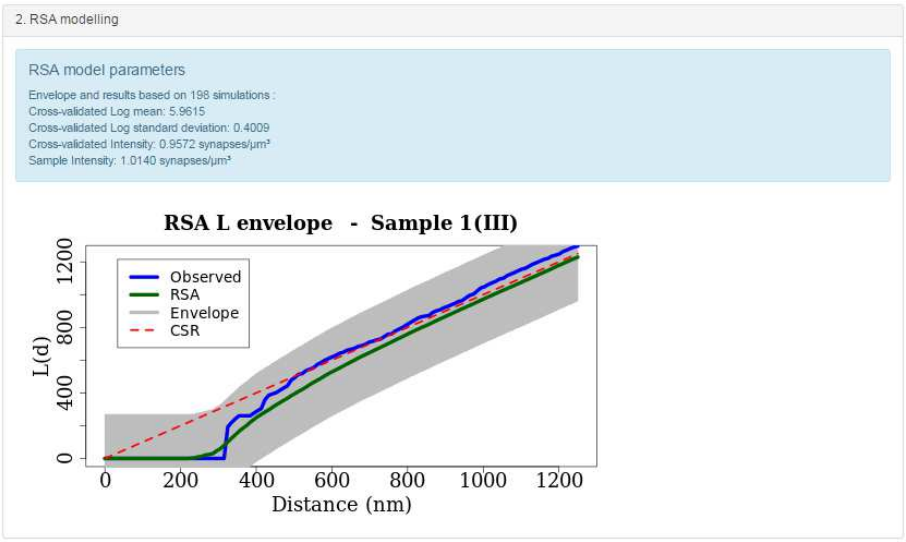


Figure 9: RSA envelope

Original data spatial distribution is said to verify RSA assumption if its curve in the plot is bounded within the RSA envelope. When testing a single sample, cross-validated parameters are obtained from other samples of the same layer if there is any. In the case of a whole layer, each sample from that layer is represented as a dashed-line.

It is important to note that, since the simulation process is computationally expensive, this operation can take several minutes depending on the number of CPUs available on the server.

Finally, the last step is to create a plot (fig. 10) that compares the curves of the original data, CSR model and RSA model with respect to the common summary functions in spatial statistics (F,G,K and L). Since there is no closed-form for the RSA model, for any of these functions, plots are based on previously computed RSA simulations.

In this case, the control panel (fig. 11) will show up as the user advances through the process and each button will activate or deactivate according to the current status of the process. Also, short help texts are included in the panel to guide the user.

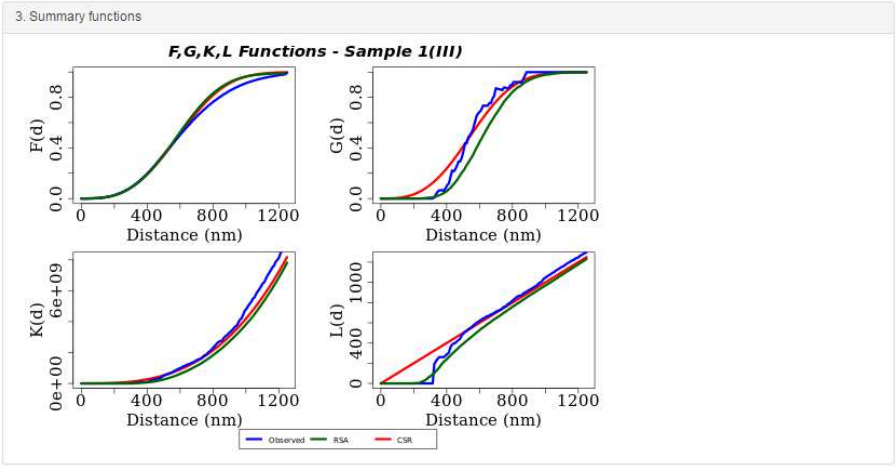


Figure 10: F,G,K,L Functions plot

4.3 Simulation tab

The simulation tab supports the generation of replicated samples, assuming log-normal distribution of Feret's diamters and RSA spatial pattern. The simulation is mainly driven by three parameters: Feret's diameters log mean (μ), Feret's diameters log standard deviation (σ) and simulation synaptic density/intensity λ.

The control panel in the simulation tab (fig. 12) gives three alternatives to set the former parameters:

- Input the parameters manually
- Calculate the parameters from a single sample
- Calculate the parameters as the volume-proportional average from a layer

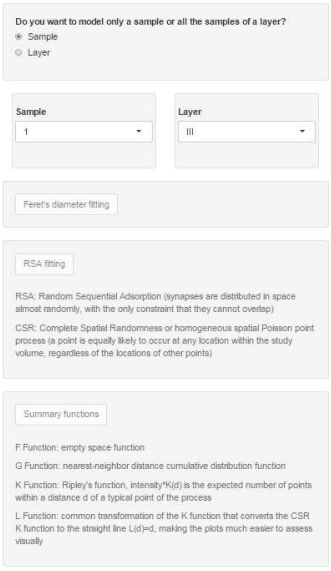
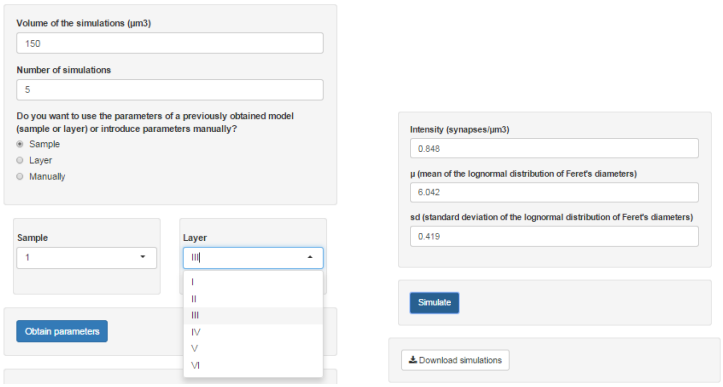


Figure 11: Model tab control panel



(a) (b)

Figure 12: Simulation tab control panel

Once simulations have been generated, a download button that generates a CSV file containing all simulations, should appear at the bottom of the control panel. Simulation number is used as the sample number, and if simulations are not based on parameters from a layer, layer field is set to 0 for every synapse.

The main panel in this tab includes 3D preview of each simulation (fig. 13), similar to the one described in the load tab. In this case, all simulated synapses are symmetric and plotted in the same color.

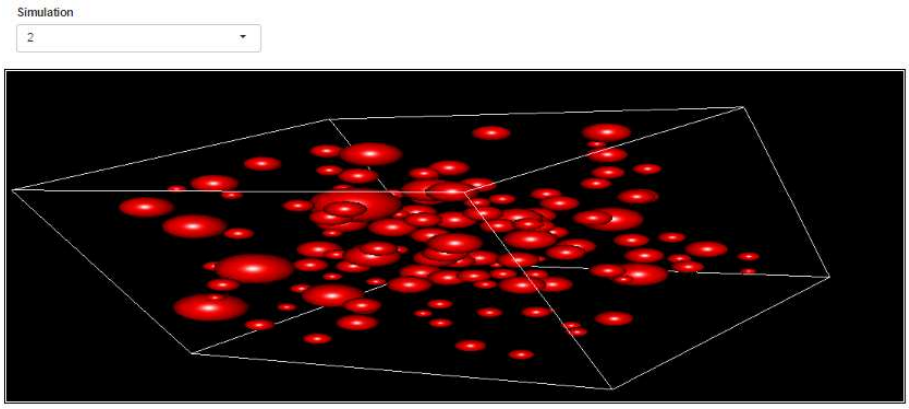


Figure 13: Simulation preview

4.4 Layer comparison tab

The last tab allows the user to perform a layer comparison on two indicators: the synaptic density and the distance to the nearest synapse (fig. 14). At least samples from two layers are required to perform this comparison, and in the case of synaptic density more than two samples per layer are needed.

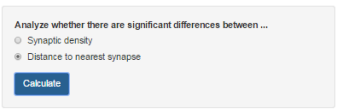


Figure 14: Layer comparison tab control panel

In both cases, normality and Levene variance homogeneity test are performed. If the data under study is normal and homogeneous, layers pairwise comparison is made by performing an ANOVA analysis combined with a Tukey honest significant differences test, otherwise, Kruskall-Wallis test is applied.

Results are presented in two ways(fig. 15):

- An information message that summarizes the normality, homogeneity and pairwise comparisons, including p-values, on the top; and
- A colored stacked-bar plot, where each layer height is proportional to the real layer's thickness and the color depends on the average value (either distance to the nearest synapse or synaptic density) for each layer

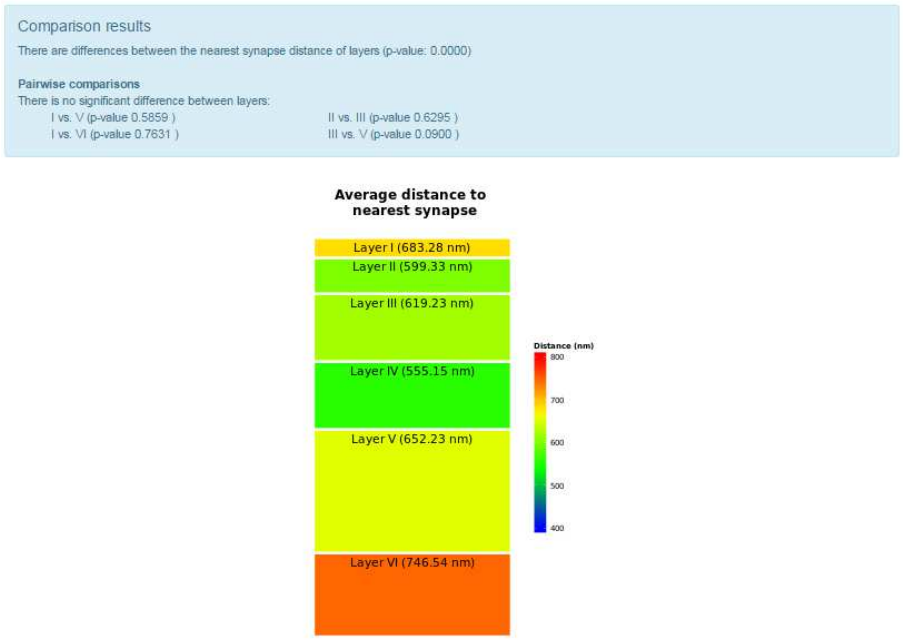


Figure 15: Layer comparsion plot

5 Common use cases

In the next subsections, basic procedures to perform common tasks with *synapsesSA* are described step-by-step. For any doubt regarding the user interface, please refer to section 4.

5.1 Detect data errors

Objective: To detect errors, such as overlaps, axis deviation,etc. In the data by visual inspection of the data tables or the 3D Preview.

Steps

1. Go to the *data loader* tab in the web interface
2. Click on choose files and select CSV files to be reviewed
3. Select appropriate decimal dot and field separator. Click load
4. Data table or error message should appear in the main panel
5. *[If there is no error]* Use table controls to browse through loaded data. Review loaded values for each column
6. Go to the *data preview* subtab
7. For every pair sample-layer inspect the visualization looking for overlaps or anomalies

END

5.2 Compare different layers

Objective: To compare distance to the nearest synapse for each layer, looking for similarities/differences between them.

Steps

1. Go to the *data loader* in the web interface
2. Click on choose files and select CSV files to be analyzed
3. Select appropriate decimal dot and field separator. Click load.
4. Data table or error message should appear in the main panel
5. Repeat steps 3 and 4 to load more files if needed
6. Go to the *layer comparison* tab
7. Select *distance to nearest synapse* and click 'Calculate'
8. Error or Info message and plot should appear in the main panel
9. Right click on the plot and select *save image as* to download the plot

END

5.3 Model synapses Feret's diameters

Objective: To determine if the Feret's diameters of a given layer follow a log-normal distribution or not.

Steps

1. Go to the *data loader* tab in the web interface

2. Click on choose files and select CSV files to be analyzed
3. Select appropriate decimal dot and field separator. Click load.
4. Data table or error message should appear in the main panel
5. Repeat steps 3 and 4 to load more files if needed
6. Go to the *model builder* tab
7. Select *layer* on the left panel
8. Select the layer to be analyzed in the selector
9. Click *Feret's diameter fitting* button
10. Warning or success message and plot should appear in the main panel
11. If the Feret's diameters are not log-normal, a yellow message stating that log-normal hypothesis has been rejected should appear. Otherwise, a green message with fitted parameters is displayed
12. Copy text in the info message and/or save the plot

END

5.4 Model synapses spatial distribution

Objective: To determine if the synapses of a given sample follow RSA spatial pattern or not.

Steps

1. Go to the *data loader* tab in the web interface
2. Click on choose files and select CSV files to be analyzed
3. Select appropriate decimal dot and field separator. Click load
4. Data table or error message should appear in the main panel
5. Repeat steps 3 and 4 to load more files if needed
6. Go to the *model builder* tab
7. Select *Sample* on the left panel
8. Select the sample and layer to be analyzed in the selector
9. Click *Feret's diameter fitting* button
10. Warning or success message and plot should appear in the main panel
11. If the Feret's diameters are not log-normal, a yellow message stating that log-normal hypothesis has been rejected should appear. Otherwise, a green message with fitted parameters is displayed
12. *[Stop if log-normal hypothesis has been rejected]*
13. Click on the *RSA fitting* button *[This process may take several minutes]*
14. Error or info message and plot should appear in the main panel
15. If the sample line (blue) is bounded within the RSA envelope (grey area), the sample is distributed following a RSA pattern

END

5.5 Create sample replicas

Objective: To create 20 simulated samples based on a given real sample.

Steps

1. Go to the load tab in the web interface
2. Click on choose files and select CSV files to be analyzed
3. Select appropriate decimal dot and field separator. Click load
4. Data table or error message should appear in the main panel
5. Repeat steps 3 and 4 to load more files if needed
6. Go to the *model builder* tab
7. Select *Sample* on the left panel
8. Select the sample and layer to be analyzed in the selector
9. Click *Feret's diameter fitting* button
10. Warning or success message and plot should appear in the main panel
11. If the Feret's diameters are not log-normal, a yellow message stating that log-normal hypothesis has been rejected should appear. Otherwise, a green message with fitted parameters is displayed
12. *[Stop if log-normal hypothesis has been rejected]*
13. Click on the *RSA fitting* button *[This process may take several minutes]*
14. Error or info message and plot should appear in the main panel
15. If the sample line (blue) is bounded within the RSA envelope (grey area), the sample is distributed following a RSA pattern
16. *[Stop if sample is not distributed following a RSA pattern]*
17. Click on the *Summary functions* button to generate a plot with the F,G,K and L functions.
18. Go to the *Model simulator* tab
19. Select *sample* on the left panel and set number of simulations to 20
20. Select the sample and layer to be replicated in the selector and Click the *Obtain parameters* button
21. Computed parameters should appear as numeric inputs in the left panel
22. Verify computed parameters and click on the *Simulate* button
23. Selector and 3D plot should appear in the main panel
24. Click the *Download simulations* button and save the CSV file

END

References

Laura Anton-Sanchez, Concha Bielza, Angel Merchán-Pérez, José-Rodrigo Rodríguez, Javier DeFelipe, and Pedro Larrañaga. Three-dimensional distribution of cortical synapses: a replicated point pattern-based analysis. *Frontiers in neuroanatomy*, 8, 2014.

Adrian Baddeley et al. Analysing spatial point patterns in r. Technical report, Technical report, CSIRO, 2010. Version 4. Available at www. csiro. au/resources/pf16h. html, 2008.

Adrian J Baddeley and Rolf Turner. Spatstat: An r package for analyzing spatial point pattens, 2004.

Janine Illian, Antti Penttinen, Helga Stoyan, and Dietrich Stoyan. *Statistical analysis and modelling of spatial point patterns*, volume 70. John Wiley & Sons, 2008.

Brian D Ripley. Modelling spatial patterns. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 172–212, 1977.

List of Figures