Jellyfish manual

Wouter Franssen & Bas van Meerten

25th June 2019

Contents

1 Introduction

Jellyfish is a program for the simulation of 1D NMR spectra for liquid state samples specialised in complicate J-coupling patterns. It features a graphical user interface for intuitive simulations, and a python library for advanced use. Jellyfish is written in the Python programming language, and is cross-platform and open-source (GPL3 licence).

2 Running Jellyfish

2.1 Python and library versions

Jellyfish has been programmed to run on both the python 2.x and 3.x. Jellyfish should run on python versions starting from 2.7 and 3.4. For the library version, the following are needed:

- `numpy` >= 1.11.0
- `matplotlib` >= 1.4.2
- `scipy` >= 0.14.1

Jellyfish also needs the `PyQt` (version 4 or 5) library.

2.2 Installing

2.2.1 Linux

On Linux, Jellyfish can be most efficiently run using the python libraries that are in the repositories. For Ubuntu, these can be installed by running:

```
sudo apt install python python-numpy python-matplotlib python-scipy
python-pyqt5
```

Navigating to the `Jellyfish` directory in a terminal, Jellyfish can then be run by executing `python Jellyfish.py`.

2.2.2 Windows

On Windows, the relevant python libraries can be installed by installing `anaconda`: https://www.anaconda.com/download/. If you do not have another python version installed already, make sure to add anaconda to the `path` during the install. In this case, Jellyfish can be run by executing the `WindowsRun.bat` file in the `Jellyfish` directory. Destktop and start menu shortcuts to this file can be created by executing the `WindowsInstall.vbs` file.

If you already have other version of python installed, adding anaconda to the `path` might create issues. Do not do this in this case. When not added to the path, the `WindowsRun.bat` should be edited in such a way that `pythonw` is replaced with the path to your `pythonw.exe` executable in the anaconda install directory.

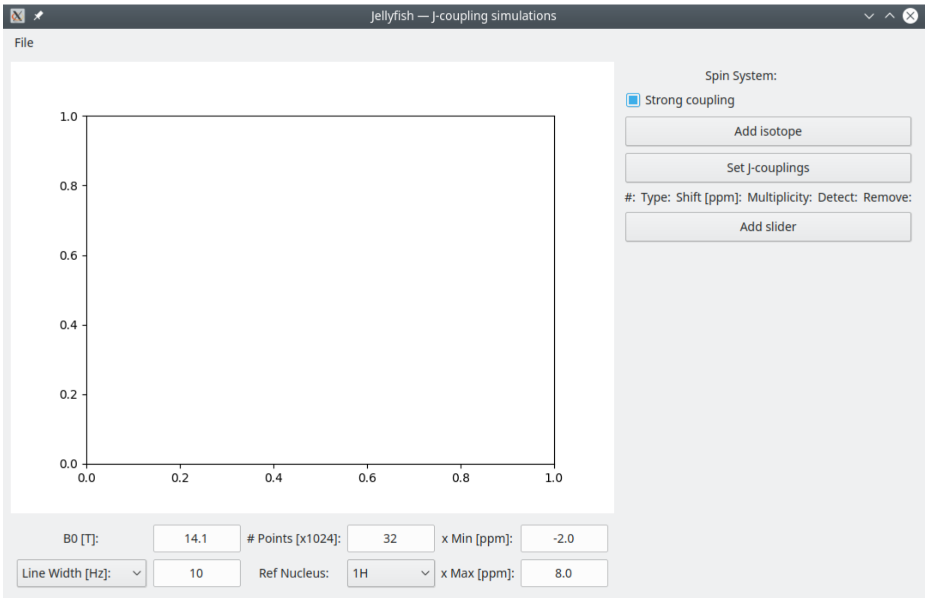
2.2.3 OS X

On OS X, the relevant python libraries can be installed by installing `anaconda`: https://www.anaconda.com/download/. Navigating to the `Jellyfish` directory in a terminal, Jellyfish can then be run by `anacondapython Jellyfish.py`, with `anacondapython` replaced by the path to your anaconda python executable.

3 The graphical user interface

Jellyfish can be run either as a standalone program, via its graphical user interface (GUI), or as a library by loading it in a Python script. Firstly, I will describe how to use the GUI.

Opening the program shows the main menu:



The window has two regions of settings: the spin system on the right, and the spectrum settings in the bottom. Also, there is a plot region.

3.1 Plot region

There are several ways to control the display of the spectrum using the mouse. Below, a list of these is given:

- Dragging a box while holding the left mouse button creates a zoombox, and will zoom to that region.
- Dragging while holding the right mouse button drags (i.e. pans) the spectrum. Doing this while holding the Control button pans only the x-axis, holding Shift pans only the y-axis.
- Double clicking with the right mouse button resets the view to fit the whole plot (both x and y).
- Scrolling with the mouse wheel zooms the y-axis. Doing this while also holding the right mouse button zooms the x-axis. By holding Control the zooming will use a larger step size.

3.2 The bottom frame

In the bottom frame, several parameters can be changed:

- B0 [T]: the magnetic field strength in Tesla. This value can also be changed from a slider (see below)
- Line Width [Hz/ppm]: the line broadening added to the spectrum simulation. Input either in Hertz or ppm (can be changed via the dropdown menu). The ppm setting is most useful when changing the magnetic field: the effective line width remains constant then.

- # Points [x1014]: The number of points in the spectrum.
- Ref Nucleus: Defines the reference frequency (i.e. nucleus) of the spectrum.
- x Min [ppm]: The minimum x-value in ppm
- x Max [ppm]: The maximum x-value in ppm

3.3 The spin system frame

On the right-hand side of the window, the spin system can be defined. Jellyfish support spin systems of any size (but your computer will not be able to handle very large systems, of course). There are several buttons always available:

- Strong coupling: toggle that determines if strong coupling is present. This determines if the J_{xy} part of the Hamiltonian is added. In realty, this is always on. The toggle is provide here to be able to show the effect of this strong coupling.
- Add isotope: Adds an isotope to the spin system. The gives a pup-up window were the type of nucleus can be selected, the chemical shift can be given, and the multiplicity defined. Pushing OK adds this spin to the system. This adds a row of input boxes to the interface, were the shift and multiplicity can be changed.
- Set J-couplings: Opens a window to set the J-coupling parameters. By default, all are 0. The amount of entries depends on the spin-system that has been defined at the moment.
- Add slider: Adds a slider to the interface. This can be used to change a specific parameter while watching the resulting spectrum. Ideal for teaching purposes on the effect of strong coupling, field strengths, etc. Pushing this button opens an input window, where the type (B0, Shift, or J-coupling) needs to be defined, as well as the minimum and maximum value of the slider. For the 'Shift' or 'J-coupling' type, one or two spin numbers also need to be given.

When a spin is defined, an extra line is added to the interface with the following options:

- Shift [ppm]: the chemical shift in ppm
- Multiplicity: the multiplicity of the nucleus (i.e. the ^{1}H multiplicity of a CH_3 group is '3')
- Detect: whether or not this nucleus is detected (i.e. added to the spectrum)
- X: remove this nucleus from the spin system. This also clears all relevant J-couplings, and removes and slider that has a relation to this spin.

When a slider is defined, it adds a row to the slider space. Here, there now is a slider which can be used to change the specified parameter. There is also a 'X' button to remove the slider. When using a slider, the respective value is also changed in the rest of the interface (e.g. chemical shift are also updated in the spin system rows).

3.4 Menu

Jellyfish also has a menu on the top of the interface. It has the following options:

- File:
  - Export Figure: export the current plot as an .png image.
  - Export Data (ASCII): save the current spectrum as a text file (x and y data).
  - Export as ssNake .mat: save spectrum as a Matlab file as used by the ssNake software (see `https://www.ru.nl/science/magneticresonance/software/ssnake/`).
  - Export as Simpson: save as a text file as supported by the Simpson simulation software (see `http://inano.au.dk/about/research-centers/nmr/software/simpson/`).
  - Quit: closes the program

4 Running as a script

Apart from running Jellyfish as a program via the user interface, it can also be used as a library from within python. This requires that you have the source code of Jellyfish (and not a compiled version). The 'Examples' directory holds some examples on how to do simulations from a script. Below an easy example is given. Here, it is assumed that the Jellyfish main directory (which holds `engine.py` is one directory higher than this file.

```python
import numpy as np
import sys
sys.path.append("..")
import engine as en

#--------Spectrum settings--------------
Base = 42.577469e6
RefFreq = 600e6 #zero frequency
B0 = RefFreq/Base #B0 is proton freq divided by the base scale
StrongCoupling = True #Strong coupling on
Lb = 0.2 #Linewidth in Hz
NumPoints = 1024*128 #Number of points
Limits = np.array([-1,3]) #Limits of the plot in ppm

#--------Spin system--------------
# add spin as ['Type', shift, multiplicity, Detect]
```

```
SpinList = [['1H',0,1,True]]
SpinList.append(['1H',2,1,True])

Jmatrix = np.array([[0, 10],
                    [ 0, 0]])


#---------Make spectrum-----------
# Prepare spinsys:
spinSysList = en.expandSpinsys(SpinList,Jmatrix)
# Get frequencies and intensities
Freq, Int = en.getFreqInt(spinSysList, B0, StrongCoupling)
# Make spectrum
Spectrum, Axis, RefFreq = en.MakeSpectrum(Int, Freq,
                                          Limits, RefFreq,Lb,NumPoints)
#Save as ssNake Matlab file
en.saveMatlabFile(Spectrum,Limits,RefFreq,Axis,'easy.mat')
```

5 Contact

To contact the `Jellyfish` team write to `ssnake@science.ru.nl`.

Bibliography