

Frequently asked questions (FAQ)

Q1. What is SPIKY?

A1. SPIKY is a graphical user interface (GUI) for monitoring synchrony between artificially simulated or experimentally recorded neuronal spike trains. It is based on three recently proposed methods to measure spike train (dis)similarity, the ISI- and the SPIKE-distance as well as SPIKE synchronization (for references see **Q15**). SPIKY is a free software package programmed by Thomas Kreuz and Nebojsa Bozanic. All source codes are written in Matlab (MathWorks Inc, MA) with the most time-consuming loops coded in MEX-files. It is not stand-alone but requires Matlab to run.

Q2. Where can I get SPIKY?

A2. You can download a zip-package containing all the necessary files for free on <http://www.fi.isc.cnr.it/users/thomas.kreuz/Source-Code/SPIKY.html>.

Q3. How do I get SPIKY started?

A3. First extract the zip-package (all files should be in one directory named 'SPIKY'), then within Matlab go to this directory or add it to the path ('addpath'). Compile the MEX-files in the 'SPIKY_MEX'-folder using 'SPIKY_compile_MEX', finally run 'SPIKY'.

[Important note: Due to computer architecture or platform differences (e.g. 32 vs. 64 bit) it can be quite important that you do the compilation again on your local computer instead of using the precompiled mex-files that are already included in the ZIP-file (Depending on your platform they might have extensions such as mexw32 or mexmaci64. You can use the Matlab-command 'mexext' to find out). In case you cannot get the MEX-files compiled (for whatever reason) just delete all mex-files in the SPIKY directory and SPIKY will use alternative Matlab-codes instead. However, this might be considerably slower and for very large datasets can also be more problematic regarding memory.]

Q4. What is the general structure and workflow of SPIKY?

A4. Please have a look at the file 'SPIKY-flowchart.pdf'.

Typically the suggested element for the next user action is marked by a 'bold font'. The most complete initial example is entry number 5 ('Clustering All') in the Data listbox. We suggest to follow this example through till the end advancing from panel to panel by pressing the highlighted button. With the same example in a second step you can start to change some parameters and see the consequences. Later, if you do not want to set all the parameters each time when you start SPIKY with a new dataset have a look at the file '**SPIKY_f_user_interface**' and try to understand how for this example #5 the spike train and the parameter values have been created. Hopefully you should then be able to do similar things with your own datasets. For quick information about the individual elements of the GUI activate the 'Hints'-checkbox in the 'Options'-Menu and then hover with the mouse cursor above the elements of interest and you will see short hints. An overview of all this information can be found in the file '**SPIKY-Elements.doc**'.

Q5. How can I load / generate spike train data?

A5. There are five possibilities:

1. In the 'Selection: Data' panel there is a listbox containing predefined examples. Initially these are the examples used in the paper **Kreuz et al., JNeurophysiol 109, 1457 (2013)** (for the complete reference see A13) but you can delete them and/or add your own data via the Matlab file '**SPIKY_f_user_interface**'. Just make sure that the listbox entries defined in the variable 'listbox_str' match the examples detailed below that.
2. You can load your own data via the *Load* button in the toolbar (left upper corner) or in the menu. Currently two different file formats are allowed, '.mat' and '.txt' (ASCII) files. For the mat-files SPIKY allows three different kinds of input formats:
 - cell arrays (ca) with just the spike times (this is the preferred format used by SPIKY since it is most memory efficient. The two other formats will internally be converted into this format)
 - regular matrices with each row being a spike train and zero padding (zp) in case the spike numbers are different.
 - matrices representing time bins where each zero/one (01) indicates the absence/presence of a spike

If the spikes are stored in a Mat-file SPIKY looks for a variable called 'spikes', if it cannot find it you have the chance to select the variable name (or field name) which contains the spikes via some input mask which provides a hierarchical structure tree.

In the text format spike times should be written as a matrix with each row being one spike train. The package contains one example file for each format ('testdata_ca.mat', 'testdata_zp.mat', 'testdata_01.mat' as well as 'testdata.txt').

3. You can load data from the workspace.
4. You can generate your own spike trains via the **Spike Train Generator**. After setting some defining variables (number of spike trains, start and end time, sampling rate) you can build your spike trains by using predefined spike train patterns (such as periodic, Splay, uniform or Poisson) and/or by manually adding, shifting and deleting individual spikes or groups of spikes. For an overview of the general structure of the spike train generator please have a look at the file '**STG-flowchart.pdf**', more detailed information can be found in the file '**STG-Elements.doc**'.
5. You can use the **Event detector** in order to detect discrete events in continuous data.

Q6. How can I change elements (fonts, lines, etc.) in the figure?

A6. Simply click the right mouse button on the element you wish to change. Please use the file '**SPIKY_f_user_interface**' to set the standard values for all the parameters that describe the layout of the figure (mainly the parameter structure 'p_para').

Q7. How can I move subplots (spikes and profiles subplot, matrices, dendrograms) in the figure? How can I change their size?

A7. Move the cursor to the respective axis (either just left or just below the subplot) and click the right mouse button. Now you can either edit all position variables by hand or change the x-position, the y-position, the width and the height individually. In case there are several matrices/dendrograms you can do this either for an individual matrix/dendrogram or for all of them at the same time.

Q8. How can I extract the spike trains and the results of the analysis (measure profiles, matrices) to the workspace?

A8. By clicking (right mouse button) on the element whose data you wish to extract. Results will be stored in variables such as 'SPIKY_spikes', 'SPIKY_profile_X_1', 'SPIKY_profile_Y_1', 'SPIKY_profile_name_1' as well as 'SPIKY_matrix_1' and 'SPIKY_matrix_name_1'. In addition, the results obtained during an analysis will automatically be stored in the structure 'SPIKY_results' which will have one field for each measure selected.

Q9. How do I save/print figures?

A9. Press the *Print to postscript* button in the toolbar (left upper corner).

Q10. How do I reset?

A10. Press the *Reset* button in the toolbar (left upper corner) and you will be right at the beginning. Press the *Reset with same data* button right next to it and you have the chance to change your initial parameter settings (including the selection of the measures to be calculated). You can even change your data via the *Spike train generator* in order to estimate the significance of certain features in your spike trains (i.e., look at the spike train dissimilarity with these features vs. the spike train dissimilarity without them).

Q11. How do I make a movie?

A11. In *Selection: Plots* you should check *Record movie*.

Q12. How can I add a graphical representation of my stimulus (aligned in time to the spike trains)?

A12. Not yet finished but to be included soon. The respective element is currently disabled.

Q13. How do I include selective or triggered averages based on the data?

A13. In *Selection: Measures* you should fill the textboxes for *Selective averages* or *Triggered averages* using Matlab format (please see the initial example number 5 ('Clustering All') in the Data listbox. If you would like to use data-dependent averages you can also put the name of a

Matlab file which will be executed before the calculation. An example is provided (see the file 'SPIKY_trig_ave.mat').

Q14. How can I get the statistics of a certain quantity over a large number of datasets?

A14. For this purpose you can use the program 'SPIKY_loop' which is complementary to the graphical user interface 'SPIKY'. Both programs can be used to calculate time-resolved spike train distances (ISI and SPIKE) between two (or more) spike trains. However, whereas SPIKY was mainly designed to facilitate the detailed analysis of one dataset, 'SPIKY_loop' is meant to be used in order to compare the SPIKY_results for many different datasets (e.g. in some kind of loop). [Note that the new program '**SPIKY_loop**' replaces the old 'SPIKY_no_plot'. The scope is the same but it adds the full functionality of SPIKY (access to time instants, selective and triggered averages as well as averages over spike train groups).]

Q15. How can I estimate the significance of my results?

A15. Please have a look at the program 'Spiky_loop_surro.m'. This is similar to 'SPIKY_loop.m' only that you don't look at different datasets; rather you can compare the results obtained for one dataset against the results obtained for spike train surrogates generated from that dataset.

Q16. I have a problem. How can I get help?

A16. Some general information is provided on this webpage:

<http://www.fi.isc.cnr.it/users/thomas.kreuz/Source-Code/SPIKY.html>.

You can also contact the authors via email: thomas.kreuz@cnr.it and nebojsa.bozanic@fi.isc.cnr.it.

We have also set up a [Facebook page](#) where you have the opportunity to provide feedback and ask any questions you might have.

Q17. I want to use the results/plots in my scientific work. How can I acknowledge the authors of the code?

A17. Regarding the methods of spike train analysis, if you use the SPIKE-distance please cite this paper:

Kreuz T, Chicharro D, Houghton C, Andrzejak RG, Mormann F:
Monitoring spike train synchrony.
J Neurophysiol 109, 1457-1472 (2013). [[PDF](#)].

If you use the ISI-distance please cite this paper:

Kreuz T, Haas JS, Morelli A, Abarbanel HDI, Politi A:
Measuring spike train synchrony.
J Neurosci Methods 165, 151 (2007). [[PDF](#)].

If you use SPIKE synchronization please cite this paper:

Kreuz T, Mulansky M, Bozanic N:
SPIKY: A graphical user interface for monitoring spike train synchrony.
J Neurophysiol 113, 3432 (2015), see also [arXiv](#) [[PDF](#)]

Also, if you use a figure or a movie created with SPIKY please cite this:

Kreuz T, Mulansky M, Bozanic N:
SPIKY: A graphical user interface for monitoring spike train synchrony.
J Neurophysiol 113, 3432 (2015), see also [arXiv](#) [[PDF](#)]

Further information can be found in these papers:

Bozanic N, Mulansky M, Kreuz T:
[SPIKY](#)
Scholarpedia 9(12), 32344 (2014).

Kreuz T:
[SPIKE-distance](#).
Scholarpedia 7(12), 30652 (2012).

Kreuz T:
[Measures of spike train synchrony](#).
Scholarpedia 6(10), 11934 (2011).

Kreuz T, Chicharro D, Greschner M, Andrzejak RG:
Time-resolved and time-scale adaptive measures of spike train synchrony.
J Neurosci Methods **195**, 92 (2011) [[PDF](#)].

Kreuz T, Chicharro D, Andrzejak RG, Haas JS, Abarbanel HDI:
Measuring multiple spike train synchrony.
J Neurosci Methods **183**, 287 (2009) [[PDF](#)].