# swan Documentation

*Release 1.0.0*

**Christoph Gollan**

**Jun 16, 2021**

# Contents

Swan (="sequential waveform analyzer") is a GUI application built with PyQt4. For plotting it also uses matplotlib's qt aggregation classes and pyqtgraph's plot widget. The calculation is done by numpy.

Swan is built to show reach-and-grasp data. It is possible to load multiple sessions to get an overview of all units from all sessions for one selectable electrode. In advance you can compare the units by either looking at the overview or by using the different views of the data, e.g. a 2d plot. A special feature of swan is that you can create virtual units. Virtual units are a mapping of units from multiple sessions that you define as the same unit.

Here is a list of the swan features:

- Show an overview of one electrode over many sessions.

- Show different views and layers of the data.

- Do a virtual unit mapping. Sometimes units are not mapped correctly so you can change the mapping virtually and save the mapping in a file.

- Calculate the mapping via algorithm or do it by hand

- Export the mapping to csv and odML

> **Warning:** This application is built on Linux and works on Linux. It *should* work on Windows as well but there could be some problems.

Contents:

# Getting started

Here you can find out how to get started with swan.

## 1.1 Requirements

Here is a list of packages needed by swan:

- matplotlib
- quantities
- PyQt4
- enum
- pyqtgraph
- lxml
- python-neo
- python-odml

The python version that is used is *2.7.6*.

## 1.2 Installing swan

### 1.2.1 Getting the packages

If you are using Ubuntu - or any other Debian based machine - you can install some of the packages via apt.

The command looks like this:

```
$ sudo apt-get install <package_name>
```

For the other packages (and other operating systems) you have to use pip.

Here is the command for pip:

```
$ pip install --user <package_name>
```

Look at the table to get the right package name:

| package | pip package | apt package |
|---------|-------------|-------------|
| matplotlib | matplotlib | python-matplotlib |
| quantities | quantities | python-quantities |
| PyQt4 | | python-qt4 |
| enum | enum | python-enum |
| lxml | lxml | python-lxml |
| pyqtgraph | pyqtgraph | |

In the top-level directory of swan you can find the file `install.sh` in which the commands for installing the packages can be found. You can use this file for installing the packages (this will only work on machines which support `apt-get install`):

```
$ ./install.sh
```

### 1.2.2 Getting Swan

To get the swan repository use the following command in your console:

```
$ git clone --recursive https://github.com/INM-6/swan.git
```

The `--recursive` is very **important** because swan depends on other repositories.

## 1.3 Running swan

If you are in the `swan/src` directory, the command to start swan looks like this:

```
$ python run.py [<home_dir>]
```

You can see that *<home_dir>* is optional. If you are fine with the fact that swan saves all project files in your home directory - which is usually ~ - you don't have to provide arguments.

But if you want to choose another directory you can just provide it as first argument.

# Tutorials
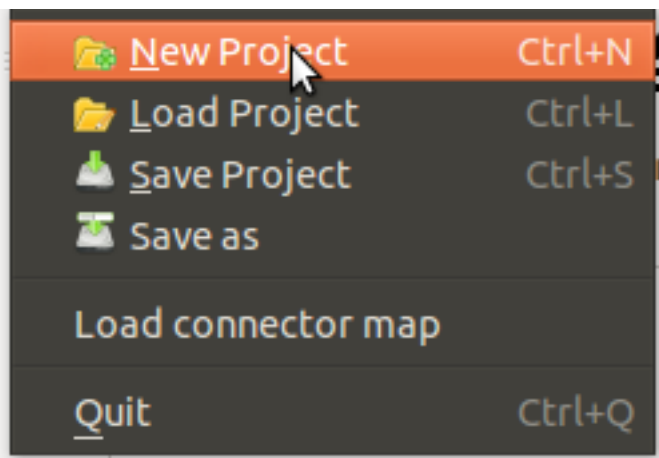
Here you can find tutorials for swan.

You can see how to create a new project, load a project, how the project files look like and so on.

Of course you can learn how to use swan's tool bar functions as well.

## 2.1 Projects

### 2.1.1 Create a new project

To create a new project you have to click on *File->New Project* or on the folder icon with the little plus on it.

| | | |
|---|---|---|
| 📁 New Project | Ctrl+N | |
| 📁 Load Project | Ctrl+L | |
| 📥 Save Project | Ctrl+S | |
| 📥 Save as | | |
| Load connector map | | |
| Quit | Ctrl+Q | |

A dialog will open where you can choose a directory containing ".nev" files.

Just click on *browse...* and confirm the directory by clicking on *Open*.

You will now see a list of files on the left of the dialog.

Choose the files you want to load. "Ctrl-Click" and "Shift-Click" for multiple file selection are supported.

By clicking on *Add* the files will be added to the right list. These are the files that will be loaded.

If you want to add another file(s), you can just choose them and click *Add* again.

and the project will be created.

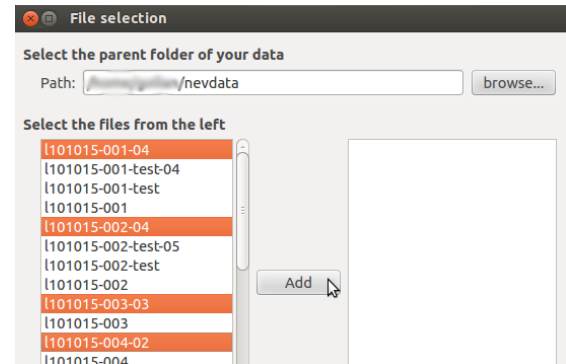The files will be loaded directly after that.

### 2.1.2 Load a project

To load a project click on *File->Load Project* or on the folder icon with the little pencil on it and choose the project file.
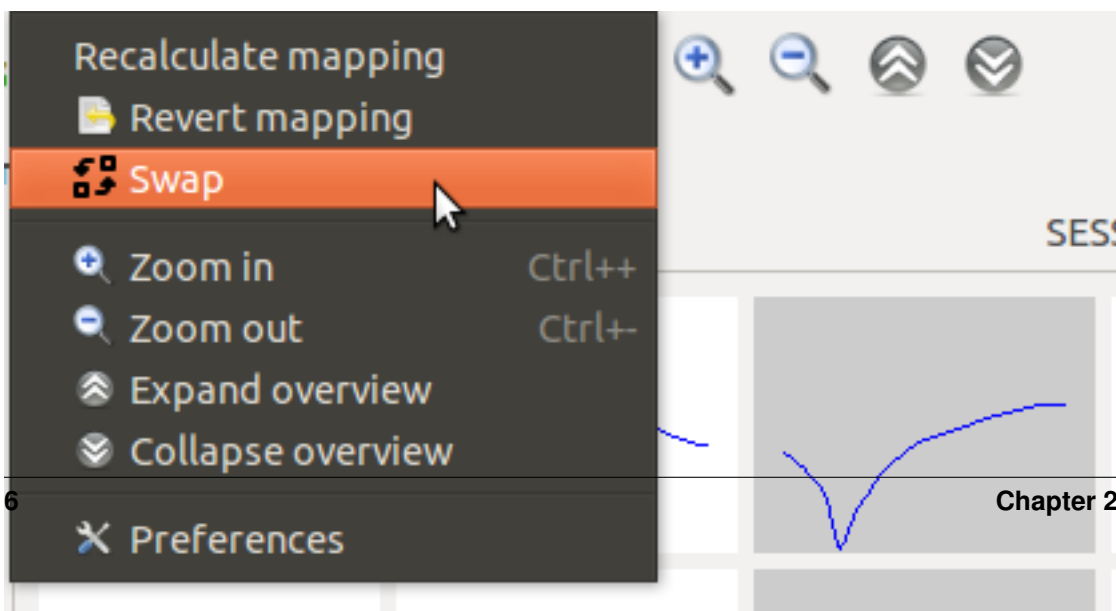
Now click on *Open* and the project will be loaded.

If you want to remove files from the right list, just select them and click *Remove*.

Now that you have your files selected, click on *Ok*



## 2.2 Overview

### 2.2.1 Swapping units

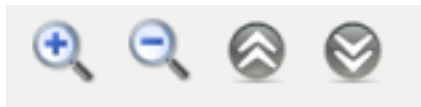To swap units you have to click on *Edit->Swap* or on the swap icon.

Swapping is only allowed if you have two units selected which both are in the same session. Otherwise

nothing will happen.

To select units just click on them.

## 2.2.2 Resizing and Zooming

You can change the size of the overview plots by clicking on the loupe icon with the plus and the minus inside of it.

Or you click on *Edit->Zoom in* or *Edit->Zoom out*. The keyboard shortcuts *Ctrl +* and *Ctrl -* work as well.

To change the y-limits of the plots, just click on the icon with the two arrows that are pointing down or up.

Of course you can do it via menu, too. Therefore you have to click on *Edit->Expand Overview* or *Edit->Collapse Overview*.

## 2.3 Preferences

To open the preferences click on the screwdriver and screw-wrench icon or open it via menu by selecting *Edit->Preferences*.

### 2.3.1 General

**Default project name:** Sets the default name for the project file which will be used if you create a new project. The project file contains the (absolute) paths to the files with the data for this project.

**Cache location:** Sets the location for the cache files. Cache files are needed to load the data faster after loading it the first time.
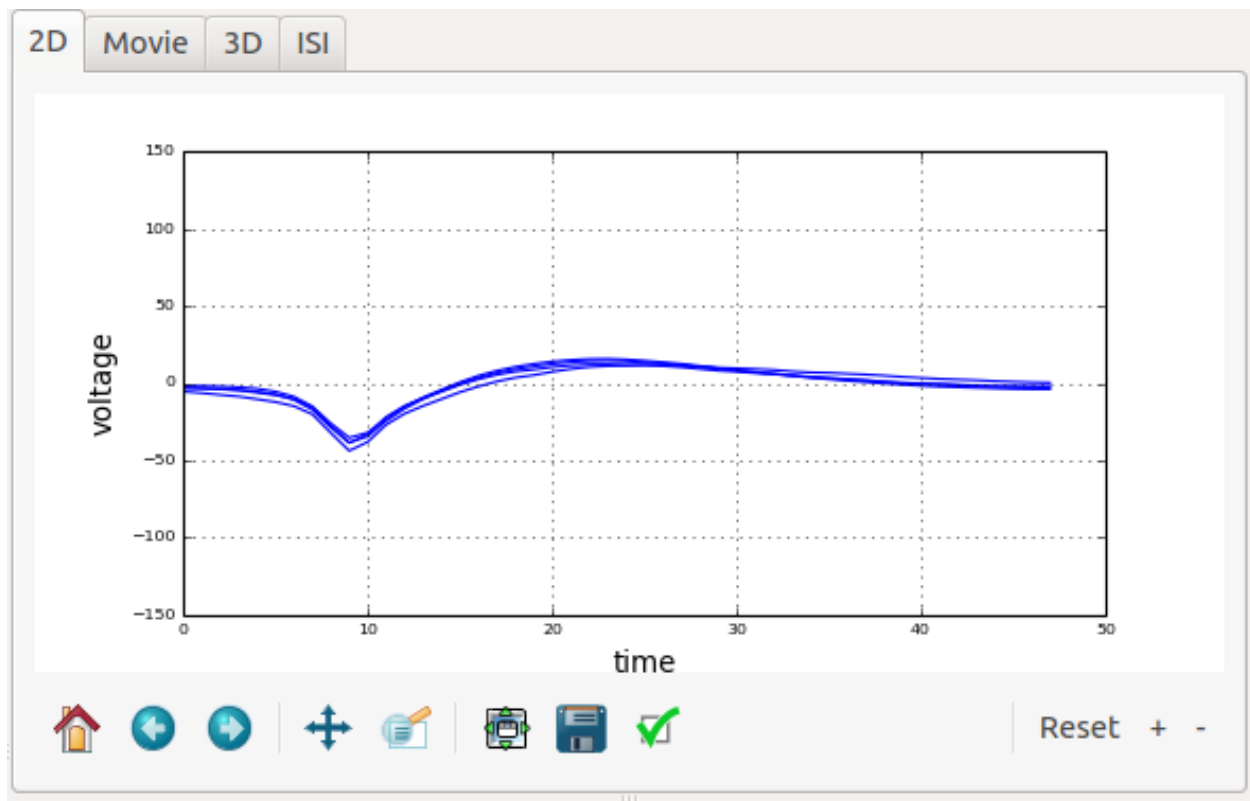
---

## 2.3.2 Overview

**Resizing:** Sets the zoom in and zoom out steps for the overview plots. Look at *Resizing and Zooming* for more information.

**Reranging:** Sets the steps for changing the y-limits of the overview plots. Look at *Resizing and Zooming* for more information.

# 2.4 Views

## 2.4.1 2d view



In this view you can see two-dimensional data from all units and sessions you loaded. The colors are the same as you see them in the overview. The mapping of the units are the same, too. With this view you can compare the mappings you made in the overview in detail. To better compare the units you can deselect complete rows of units (virtual units) by deselecting them in the *units* tab. In the *layers* tab you can choose which data you want to see.

### 2.4.2 Movie view
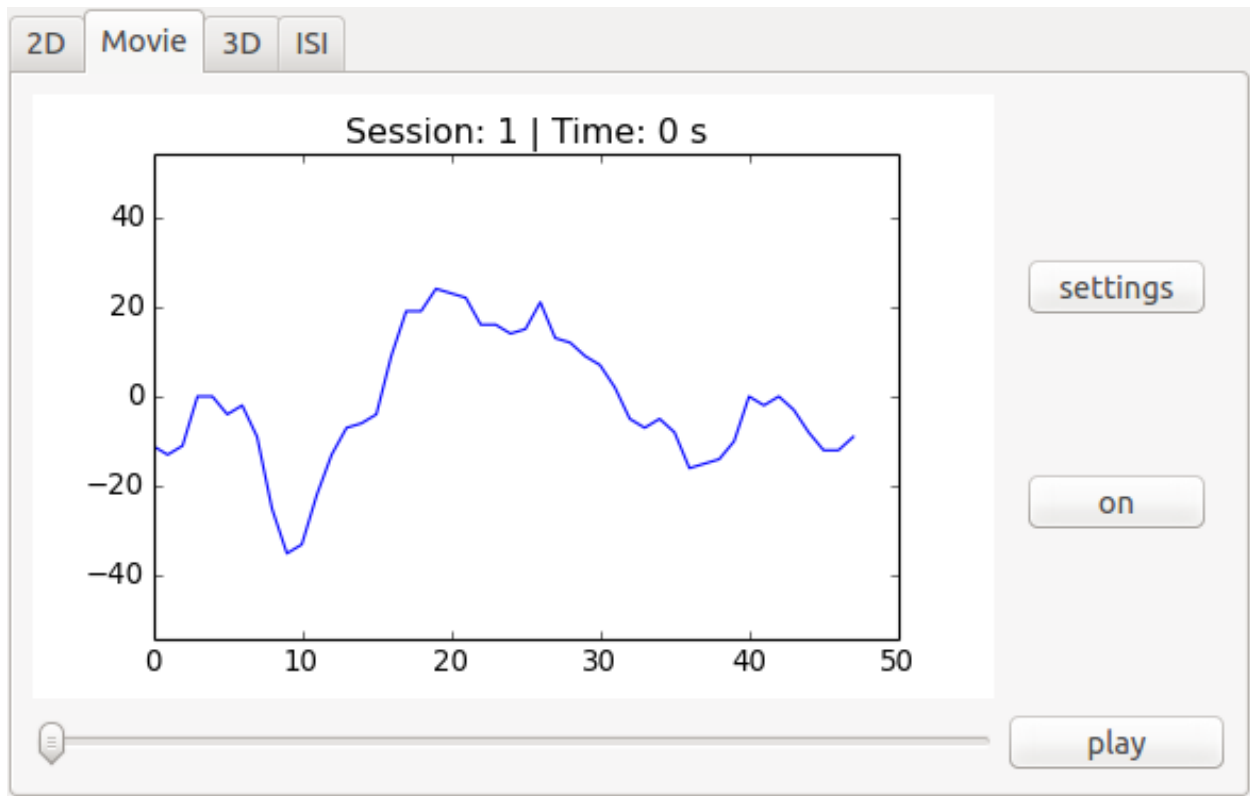
This view shows data (waveforms) of the units for all session. You can choose the unit rows (virtual units) to show via the *units* tab.

By selecting *off* the button will turn to *on* and the data will be loaded. By selecting *play* you can see the data in a movie-like way, the plot will update itself automatically. To change the settings for this view select *settings*.
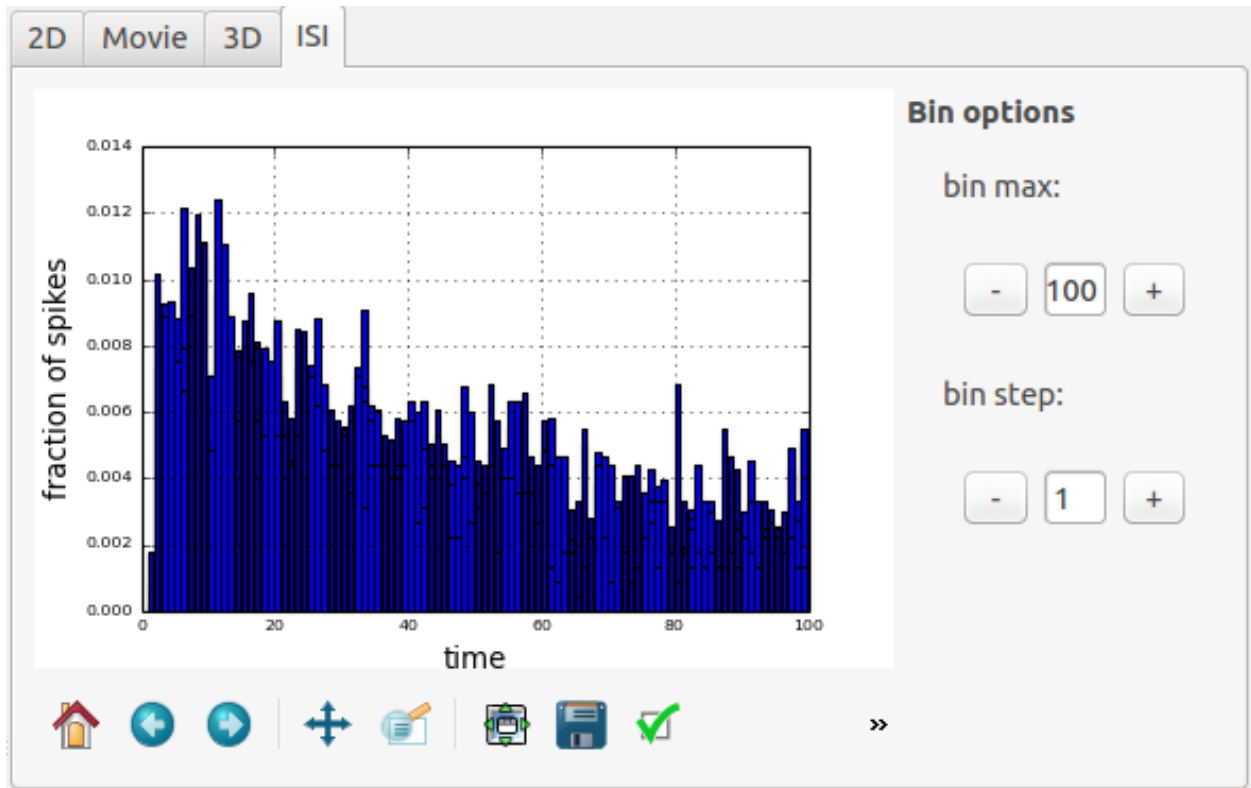
### 2.4.3 3d view
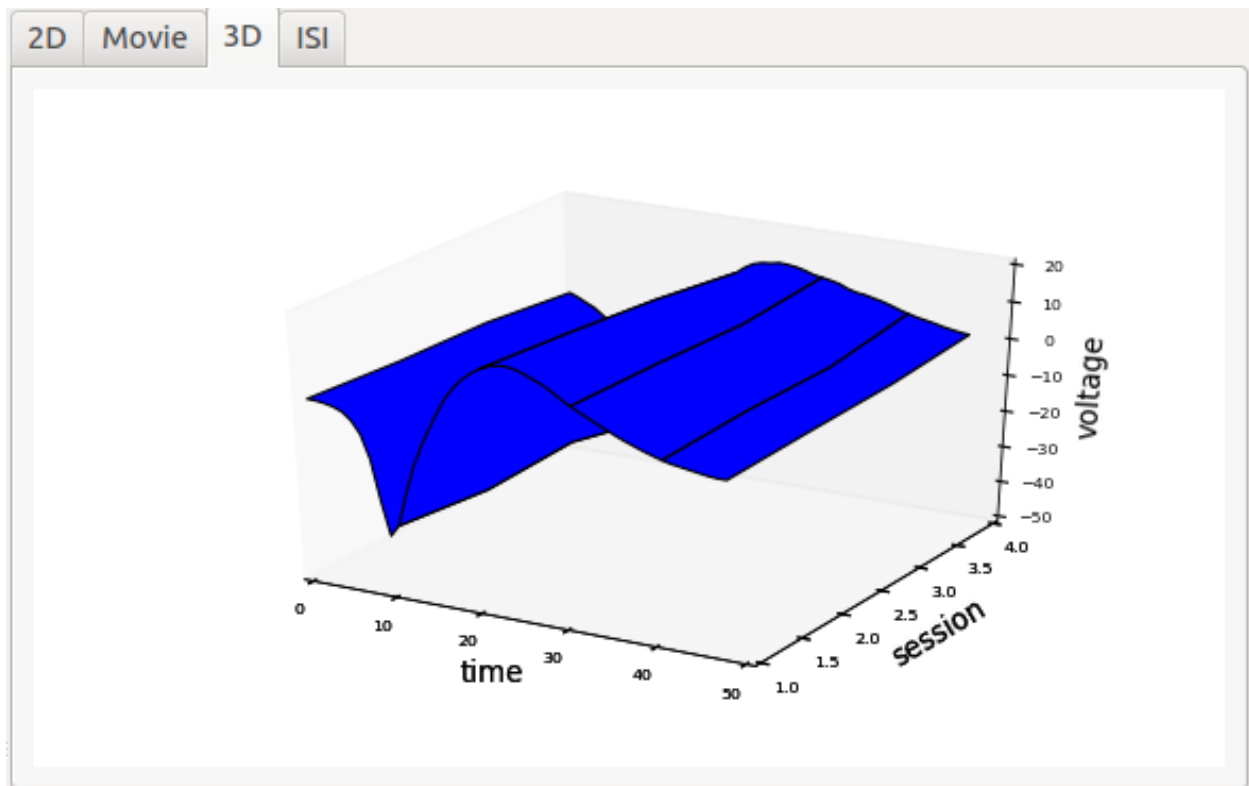


This view shows the data from one unit row (virtual unit) in a three-dimensional plot. To change the unit row (virtual unit) which is shown deselect the ones you don't want to see in the *units* tab. It will be plotted the first selected row in the *units* tab. In the *layers* tab you can choose which data you want to see.
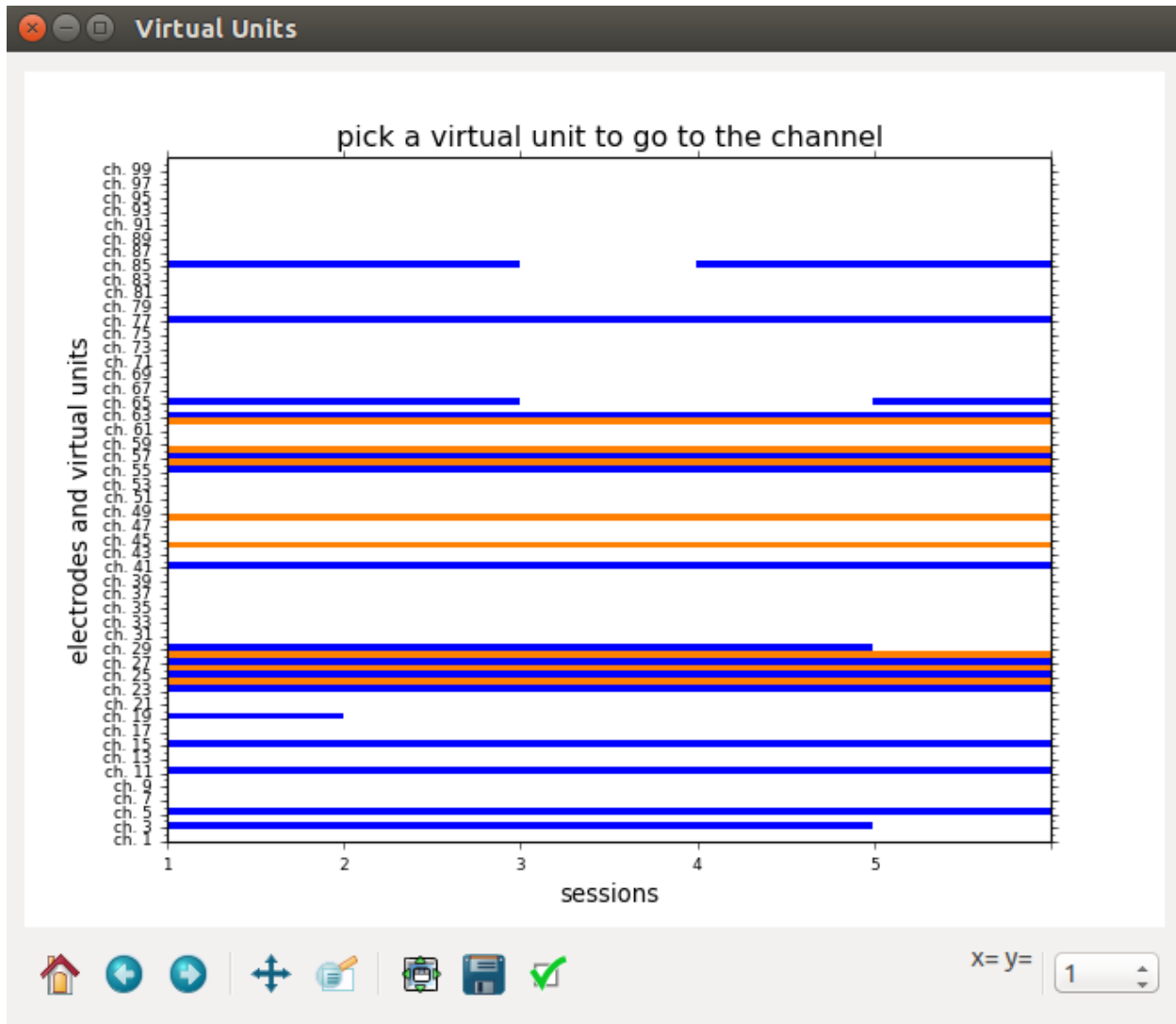
### 2.4.4 ISI view



In this view histograms are plotted. You can see the inter-spike-interval histograms for the units and session. You can de-s-elect unit rows (virtual units) in the *units* tab if you don't want to see their data. You can choose different histogram data in the *layers* tab.

## 2.4.5 Virtual units view



To see this view click on *virtual units* on the toolbar or select *View->Virtual units*. This view gives you an overview of existing virtual units. After you loaded an electrode and are done with the mapping you can save the project. Then this view will show the virtual units of that electrode. If you that the virtual unit is colored it means that there are existing units. Not existing units are not colored. You can change the number of shown virtual units by selecting an option in the bottom right corner of the view. If you select a virtual unit in the plot the electrode this virtual unit belongs to will be loaded. This will not work if you select the *auto* option.

Packages and modules

## 3.1 Packages

### 3.1.1 src - Application layer classes

**src package**

**Submodules**

**src.export module**

**src.file_dialog module**

**src.layerwidget module**

**src.main module**

**src.matplotlibwidget module**

**src.movie_settings module**

**src.mplwidget2d module**

**src.mplwidget3d module**

**src.mplwidgetisi module**

**src.mplwidgetmovie module**

**src.mylistwidget module**

**src.myplotgrid module**

**src.myplotwidget module**

**src.mystorage module**

**src.neodata module**

**src.preferences_dialog module**

**src.run module**

**src.selectorwidget module**

**src.swan module**

**src.unitswidget module**

**src.virtualunitmap module**

**src.vunits module**

**Module contents**

## 3.1.2  stg - General storage classes

**stg package**

**Submodules**

**stg.project module**

**stg.storage module**

**Module contents**

# CHAPTER 4

# Indices and tables

- genindex
- modindex
- search