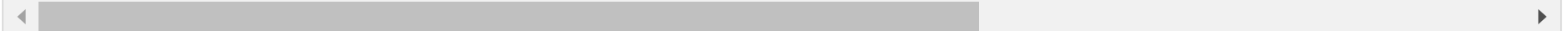


```
In [1]: # define the functions

%pylab nbagg
from tvb.simulator.lab import *
from tvb.datatypes.time_series import TimeSeriesRegion
import numpy as np
import time as tm
import matplotlib.pyplot as plt
import sys
```

Populating the interactive namespace from numpy and matplotlib

```
In [ ]: # Connectivity
We start by loading and visualizing the structural connectivity matrix that represents the set of all existing anatomical connections.
we can then alter the speed of signal propagation through the network.
```



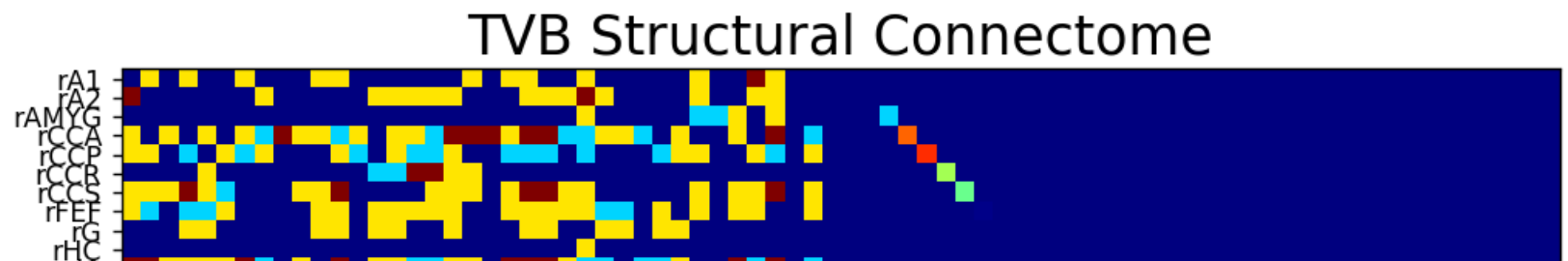
In [2]: *# Initialise the Connectivity.*

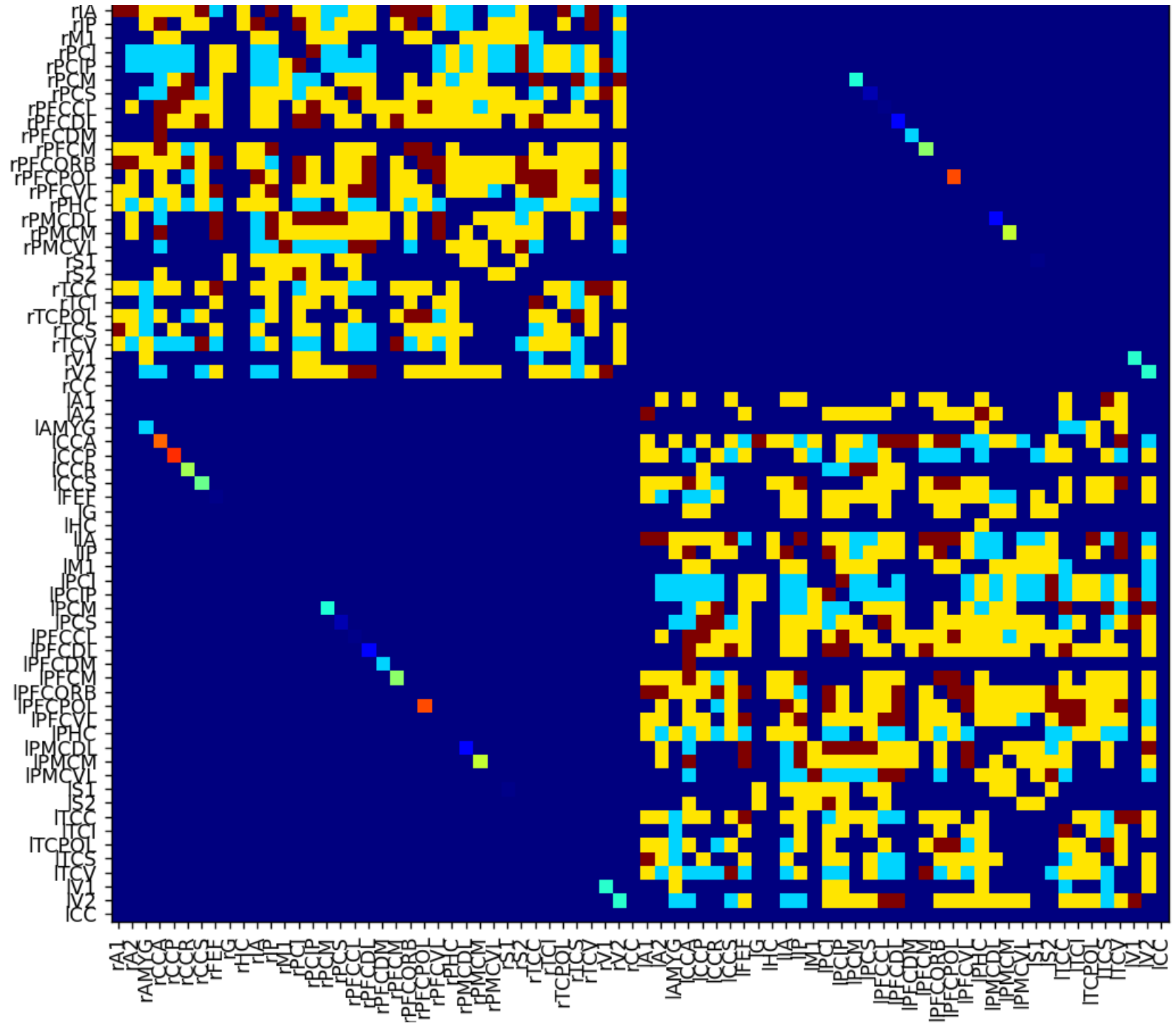
```
con = connectivity.Connectivity.from_file('connectivity_76.zip')
nregions = len(con.region_labels)           #number of regions
con.weights = con.weights - con.weights * eye((nregions))  #remove self-connection
con.speed = np.array([sys.float_info.max])  #set conduction speed (here we neglect it)
con.configure()

# Visualization.
figure(figsize=(12,12))
imshow(con.weights, interpolation='nearest', aspect='equal', cmap='jet')
title('TVB Structural Connectome', fontsize=20)
xticks(range(0, nregions), con.region_labels, fontsize=10, rotation=90)
yticks(range(0, nregions), con.region_labels, fontsize=10)
cb=colorbar(shrink=0.8)
cb.set_label('weights', fontsize=14)
show()
```

WARNING File 'hemispheres' not found in ZIP.

Figure 1







In [3]: *# Initialise the Model*

```
mod = models.ReducedWongWang(a=numpy.array([0.27]), w=numpy.array([1.0]), I_o=numpy.array([0.3]))
S = linspace(0, 1, 50).reshape((1, -1, 1))
C = S * 0.0
dS = mod.dfun(S, C)

sim = simulator.Simulator(
    model=mod,
    connectivity=con,
    coupling=coupling.Linear(a=np.array([0.5 / 50.0])), # initialise the coupling function
    integrator=integrators.EulerStochastic(dt=1, noise=noise.Additive(nsig=numpy.array([1e-5]))), # Initialise an Integrator
    monitors=(monitors.TemporalAverage(period=1),),
    simulation_length=60e3
).configure()

(time, data), = sim.run()
```

```
In [4]: # Perform simulation.
tic = tm.time()

tavg_time, tavg_data = [], []
for tavg in sim(simulation_length=60000):
    if not tavg is None:
        tavg_time.append(tavg[0][0])
        tavg_data.append(tavg[0][1])

'simulation required %0.3f seconds.' % (tm.time()-tic)
```

Out[4]: 'simulation required 10.178 seconds.'

```
In [5]: # Normalize time series
tavg_data /= (np.max(tavg_data, 0) - np.min(tavg_data, 0))
tavg_data -= np.mean(tavg_data, 0)

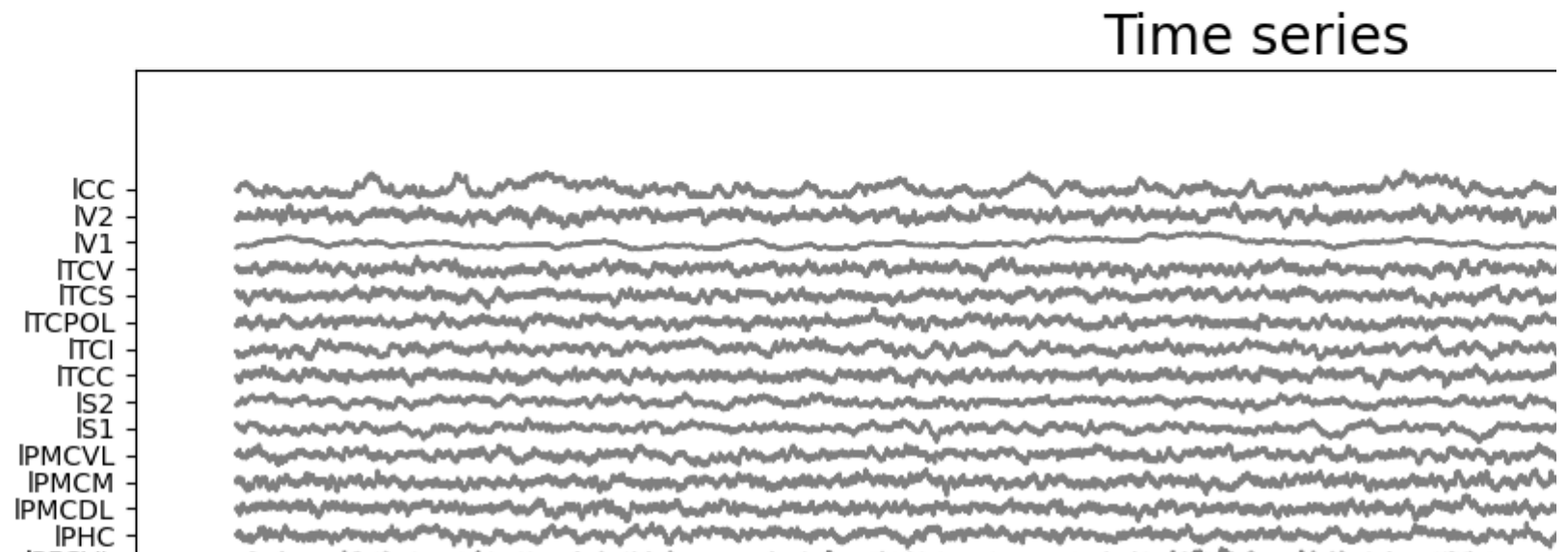
# Make lists numpy.arrays for easier use.
TAVG = np.squeeze(np.array(tavg_data))
TAVG.shape
```

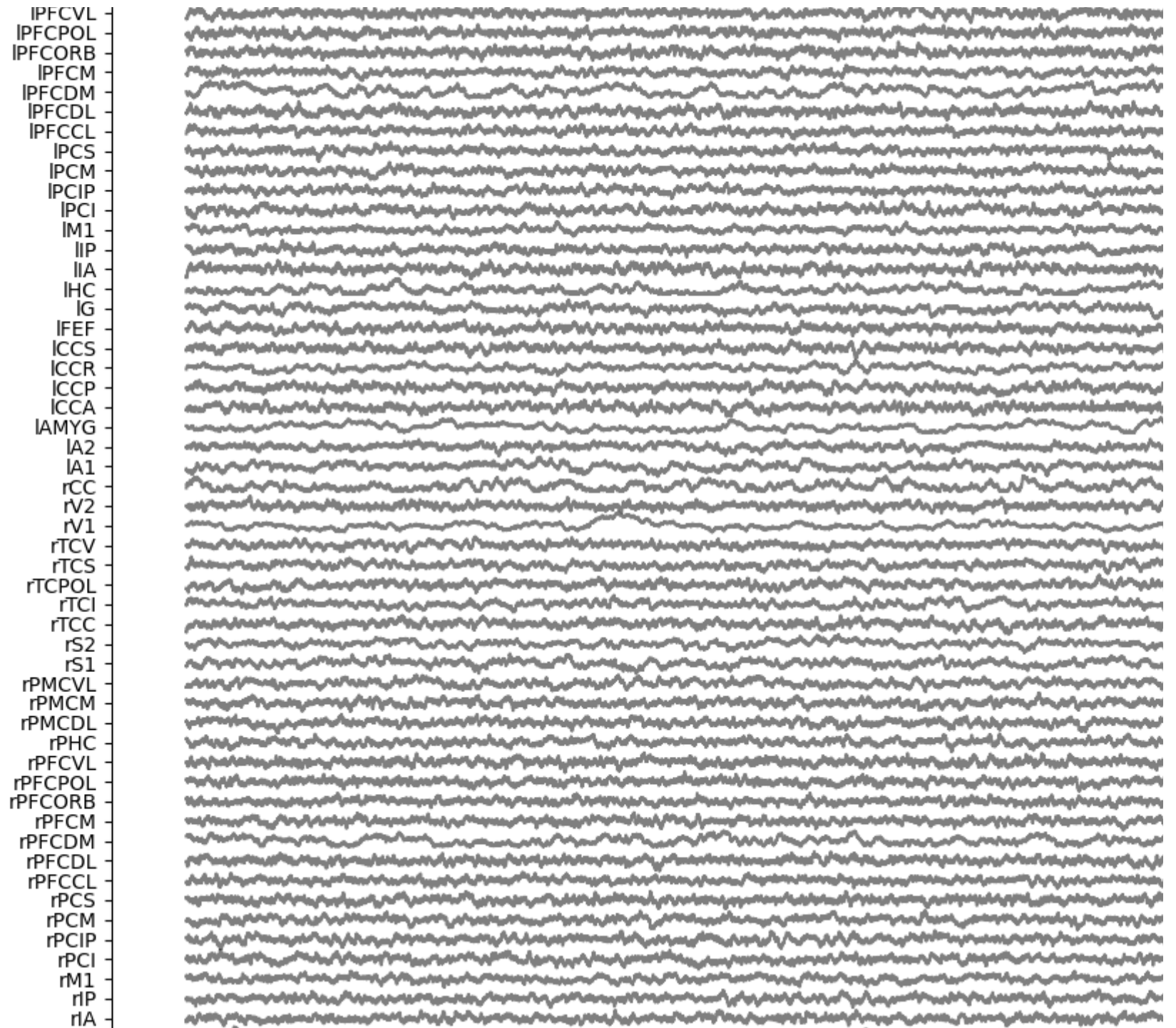
Out[5]: (60000, 76)

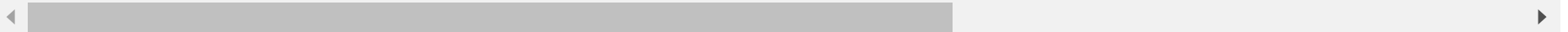
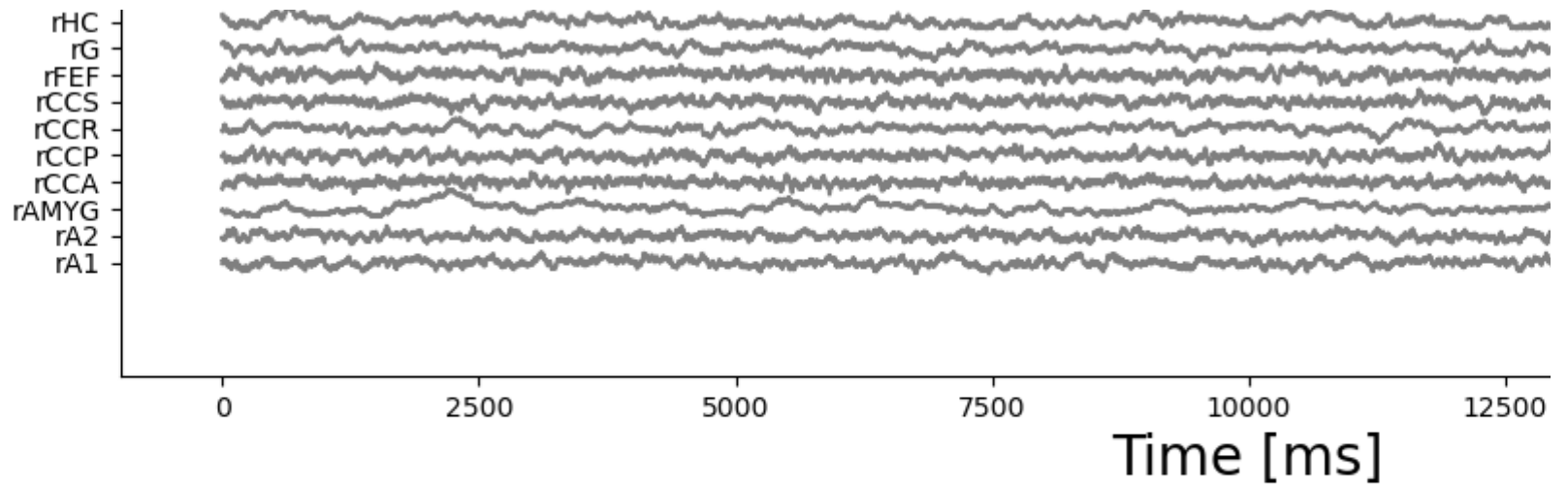
```
In [22]: # Plot time series.
fig1 = plt.figure(figsize=(15,15))
plt.plot(TAVG[20000:40000, :] + r_[:nregions], 'k', alpha=0.5)
plt.title('Time series', fontsize=20)
plt.xlabel('Time [ms]', fontsize=20)
plt.yticks(np.arange(len(con.region_labels)), con.region_labels, fontsize=10)
plt.show()

# To save the plot fig1.savefig('test.png')
```

Figure 2







```
In [68]: import numpy
import tvb.datatypes.time_series

#Create a tvb TimeSeries object
tsr = tvb.datatypes.time_series.TimeSeriesRegion()
tsr.data = TAVG
tsr.sample_period = 0.001

#Create and launch the interactive visualiser
import tvb.simulator.plot.power_spectra_interactive as ps_int
psi = ps_int.PowerSpectraInteractive(time_series=tsr)
```



```
In [69]: psi.show()
```

```
-----  
TraitAttributeError                                Traceback (most recent call last)  
<ipython-input-69-6e073a744135> in <module>  
----> 1 psi.show()  
  
~\Downloads\TVB_Distribution\ tvb_data\ Lib\ site-packages\ tvb\ simulator\ plot\ power_spectra_interactive.py in show(self)  
    170         """ Generate the interactive power-spectra figure. """  
    171         #Make sure everything is configured  
--> 172         self.configure()  
    173  
    174         #Make the figure:  
  
~\Downloads\TVB_Distribution\ tvb_data\ Lib\ site-packages\ tvb\ simulator\ plot\ power_spectra_interactive.py in configure(self)  
    154     def configure(self):  
    155         """ Seperate configure cause ttraits be busted... """  
--> 156         LOG.debug("time_series shape: %s" % str(self.time_series.data.shape))  
    157         #TODO: if isinstance(self.time_series, TimeSeriesSurface) and self.first_n == -1: #LOG.error, return.  
    158         self.data = self.time_series.data[:, :, :self.first_n, :]  
  
~\Downloads\TVB_Distribution\ tvb_data\ Lib\ site-packages\ tvb\ basic\ neutraits\ _attr.py in __get__(self, instance, owner)  
    168         if self.required and value is None:  
    169             raise TraitAttributeError('required attribute referenced before assignment. '  
--> 170                                     'Use a default or assign a value before reading it', attr=self)  
    171         return value  
    172  
  
TraitAttributeError: required attribute referenced before assignment. Use a default or assign a value before reading it  
attribute tvb.simulator.plot.power_spectra_interactive.PowerSpectraInteractive.time_series = Attr(field_type=<class  
'tvb.datatypes.time_series.TimeSeries'>, default=None, required=True)
```

In [67]: whos

Variable	Type	Data/Info
C	ndarray	1x50x1: 50 elems, type `float64`, 400 bytes
S	ndarray	1x50x1: 50 elems, type `float64`, 400 bytes
TAVG	ndarray	60000x76: 4560000 elems, type `float64`, 36480000 bytes (34.790039
0625 Mb)		
TimeSeriesRegion	MetaType	<class 'tvb.datatypes.tim<...>series.TimeSeriesRegion'>
TvbProfile	type	<class 'tvb.basic.profile.TvbProfile'>
cb	Colorbar	<matplotlib.colorbar.Colo<...>ct at 0x00000129602B4208>
con	Connectivity	Connectivity (\n Number <...>... [0.00462632, 2, 3]\n)
configure	function	<function configure at 0x000001295E0404C8>
connectivity	module	<module 'tvb.datatypes.co<...>atypes\\connectivity.py'>
cortex	module	<module 'tvb.datatypes.co<...>b\\datatypes\\cortex.py'>
coupling	module	<module 'tvb.simulator.co<...>\\simulator\\coupling.py'>
dS	ndarray	1x50x1: 50 elems, type `float64`, 400 bytes
data	ndarray	60000x1x76x1: 4560000 elems, type `float64`, 36480000 bytes (34.79
00390625 Mb)		
equations	module	<module 'tvb.datatypes.eq<...>datatypes\\equations.py'>
fig1	Figure	Figure(1500x1500)
fig2	Figure	Figure(1500x1500)
get_logger	function	<function get_logger at 0x0000012959F80318>
hinton_diagram	function	<function hinton_diagram at 0x000001295E1129D8>
integrators	module	<module 'tvb.simulator.in<...>mulator\\integrators.py'>
local_connectivity	module	<module 'tvb.datatypes.lo<...>\\local_connectivity.py'>
log_debug	function	<function log_debug at 0x0000012959F44CA8>
mod	ReducedWongWang	ReducedWongWang (\n I_o <...>..... (1,)\n)
models	_Module	<tvb.simulator.models._de<...>ct at 0x000001295B494B08>
monitors	module	<module 'tvb.simulator.mo<...>\\simulator\\monitors.py'>
noise	module	<module 'tvb.simulator.no<...>vb\\simulator\\noise.py'>
nregions	int	76
os	module	<module 'os' from 'C:\\Us<...>n\\tvb_data\\Lib\\os.py'>
patterns	module	<module 'tvb.datatypes.pa<...>\\datatypes\\patterns.py'>
period	float	0.00048828125
plot_3d_centres	function	<function plot_3d_centres at 0x000001295E142E58>
plot_connectivity	function	<function plot_connectivi<...>ty at 0x000001295E112948>
plot_fast_kde	function	<function plot_fast_kde at 0x000001295E142F78>
plot_local_connectivity	function	<function plot_local_conn<...>ty at 0x000001295E139438>
plot_matrix	function	<function plot_matrix at 0x000001295E13E708>
plot_pattern	function	<function plot_pattern at 0x000001295E139C18>

plot_tri_matrix	function	<function plot_tri_matrix at 0x000001295E142EE8>
ps_int	module	<module 'tvb.simulator.pl<...>_spectra_interactive.py'>
psi	PowerSpectraInteractive	<object with id 1277254928136 (str() failed)>
region_mapping	module	<module 'tvb.datatypes.re<...>ypes\\region_mapping.py'>
sensors	module	<module 'tvb.datatypes.se<...>\\datatypes\\sensors.py'>
sim	Simulator	Simulator (\n Type .....<...>428d-bee0-51e2892e1d5b\n)
simulator	module	<module 'tvb.simulator.si<...>simulator\\simulator.py'>
surfaces	module	<module 'tvb.datatypes.su<...>\\datatypes\\surfaces.py'>
tavg	list	n=1
tavg_data	ndarray	60000x1x76x1: 4560000 elems, type `float64`, 36480000 bytes (34.79
00390625 Mb)		
tavg_time	list	n=60000
tic	float	1629104639.7249508
time	ndarray	60000: 60000 elems, type `float64`, 480000 bytes (468.75 kb)
time_series	module	<module 'tvb.datatypes.ti<...>tatypes\\time_series.py'>
tm	module	<module 'time' (built-in)>
ts_int	module	<module 'tvb.simulator.pl<...>meseries_interactive.py'>
tsr	TimeSeriesRegion	<object with id 1277254871624 (str() failed)>
tvb	module	<module 'tvb' from 'C:\\U<...>kages\\tvb\\__init__.py'>

