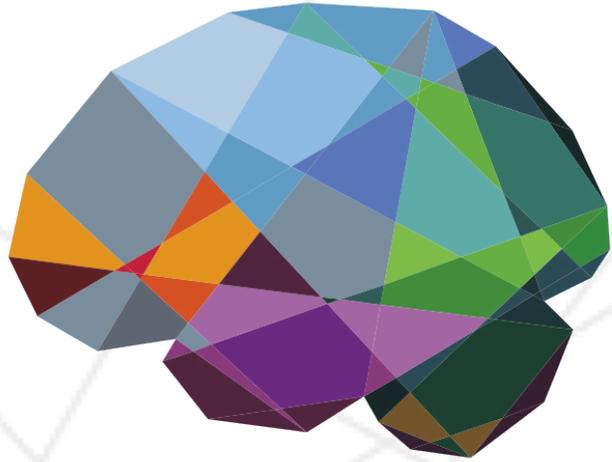


NODE#1
HAMBURG



THE VIRTUAL BRAIN.

June 7, 2014

NODE #1
HAMBURG



THE VIRTUAL BRAIN.

Hands-on Session #1

Building Your Own Brain Network Model

June 7, 2014

TVB allows for a systematic exploration and manipulation of every underlying component of a large-scale brain network model, such as the neural mass model governing the local dynamics ¹ or the structural connectivity constraining the space-time structure of the network couplings.

¹ There are a number of predefined models available in TVB

Objectives

THIS TUTORIAL PRESENTS the basic anatomy of a brain network model at the region level using The Virtual Brain's (TVB's) graphical interface. You are not expected to launch all the simulations. However, following these steps you should be able to reproduce the results from the simulations in the project Building Your Own Brain Network Model.

Project: *Session_I_BuildingYourOwnBrainNetworkModel*

IN THIS PROJECT, all the data were already generated. We'll only go through the necessary steps required to reproduce some simulations listed in Table 1. You can always start over, click along and/or try to change parameters.

Building A Discrete Brain Network Model

A BASIC SIMULATION at the region level uses a coarse representation of the brain and consists of five main components, each of these components is a configurable object in TVB:

- Model or Local population model, which is, at its core, a set of differential equations describing the local neuronal dynamic;
- Connectivity, represents the large scale structural connectivity of the brain, ie white-matter tracts;
- Long range Coupling, is a function that is used to join the local Model dynamics at distinct locations over the connections described in Connectivity;
- Integrator, is the integration scheme that will be applied to the coupled set of differential equations;
- Monitors, one or more Monitors can be attached to a simulation, they act to record the output data.

Name
Building A Discrete Brain Network Model <i>AnatomyOfARegionSimulation_a</i> <i>AnatomyOfARegionSimulation_b</i>
Looking At The Results
Parameter Space Exploration <i>AnatomyOfARegionSimulation_pse</i>
Simulation Continuation or branching <i>AnatomyOfARegionSimulation_a_branch1</i>
Stochastic Simulations <i>AnatomyOfARegionSimulation_c</i>
Modelling the Neural Activity On the Folded Cortex <i>AnatomyOfASurfaceSimulation</i> <i>AnatomyOfASurfaceSimulation_branch1</i>
Define Your Own Local Connectivity <i>SurfaceSimulation_GaussianLocalConnectivity</i> <i>SurfaceSimulation_MexicanHatLocalConnectivity</i>

Table 1: Simulations in this project.

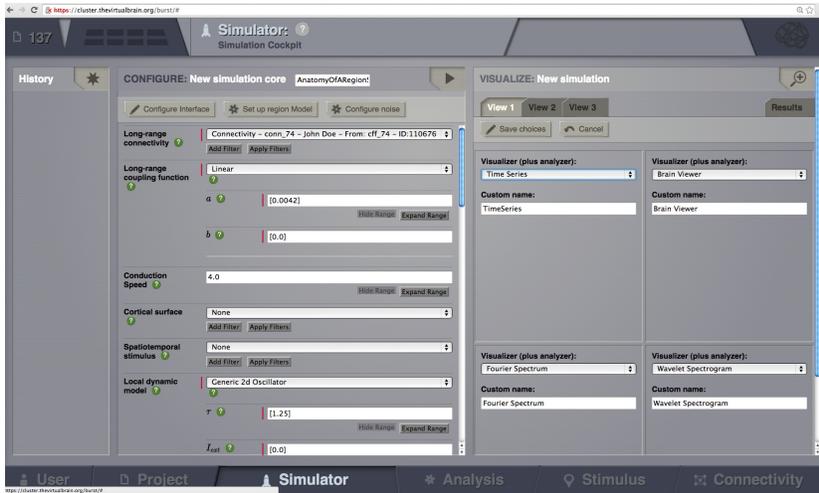
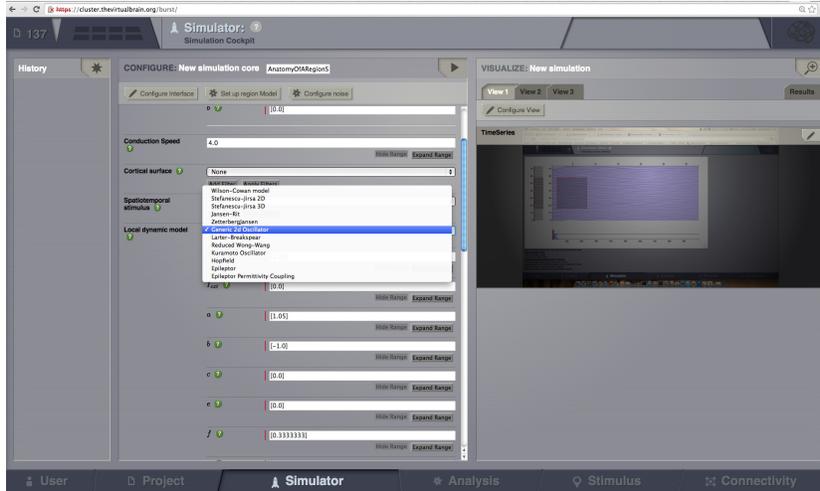


Figure 1: TVB's SIMULATOR page

In this example, *AnatomyOfARegionSimulation_a*, we will change the default parameters and configure some visualizers on the right column to have a quick idea of some properties of the simulated data.

1. Go to the SIMULATOR page. (Fig. 1)
2. Connectivity: Define some structure for your network. Here, we'll rely on **TVB's default matrix**.
3. Set a Long range coupling function. For our present purposes, we happen to know that for the parameters we will use through TVB's default Connectivity matrix, a linear function with a slope of $a = 0.0042$ is a reasonable thing to use.
4. Conduction speed: Alter the speed of signal propagation through the network to 4 mm/ms .
5. Local dynamics: Then define a Model for the local dynamics. Here we'll use the **generic 2 dimensional oscillator** (see Fig. 2) with the parameters shown in Table 2.
6. Now that we've defined our structure and dynamics we need to select an integration scheme. We'll use **HeunDeterministic**. The most important thing here is to use a step size that is small enough for the integration to be numerically stable. Here, we chose a value of $dt = 0.1 \text{ ms}$.
7. Select the **Temporal Average** monitor. It averages over a time window of length sampling period returning one time point every period. It also, by default, only returns those state-variables flagged in the Models definition as Variables watched by Monitors. For our example the Monitor's sampling period is **1 ms**.

Figure 2: Selecting a Model

Although there are Monitors which apply a biophysical measurement process to the simulated neural activity, such as EEG, MEG, SEEG, etc., here we'll select only one simple monitor just to show the idea. The Raw Monitor takes no arguments and simply returns all the simulated data.²

8. Provide the simulation length, that here we'll use the default value of **1000 ms**.
9. Before launching the simulation, configure a set of Visualizers and/or Analyzers by clicking on Configure, selecting the what you want to see and saving your choices. (see Table 3). These windows will enable you to have a glimpse of the results as soon as the simulation ends.
10. Enter a name for the current simulation (e.g., *AnatomyOfARegionSimulation_a*) and click on ►.

Looking at the Results

THE TRANSIENT LARGE amplitude oscillatory activity at the beginning of the simulation (see Fig. 3) is a result of the imperfectly set initial conditions.

The initial history (i.e., initial conditions) is merely set by default to be random walks within the general range of state-variable values expected from the model. As the current simulation is configured with fixed point dynamics, if

Model parameter	Value
a	1.05
b	-1.0
c	0
d	0.1
e	0
f	1/3
g	1
I	0
α	1
β	0.2
γ	-1
τ	1.25

Table 2: Generic 2d oscillator parameters

² As a general rule this monitor shouldn't be used for anything but very short simulations as the amount of data returned can become prohibitively large.

Window	Visualizer/Analyzer
top-left	Brain Viewer
top-right	Fourier Transform
bottom	Wavelet Spectrogram

Table 3: Selected visualizers. It is possible to configure up to 12 different windows.

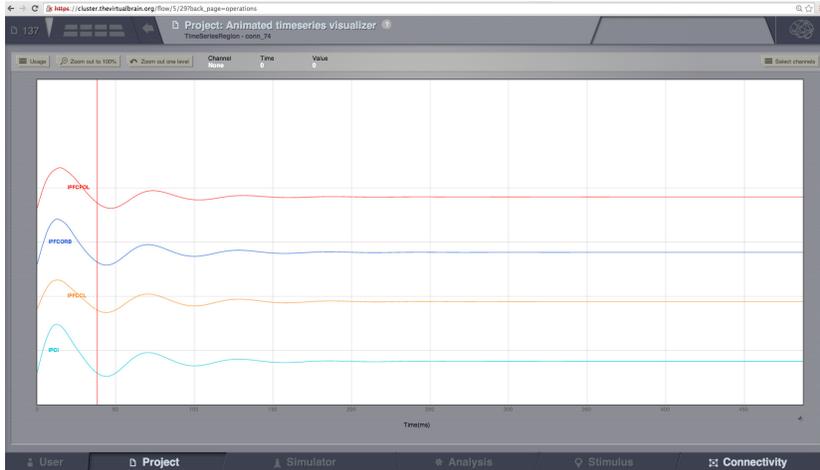


Figure 3: Time-series from *AnatomyOfARegionSimulation_a*

we were to set the initial conditions exactly to the values corresponding to that fixed point there would be no such initial transient (we will see how to achieve that later on).

1. Go to PROJECTS → OPERATIONS DASHBOARD.
2. Click on the icon of the time-series . From the metadata overlay's visualizers tab, launch the Animated Time Series Visualizer.
3. Go back to the SIMULATOR page and check the Fourier spectrum. Select a linear scale on the Y axis. We see that the intrinsic frequency of the oscillations is set at about 11 Hz (see Fig. 4).

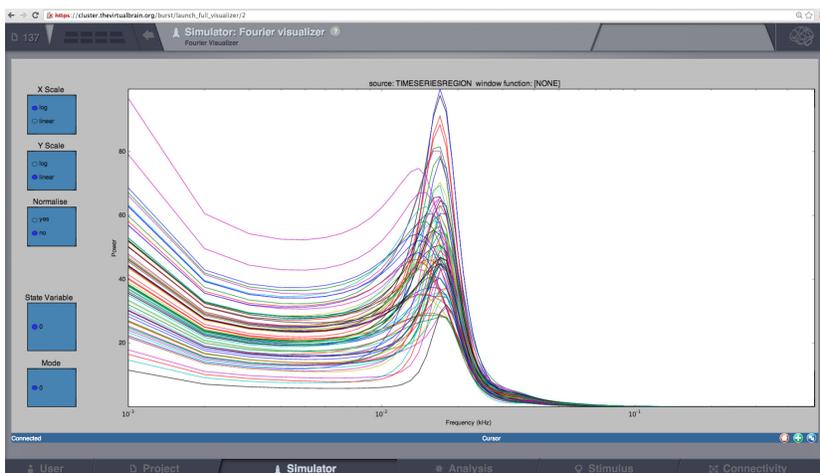


Figure 4: Fourier spectra of the time-series from *AnatomyOfARegionSimulation_a*

Figure 5: Copy a simulation.

1. Now let's have a look at a second simulation, which has the same parameters as *AnatomyOfARegionSimulation_a* except that the coupling strength has been increased by an order of magnitude. Hence, the slope of the linear coupling function is $a = 0.042$.
2. To make things easy, we copy the first simulation by clicking on  on the top right corner of a simulation tab. From the menu you can get a copy, edit the name the simulation or delete it. (Fig. 5).
3. Change the name of the new simulation (e.g., *AnatomyOfARegionSimulation_b*) and set the coupling strength to the value in step 1. Launch the simulation.

Looking at the time series of *AnatomyOfARegionSimulation_b*, we can see that the system exhibits self-sustained oscillations.

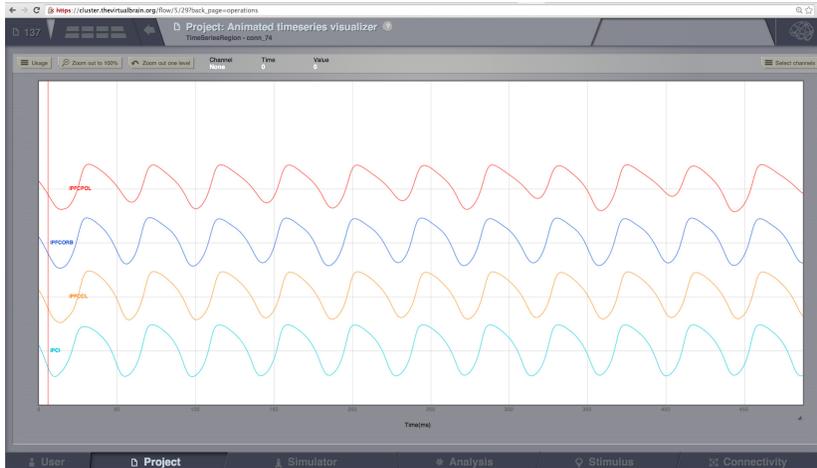


Figure 6: Time-series from *AnatomyOfARegionSimulation_b*

A frequent question is at which value of coupling strength this "bifurcation" occurs. Well, we can easily set up a parameter search by defining a range of values that will be explored. We'll see how to do this in the next section.

Parameter Space Exploration (PSE)

TVB WILL LAUNCH A SIMULATION for every value. ³ The example is set up in *AnatomyOfARegionSimulation_pse*.

1. In Long range coupling function, under **a**, click on **Expand Range**. Set the range between **0.012** and **0.042** and the step to **0.002**.
2. Do the same under conduction speed, setting the range between **1 – 10** mm/ms and the step to **1** mm/ms.
3. Set the simulation length to **2000** ms and launch the simulations.

All the 150 simulations are presented as a discrete 2D map in Parameter Space Discrete ⁴ (Fig. 7) or a continuous pseudocolor map in Parameter Space Continuous ⁵ (Fig. 8).

These results are those presented in [Ghosh et al., 2008] and [Knock et al., 2009].

Simulation continuation or Branching

OTHER PARAMETERS COULD BE adjusted as well. We mentioned before that the big transient at the beginning of the time-series is due to the initial conditions. To overcome this issue we have a couple of alternatives. First, we could narrow the range of the state variables around the values of a fixed point. How can we know this value? ⁶.

1. Click on Set up region model, you'll be redirected to a new working area.

In this area there's a an interactive tool, the Phase Plane, which allows you to understand the local dynamics, that is the dynamics of a single isolated node, by observing how the model parameters change its phase plane. (See Fig. 9).

2. Click on any point of the phase plane. A trajectory will be drawn. We see that the fixed point is approx $(V, W) = (1.5, -0.6)$

However, there certainly is a more elegant way.

³ For the time being, parameter space explorations are restricted to two dimensions.

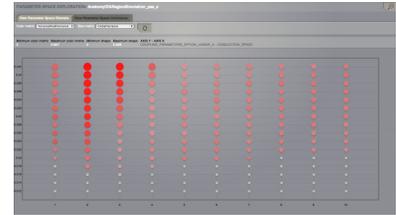


Figure 7: Discrete parameter space map from *AnatomyOfARegionSimulation_pse*

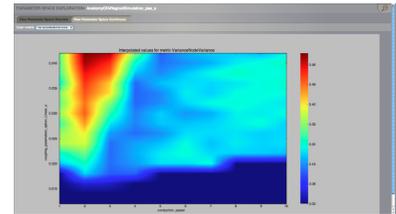


Figure 8: Continuous parameter space map from *AnatomyOfARegionSimulation_pse*

⁴ Each point represents a simulation. The size and the colour of the marker correspond to one of a few variance metrics implemented to have an idea of the network node and temporal variability.

⁵ The colour scale corresponds to one of the variance metrics

⁶ go back to the **SIMULATOR** page and load either *AnatomyOfARegionSimulation_a* or *AnatomyOfARegionSimulation_b*

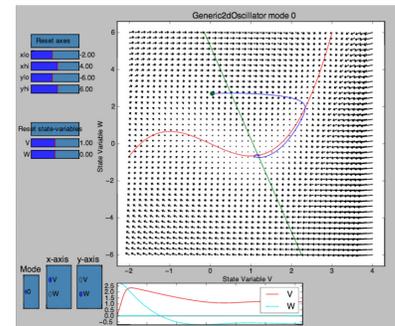


Figure 9: Phase Plane Interactive

3. Set your model with fixed point dynamics and a weak coupling strength (e.g., *AnatomyOfARegionSimulation_a*)
4. Run a simulation for **1000 ms**.

TVB has a branching mechanism that allows you to use the data of a simulation, as the initial history for a new simulation. The only thing you need to know is that the spatio-temporal structure of the network should remain unchanged (e.g., the number of nodes, conduction speed, the recorded state-variables, integration time-step size and selected monitors should be the same.)

- In *AnatomyOfARegionSimulation_a*, set **a = 0.042** in the long range coupling function. Then, click on .

AnatomyOfARegionSimulation_a_branch1 is an example of this functionality, using the results from *AnatomyOfARegionSimulation_a* as initial conditions.

Stochastic Simulations

As a last point, we will show the basics of running a simulation driven by noise (i.e., using a stochastic integration scheme⁷). Here we'll also use a region level simulation, but the considerations for surface simulations are the same. In a stochastic integration scheme Noise enters through the integration scheme.

⁷ We can choose between Additive and Multiplicative Noise functions when defining the way noise enters our simulation. In the case of Multiplicative Noise the function depends on the state of the system.

Here we'll define a simple constant level of noise that enters all nodes and all state variables, however, the noise is configurable on a per node and per state variable level, and as such the noise can be reconfigured to, for example, only enter appropriate state variables of certain thalamic nodes, thus emulating a very crude model of external inputs to the brain.

1. After configuring a model similar to the one presented in *AnatomyOfARegionSimulation_b*, we select **HeunStochastic** as an integration scheme.

The Noise functions are fed by a random process generated by a pseudo-random number generator (PRNG). The random processes used have Gaussian amplitude and can potentially be given a temporal correlation. The random process is defined using two parameters plus the seed of the PRNG. The two parameters are: **D**⁸, defining

⁸ As a general rule of thumb, the noise amplitude, when being applied to all nodes and all state-variables, should be much less than 1% of the expected state-variable range if you don't want the noise to dominate the intrinsic dynamics of the system. If you are not sure about the level of noise, using the Phase Plane Interactive tool for plotting noisy trajectories is a good place to start. See Fig. 10

the standard deviation of the noise amplitude; and τ which defines the correlation time of the noise source, with $\tau = 0$ corresponding to white noise and any value greater than zero producing coloured noise.

2. After configuring a model similar to the one presented in *AnatomyOfARegionSimulation_b*, we select **HeunStochastic** as our integration scheme.
3. Set the values for $\tau = 0$ and **seed=42**.
4. Set the noise dispersion, **D = 0.005**

AnatomyOfARegionSimulation_b and *AnatomyOfARegionSimulation_stochastic* have the same parameters but the latter has an extra background noisy input.

Observe the differences using the Spectrogram of the Wavelet Transform. Figs. 11 and 12

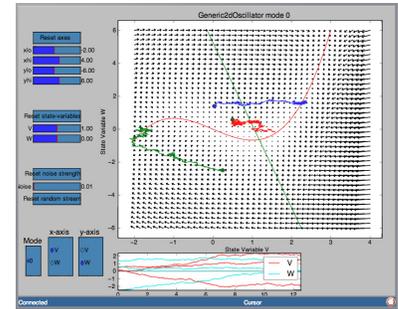


Figure 10: Stochastic trajectories

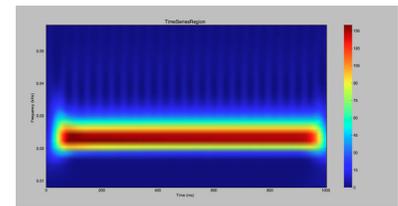


Figure 11: Spectrogram of the wavelet transform from *AnatomyOfARegionSimulation_b*

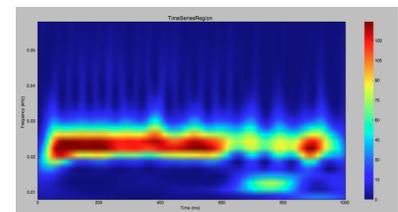


Figure 12: Spectrogram of the wavelet transform from *AnatomyOfARegionSimulation_c*

Modelling the Neural Activity on the Folded Cortex

This extends the basic region simulation to include the folded cortical surface to the anatomical structure on which the simulation is based. If you haven't read or followed what was written above you probably should do that now as here we only really discuss in detail the extra components that are specific to a simulation on the cortical surface.

In addition to the components discussed for a region simulation here we introduce two major components, that is:

- Cortical Surface, which is a mesh surface defining a 2d representation of the convoluted cortical surface embedded in 3d space.
- Local Connectivity, that represents the probability of the interactions between neighbouring nodes on a local patch.
- Region Mapping, a breakup that defines to which anatomical region in the Connectivity each vertex of the mesh belongs to.

1. The connectivity, speed, coupling strength and local model and its parameters are the same described in *AnatomyOfARegionSimulation_b* and Table 2.
2. Select the TVB's **default Cortical Surface**, which has 16384 nodes.
3. We rely on TVB's **default Local Connectivity**.
4. Rescale the Local Connectivity with Local coupling strength equal to **0.1**.
5. For the integration we'll use **HeunDeterministic**. Here, integration time step size is the default: **dt = 0.1220703125 ms**.

The first significant thing to note about surface simulations is that certain Monitors make a lot more sense in this context than they do at the region level, and so we'll introduce a couple new Monitors here.

6. The first of these new Monitors is called **SpatialAverage**. To select several monitors press the key Command or Control while you select them.
7. The second of these new monitors, which is an instantiation of a biophysical measurement process, is called **EEG**. The third will be the **Temporal Average**.

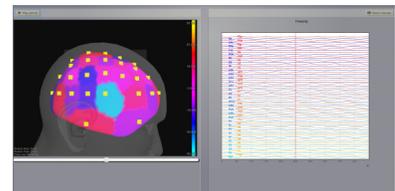


Figure 13: EEG signals from *SurfaceSimulation_MexicanHatLocalConnectivity*

8. The Monitors period is left with the default value **1.953125 ms** which is equivalent to a sampling frequency of 256 Hz.
9. Lastly, the simulation length is **500 ms**.
10. Run the simulation.
11. Once the simulation is finished, without changing any parameters, click on .

These simulations are *AnatomyOfASurfaceSimulation* and *AnatomyOfASurfaceSimulation_branch1*.

The first of these new Monitors, will average over the space (nodes) of the simulation.⁹ The basic mechanism is general, in the sense that the nodes can be broken up into any non-overlapping, complete, set of sets. In other words, each node can only be counted in one collection and all nodes must be in one collection.¹⁰

The second of these new Monitors, EEG, hopefully also unsurprisingly, returns the EEG signals resulting from the simulated neural dynamics using in the process a lead-field or Projection Matrix.¹¹

EEG signals measured on the scalp depend strongly on the location and orientation of the underlying neural sources, which is why this monitor is more realistic and useful in the case of surface based simulations – where the simulation is run on the explicit geometry of the cortex, which can potentially have been obtained from a specific individual's brain. In addition a simulation being built on the specific anatomical structure of an individual subject, the specific electrodes used in experimental work can also be incorporated, providing a link between simulation and experiment. See Fig. 13

⁹ In the case of region level simulations we already have a situation of a relatively small number of nodes, with each one representing a fairly large chunk of brain. In surface simulations on the other hand we can easily have tens of thousands of nodes, and reducing this by averaging over sensible collections of these nodes can be valuable.

¹⁰ In fact this is the default behaviour of the Spatial Average monitor when applied to a surface simulation, that is it averages over nodes and returns region based time-series.

¹¹ The EEG monitor will return a relatively standard 62 channel set based on the 10-20 system.

Define Your Own Local Connectivity

The regularized mesh can support, in principle, arbitrary forms for the local connectivity kernel. Coupled across the realistic surface geometry this allows for a detailed investigation of the local connectivity's effects on larger scale dynamics modelled by neural fields.

1. Go to CONNECTIVITY → LOCAL CONNECTIVITY. In this area we'll build two different kernels: a Gaussian and a Mexican Hat kernel. We'll start with the Gaussian kernel.
2. Select the equation defining the spatial profile of your local connectivity. Here, we'll set **sigma** to **15 mm**.

3. Ideally, you want the function to have essentially dropped to zero by the **cutoff distance**. The cutoff distance, that is, the distance up to which a given node is connected to its neighbourhood [Spiegler and Jirsa, 2013, Sanz-Leon et al., 2014] is set to **40 mm**. See Fig. 14.
4. Name your Local connectivity and save it by clicking on Create new Local Connectivity on the bottom left corner.

This data structure is saved under the name *LocalConnectivity_Gaussian_zc_40*.

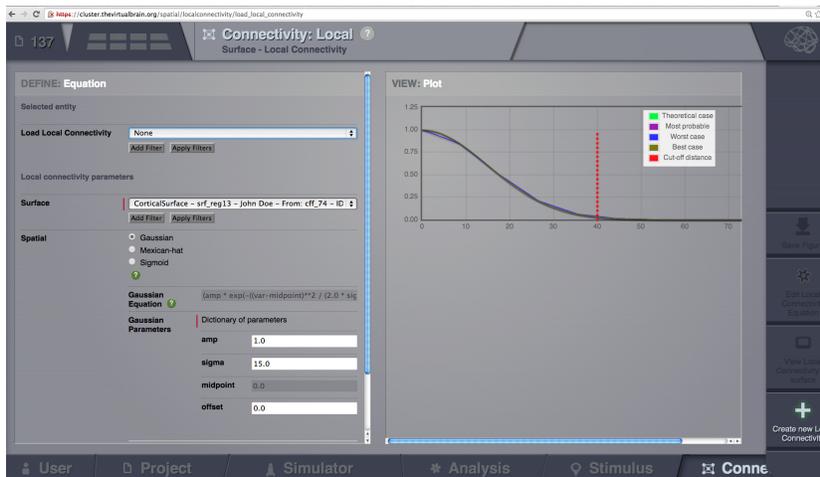


Figure 14: Gaussian local connectivity. We have on the right a representation of the kernel for a quick idea of how the Local Connectivity we've just specified will be represented on the surface. This plots the local connectivity function with different sampling based on the distribution of edge lengths in your mesh surface. If all the lines don't, at least mostly, overlap then you've probably specified a function with structure that is too fine for the resolution of your mesh surface.

5. Select the Mexican Hat equation. Here, we changed the default parameters. See the values in Table 4.
6. Save your new local connectivity.

This data structure is saved under the name *LocalConnectivity_MexicanHat_zc_40*.

Finally, we will run two more simulations using different local connectivity kernels.

Parameter	Value
midpoint_1	0 mm
midpoint_2	0 mm
amp_1	2 au
amp_2	1 au
sigma_1	5 mm
sigma_2	15 mm
cutoff distance	40 mm

Table 4: *LocalConnectivity_MexicanHat_zc_40* parameters.

5. Copy *AnatomyOfASurfaceSimulation*.
6. Change the **local connectivity** to *LocalConnectivity_Gaussian_zc_40* and set the **local connectivity strength** to **0.001**. Run the simulation.
7. Copy again *AnatomyOfASurfaceSimulation*.
8. This time select *LocalConnectivity_MexicanHat_zc_40*. The **local connectivity strength** is set to **-0.001**. Run the simulation.

These results are those of *SurfaceSimulation_MexicanHatLocalConnectivity* and *SurfaceSimulation_GaussianLocalConnectivity* respectively.

More Documentation

And that's it for this session, while the simulations are not particularly scientifically interesting, hopefully it gave you a sense of the anatomy of a simulation within TVB and many of the configurable parameters and output modalities. Online help is available clicking on the  icons next to each entry. For more documentation on The Virtual Brain, please see the following articles [Ghosh et al., 2008, Sanz-Leon et al., 2013, Spiegler and Jirsa, 2013, Woodman et al., 2014, Jirsa et al., 2010]

Support

The official TVB website is www.thevirtualbrain.org. All the documentation and tutorials are hosted on [the-virtual-brain.github.io](https://github.com/the-virtual-brain). You'll find our public GIT repository at <https://github.com/the-virtual-brain>. For questions and bug reports we have a users group <https://groups.google.com/forum/#!forum/tvb-users>

References

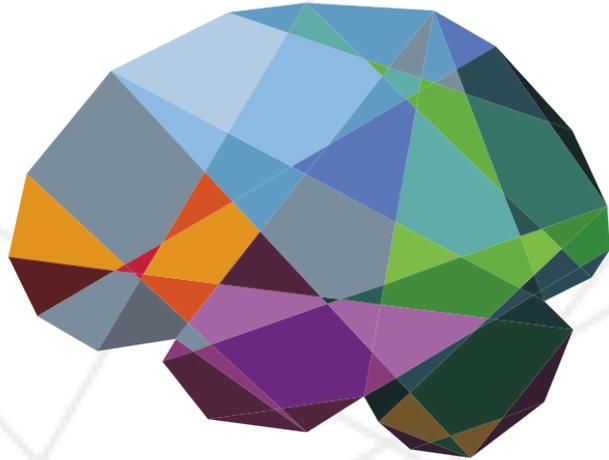
- A. Ghosh, Y. Rho, A. R. McIntosh, R. Kötter, and V. K. Jirsa. Noise during rest enables the exploration of the brain's dynamic repertoire. *PLoS Comput. Biol.*, 4(10):e1000196–e1000196, October 2008. DOI: [doi:10.1371/journal.pcbi.1000196](https://doi.org/10.1371/journal.pcbi.1000196).
- V. K. Jirsa, O. Sporns, M. Breakspear, G. Deco, and A. R. McIntosh. Towards the virtual brain: network modeling of the intact and the damaged brain. *Arch. Ital. Biol.*, 148(3):189–205, Sep 2010.
- S. A. Knock, A. R. McIntosh, O. Sporns, R. Kötter, P. Hagmann, and V. K. Jirsa. The effects of physiologically plausible connectivity structure on local and global dynamics in large scale brain models. *J. Neurosci. Methods*, 183(1):86–94, September 2009.
- P. Sanz-Leon, S. A. Knock, M. M. Woodman, L. Domide, J. Mersmann, A. R. McIntosh, and V. K. Jirsa. The virtual brain: a simulator of primate brain network dynamics. *Front Neuroinform*, 7:10, 2013. DOI: [10.3389/fninf.2013.00010](https://doi.org/10.3389/fninf.2013.00010). URL <http://dx.doi.org/10.3389/fninf.2013.00010>.

P. Sanz-Leon, S. A. Knock, A. Spiegler, and V. K. Jirsa. Mathematical framework for large-scale brain network modelling in the virtual brain. *Neuroimage (submitted)*, --, 2014.

A. Spiegler and V. Jirsa. Systematic approximations of neural fields through networks of neural masses in the virtual brain. *Neuroimage*, 83C:704–725, Jun 2013. DOI: 10.1016/j.neuroimage.2013.06.018. URL <http://dx.doi.org/10.1016/j.neuroimage.2013.06.018>.

M. M. Woodman, L. Pezard, L. Domide, S. A. Knock, P. Sanz-Leon, A. R. McIntosh, J. Mersmann, and V. K. Jirsa. Integrating neuroinformatics tools in the virtual brain. *Frontiers in Neuroinformatics*, 8:36–, 2014. DOI: 10.3389/fninf.2014.00036 - See more at: <http://journal.frontiersin.org/Journal/10.3389/fninf.2014.00036>.

NODE #1
HAMBURG



THE VIRTUAL BRAIN.

Hands-on Session #2
Import, Link and Share Data.

June 7, 2014

Objectives

THIS SHORT TUTORIAL provides the basic steps you need to know in order to import data into TVB; share data with other users and/or link data to other TVB projects. After this session you should be able to export a project and share it with your neighbour.

Project: *Session_II_ShareAndLink_a*

There are two simulations in one of this projects as described in Table 1.

Multiple Users And One Virtual Brain

In this exercise we will assume that multiple users are working with the same TVB installation. ¹

1. By default you are the Admin user.
2. Change the admin email (Fig. 1).
3. Create a user for your neighbour by registering a new user. (Fig. 2).
4. You'll receive a notification email.
5. Make sure the new user has been validated. (Fig. 3).
6. Login with the Admin account.

Importing A Connectivity

1. Create a project (e.g. *Session_II_LinkAndShare_a*).
2. Create a second project (e.g. *Session_II_LinkAndShare_b*).
3. Assuming that you are working in the first project, upload a Connectivity in a zip file. This was already done but you can repeat these steps.
4. Go to **PROJECTS** → **DATA STRUCTURE**. Click on **Upload Data**. An overlay with the current supported formats will appear. See Fig. 4.
5. Select Connectivity ZIP.
6. Select the file *connectivity_regions_96.zip* found at *TVB_Distribution/tvb_data/*.

¹ Working in pairs might be a good idea.

Name
Multiple Users And One Virtual Brain
Importing A Connectivity
Link And Share
<i>SimulationConnectivity74</i>
<i>SimulationConnectivity96</i>
Export And Read

Table 1: Simulations in this project.



Figure 1: Change Admin email.



Figure 2: Create a new user.



Figure 3: Validate user.

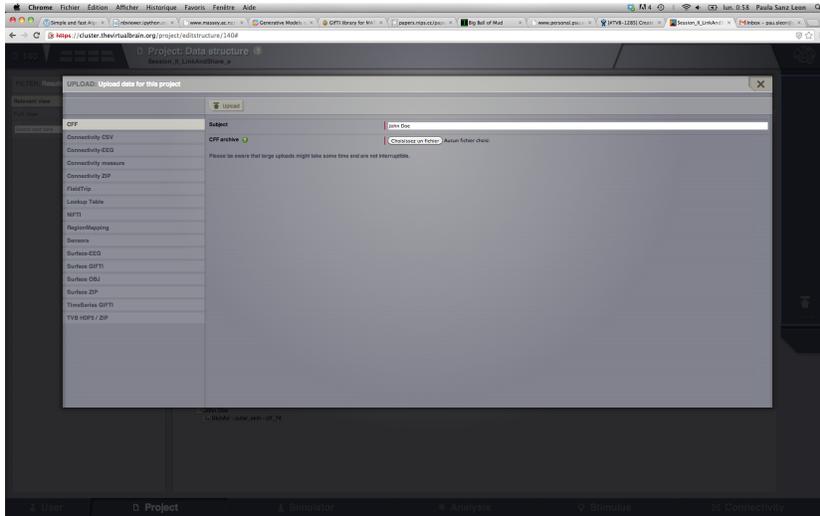


Figure 4: Supported data formats.

7. Add a personalized tag to this newly created datatype (e.g. *conn_96*). See Fig. 5.
8. Save the changes.

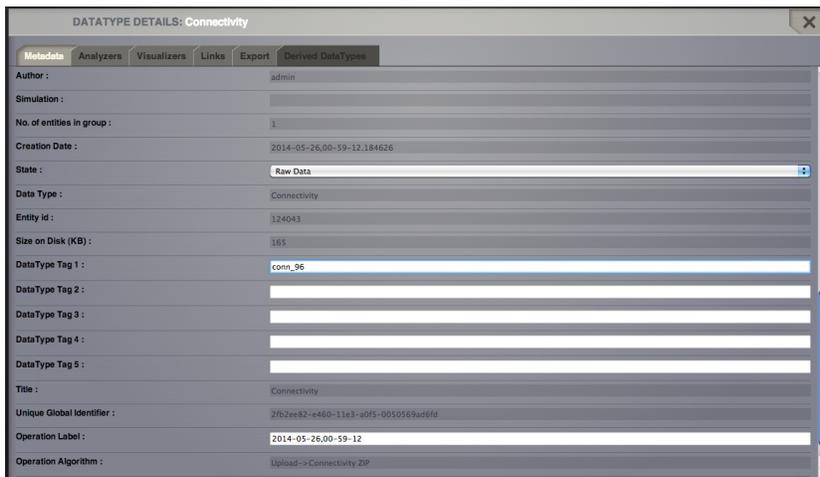


Figure 5: Add a personalized tag.

Link And Share

9. Select the connectivity you want to share.
10. In the metadata overlay, go to the tab Links. You'll see a list with all your projects. See Fig. 6.
11. Link this datatype (connectivity) with the project you'll share (e.g. *Session_II_ShareAndLink_b*).

12. Link the two time-series from simulations *SimulationConnectivity74* and *SimulationConnectivity96*.

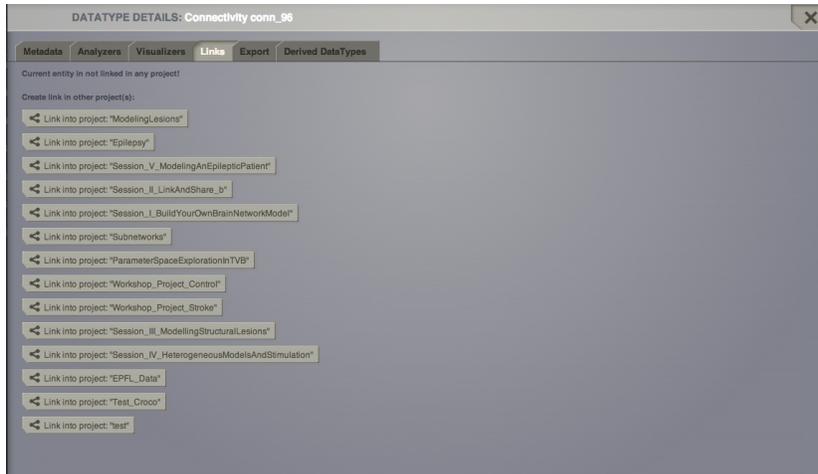


Figure 6: Links tab.

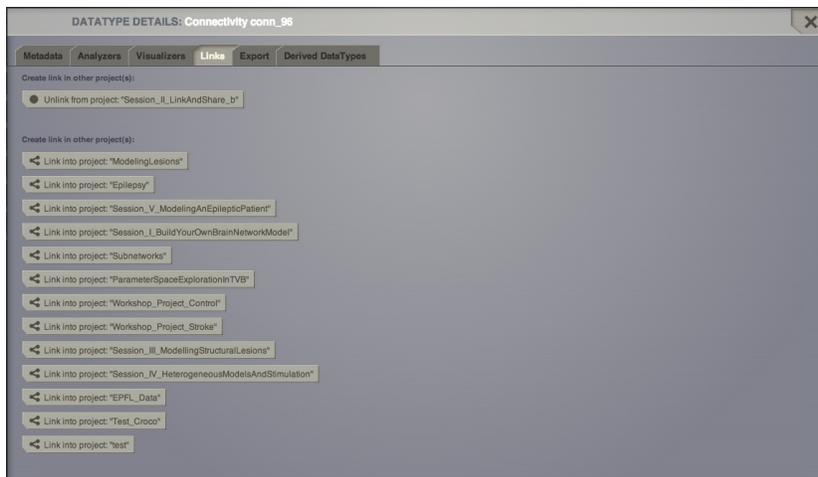


Figure 7: Link a datatype to another project.

13. Go to PROJECT → LIST OF ALL PROJECTS
14. Switch to *Session_II_ShareAndLink_b*.
15. Then from PROJECT → BASIC PROPERTIES share this project with your neighbour's user account.
16. Logout from your account and let your neighbour login to his/her account.

You should be able to see the connectivity matrix (and other datatypes) you linked from project *Session_II_ShareAndLink_a*.

Export and Read

1. Go to PROJECT → DATA STRUCTURE
2. Click on  from *TimeSeriesRegion - conn_74*.
3. From the overlay, Export tab, download the data in TVB format (h5).
4. Rename the file if you want (e.g. *LinkAndShare_TimeSeriesRegion*).

From an ipython shell you can follow the commands presented below. See Fig. 8

```
In [1]: import h5py
In [2]: import matplotlib.pyplot as plt

In [3]: f = h5py.File('LinkAndShare_TimeSeriesRegion.h5')

In [4]: f.keys()
Out[4]: [u'data', u'time']

In [5]: f.attrs.keys()
Out[5]:
[u'TVB_User_tag_1',
 u'TVB_Title',
 u'TVB_Length_2d',
 u'TVB_Gid',
 u'TVB_Length_3d',
 u'TVB_Sample_period_unit',
 u'TVB_Labels_ordering',
 u'TVB_Length_1d',
 u'TVB_User_tag_4',
 u'TVB_User_tag_5',
 u'TVB_Subject',
 u'TVB_Length_4d',
 u'TVB_Data_version',
 u'TVB_User_tag_3',
 u'TVB_Is_nan',
 u'TVB_Type',
```

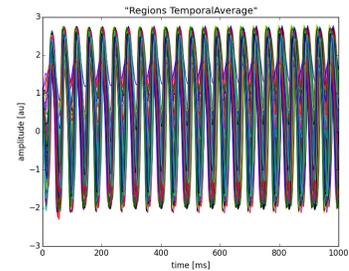


Figure 8: Plotting time-series with matplotlib.

```

u'TVB_Invalid',
u'TVB_Connectivity',
u'TVB_Create_date',
u'TVB_User_tag_2',
u'TVB_Labels_dimensions',
u'TVB_Sample_rate',
u'TVB_State',
u'TVB_Start_time',
u'TVB_Sample_period',
u'TVB_Nr_dimensions',
u'TVB_Visible',
u'TVB_Module']

In[6]: plt.plot(f['time'], f['data'][:, 0, :, 0])
...

In [7]: plt.xlabel('time [ms]')
Out[7]: <matplotlib.text.Text at 0x118e95310>

In [8]: plt.ylabel('amplitude [au]')
Out[8]: <matplotlib.text.Text at 0x118e9a190>

In [9]: plt.title(f.attrs['TVB_Title'])
Out[9]: <matplotlib.text.Text at 0x118eb0ad0>

```

In **Matlab** (see Fig. 9)²:

```

>> hinfo = hdf5info('LinkAndShare_TimeSeriesRegion.h5');
>> hinfo.GroupHierarchy.Datasets.Name
ans =
/data
ans =
/time
>> hinfo.GroupHierarchy.Attributes.Name
...
>> data = hdf5read(hinfo.GroupHierarchy.Datasets(1));
>> time = hdf5read(hinfo.GroupHierarchy.Datasets(2));
>> plot(time, squeeze(data))
>> xlabel('time [ms]')
>> ylabel('amplitude [au]')

```

² hdf5read will be removed in a future version. Use h5read instead.

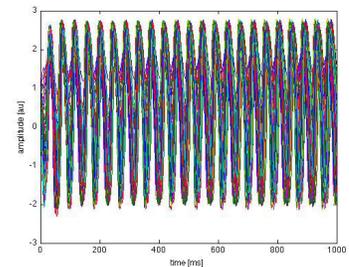


Figure 9: Plotting time-series with matlab.

In R (see Fig. 10)³:

```
> data <- h5read("/Users/paupau/GithubProjects/tvb-handbook/
tvbworkshop/LinkAndShare_TimeSeriesRegion.h5", "data")

> time <- h5read("/Users/paupau/GithubProjects/tvb-handbook/
tvbworkshop/LinkAndShare_TimeSeriesRegion.h5", "time")

> data = drop(mydata)

> plot(mytime, data[,1], type="l")
```

³ You need the rhdf5 package

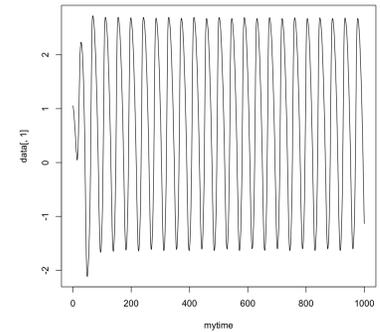


Figure 10: Plotting time-series with R.

More Documentation

For more details on technical aspects of The Virtual Brain, please see the following articles [Sanz-Leon et al., 2013, Woodman et al., 2014]

Support

The official TVB website is www.thevirtualbrain.org. All the documentation and tutorials are hosted on [the-virtual-brain.github.io](https://github.com/the-virtual-brain). You'll find our public GIT repository at <https://github.com/the-virtual-brain>. For questions and bug reports we have a users group <https://groups.google.com/forum/#!forum/tvb-users>

References

P. Sanz-Leon, S. A. Knock, M. M. Woodman, L. Domide, J. Mersmann, A. R. McIntosh, and V. K. Jirsa. The virtual brain: a simulator of primate brain network dynamics. *Front Neuroinform*, 7:10, 2013. DOI: 10.3389/fninf.2013.00010. URL <http://dx.doi.org/10.3389/fninf.2013.00010>.

M. M. Woodman, L. Pezard, L. Domide, S. A. Knock, P. Sanz-Leon, A. R. McIntosh, J. Mersmann, and V. K. Jirsa. Integrating neuroinformatics tools in thevirtualbrain. *Frontiers in Neuroinformatics*, 8:36:–, 2014. DOI: 10.3389/fninf.2014.00036 - See more at: <http://journal.frontiersin.org/Journal/10.3389/fninf.2014.00036>.

NODE #1
HAMBURG



THE VIRTUAL BRAIN.

Hands-on Session #3
Modelling Structural Lesions
June 7, 2014

Altered long-range connectivity and large-scale brain network dynamics are expected to shed light in characterizing and understanding the so-called network disorders.

Objectives

FROM A SCIENTIFIC POINT OF VIEW, the aim is to obtain a better understanding of how structural lesions can be characterized; and more specifically, how those changes may modify the local fast dynamics of the neural activity, as well as the impact on global lower time scales signals such as BOLD. The main goal of this session is to provide a clear understanding of the available tools to explore, analyze and edit a connectome. Furthermore, in addition to TVB's default connectivity matrix we will have two additional matrices, one of a control subject and the other from a stroke patient.

Project: *Session_III_ModellingStructuralLesions*

IN THIS SESSION, because the execution times for BOLD simulations are long (in the order of hours or days), we'll only go through the necessary steps required to reproduce some of the simulations listed in Table 1. You can click along and/or try to change parameters and relaunch the simulations.

FIGURE 1 shows one of the most prominent working areas in TVB: the connectivity editor. In the same working area, several visualizers are available for displaying the matrices (e.g Matrix and Space-Time); and some of them may use node-wise metrics to set the colour and size of the nodes (e.g., Hemisphere 1, Transverse and Hemisphere 2). If you do not select any metric before launching the connectivity editor, you can always set these properties later.

Name
Removing Interhemispheric Connections
Removing Nodes Based On Network Metrics
The Effect Of The Structural Connectivity On The Global Stability Map <i>NoInterhemispheric_g2d_pse</i> <i>NoHighInStrength_g2d_pse</i> <i>NoHighInDegree_g2d_pse</i> <i>Default_g2d_pse</i>
Region-based Simulations Of A Control And Stroke Connectome <i>Control_g2d_pse</i> <i>Stroke_g2d_pse</i> <i>Control_g2d_init_bold_branch1</i> <i>Stroke_g2d_init_bold_branch1</i>
Region-based Simulations Of A Control And Stroke Connectome Using A Higher-dimensional Model <i>Control_sj3d_pse</i> <i>Stroke_sj3d_pse</i> <i>Control_sj3d_init_bold_branch1</i> <i>Stroke_sj3d_init_bold_branch2</i>

Table 1: Simulations in this project.

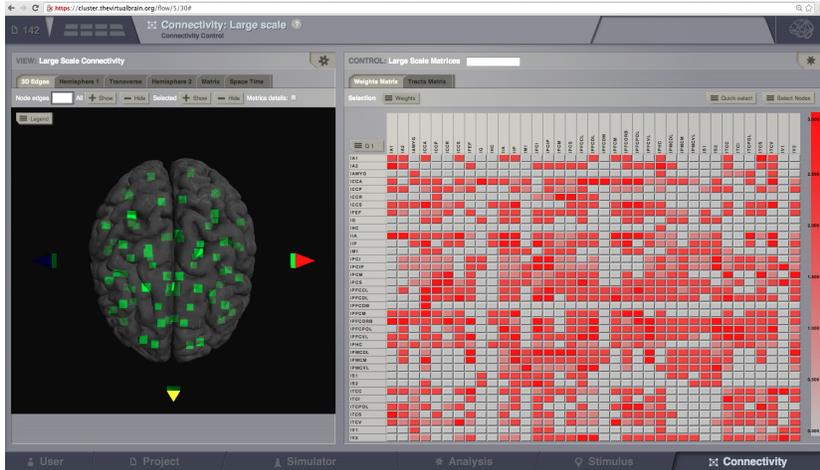


Figure 1: Long-range connectivity editor

Removing Interhemispheric Connections

AS A FIRST STEP, we will explore the TVB's default connectivity. Next, to model lesions, we will modify the connectivity, for instance, by deleting edges or nodes. Here, as a working example we will remove the interhemispheric connections.

1. Go to the **CONNECTIVITY** area → **LARGE SCALE**.
2. Select the the right nodes from the **Quick Select** menu and delete them (Fig. 2). Leave only the left nodes. (Fig. 3).
3. Apply the changes. The active or selected nodes will appear in green on the left **3D Edges**.
4. In the **Select Nodes** menu, write a name for the new selection and save the selection for later use. (Fig. 4). The connectivity editor will now be aware of two set of nodes: those in the selection (green) and those that have not been selected (white).

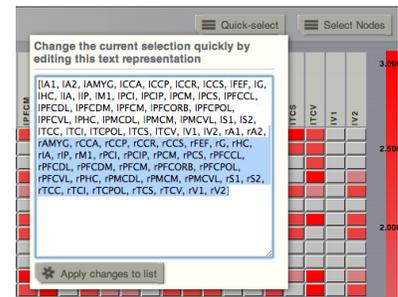


Figure 2: Remove the right nodes from the list.

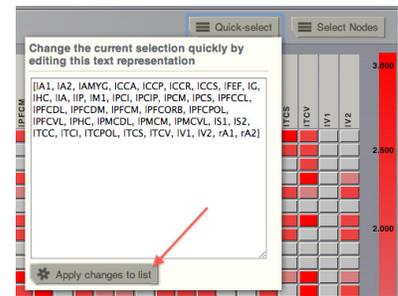


Figure 3: Select only the left nodes and apply the changes.

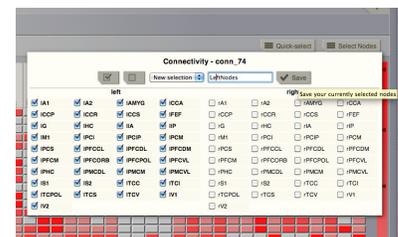


Figure 4: Save the left nodes selection.

From the quadrant selector, move to the third quadrant (Q3). You can zoom in the display on the left and select a node next to the medial plane. Then, draw a node's ingoing and/or outgoing connections by selecting a node and then right-clicking on it. From the menu shown in Fig. 5 it is possible to make the edges visible and set their colour.

Now we'll proceed to perform operations on the edge values.

5. Click on the Weights menu and select the operation **Set(n)** for edges OUT-IN. Set the value to 0 and then apply the changes.
6. Select operation **Set(n)** for edges IN-OUT, set the value to 0 and then apply the changes. (Fig. 6)

There are four category of edges depending on the nodes they connect:

- Edges IN-IN. These edges connect pair of nodes in the *selected set*. In this particular case you would modify the edges from the left hemisphere.
- Edges OUT-OUT. These edges connect pair of nodes in the *unselected set*. In this case they would be the right intrahemispheric edges.
- Edges IN-OUT. These are the edges that connect nodes in the selected set (rows) to nodes in the unselected set (columns). In this case it refers to the edges in the second quadrant (Q2).
- Edges OUT-IN. These edges connect nodes in the unselected set (rows) to nodes in the selected set (columns). In this case it refers to the edges in the third quadrant (Q3).

7. In the connectivity editor the interhemispheric connections are gone. Before you save the new connectivity matrix, make sure all the nodes are included, otherwise TVB will assume you only want the nodes in the current selection and will set the rest to zero. (Fig. 7).

8. Give a name to the new connectivity and click on  (Fig. 8)
9. The last action will take you back to the CONNECTIVITY → LONG-RANGE CONNECTIVITY. From this page, load the new matrix. (Fig. 9)

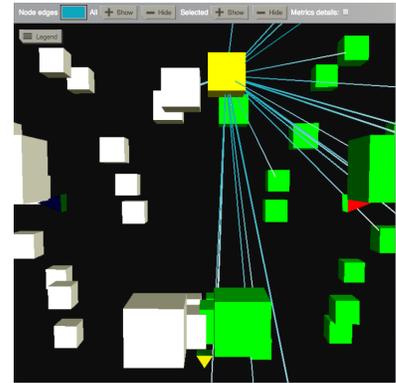


Figure 5: Draw connections.

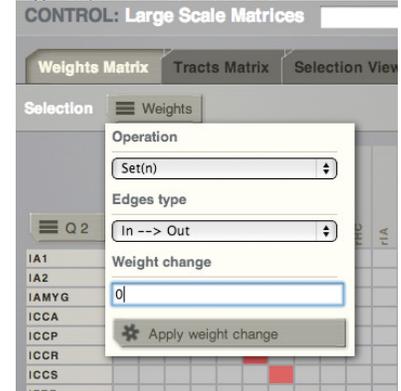


Figure 6: Set interhemispheric connections to 0



Figure 7: Select all nodes.



Figure 8: Save a new connectivity.

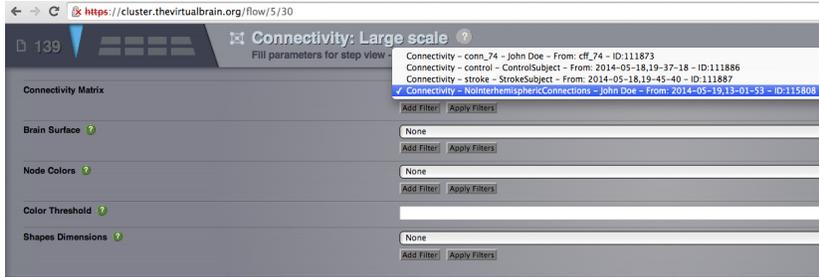


Figure 9: Load the newly created matrix.

Removing Nodes Based On Network Metrics

NEXT, WE'LL USE A CRITERION based on graph-metrics to select which nodes will be deleted. The metrics are computed using the BCT ¹ algorithms.

10. Go to ANALYSIS → DEGREE AND SIMILARITY ALGORITHMS. (Fig. 10).
11. Select the metric **Degree** and TVB's default connectivity matrix as input. Click on Launch.
12. Repeat the previous step for the following metrics: **Indegree and Outdegree, Strength, Instrength and Outstrength**.

13. Go back to PROJECTS → OPERATIONS DASHBOARD.
14. Click on . On the overlay window you can already see a summary of the basic descriptive statistics (Fig. 11).
15. You can display the results as a histogram by launching the Connectivity Measure Visualizer. However, here we'll launch the Topographic Viewer. This 2D layout of the head gives an idea of the network topology based on the node-wise metrics.

¹ Brain Connectivity Toolbox



Figure 10: Compute network metrics

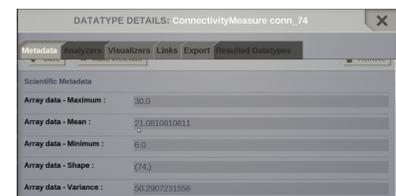


Figure 11: Descriptive summary

Since other network measures were previously computed so we can also have a look at them.

16. In this visualizer, it is possible to view and compare up to three metrics (Fig. 12). From the menu  on the top right select **strength** and **clustering coefficient**. Click on Update Visualizer. (See Fig. 13)

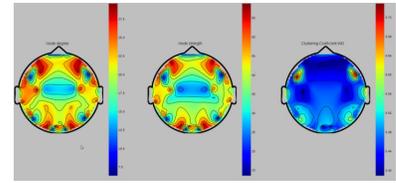


Figure 12: View and compare network

17. Go back to the CONNECTIVITY → LARGE SCALE CONNECTIVITY area.

18. In the field Node colors select the node-wise **Instrength, ID: 116005**.

19. Set a Color threshold of **56**. Launch the connectivity editor.

20. Go to the Transverse visualizer, which displays a spring-like layout of the network graph (top view). Click on Show all to apply the threshold previously set. Fig. 14



Figure 13: Select additional metrics

21. To completely remove the nodes, from Quick select we create a selection excluding these high in-strength nodes (Fig. 15) and directly save the new matrix.

22. From  you can change the measure used to colour the nodes as well as the threshold (Fig. 16). Repeat the necessary steps using **Degree**.

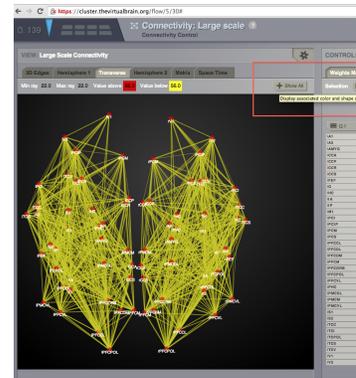


Figure 14: Show node colours.

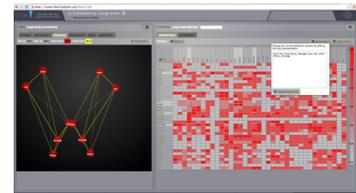
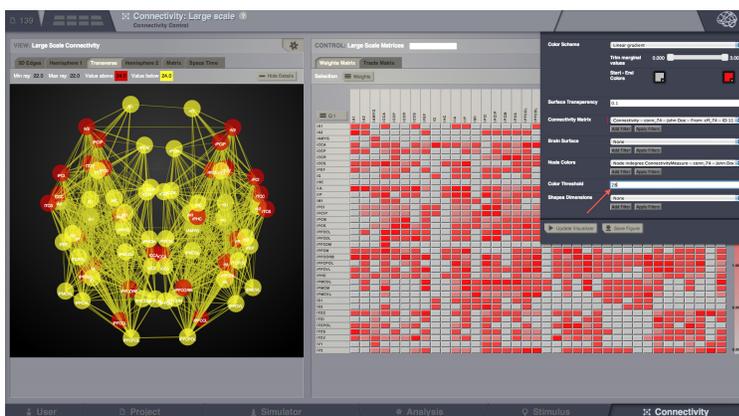


Figure 15: Select subnetwork with the highest in-strength nodes.
Figure 16: Change the metric and threshold.



The Effect Of The Structural Connectivity On The Global Stability Map

1. Go to SIMULATOR. Copy the simulation *Control_g2d_pse*. Launch a Parameter Space Exploration (PSE) using TVB's **default connectivity**.
2. Repeat the previous step but using the following matrices **NoInterhemispheric**, **NoHighInStrength** and **NoHighInDegree**. These are simulations \circ *Default_g2d_pse*, \circ *NoInterhemispheric_g2d_pse*, \circ *NoHighInStrength_g2d_pse* and \circ *NoHighInDegree_g2d_pse*. Their respective PSE maps are displayed in Figs. 17, 18, 19 and 20.

Region-based Simulations Of A Control And Stroke Connectome

WE WILL NOW to evaluate the difference between the activity generated by both control and stroke connectivity matrices. We'd like to see how the structural differences alter the stability plane of our system.

1. Go to the SIMULATOR area.
2. Select the **control connectivity, ID: 111886**. We proceed to launch a PSE using the same model and simulation parameters as found in Table 2 of the first session. This is simulation *Control_g2d_pse*.
3. Copy the previous simulation and launch a second PSE where, but this time use the **stroke connectivity, ID:111887**. The results are in simulation *Stroke_g2d_pse*.
4. Select a point in the resulting Parameter Space Discrete map, for instance the one with conduction speed equal to 7 mm/ms and global coupling strength $a=0.008 \text{ [au]}$. Tag the time-series (e.g Datatype Tag 3: **ThisPoint**).
5. Repeat the previous step for the same point in *Control_g2d_pse*.
6. Then, using the parameters from the previously tagged points, launch two **60 s** long simulations.
7. Use the **BOLD monitor** and the **Mixture of Gammas HRF kernel** with its default parameters. The sampling period of the monitor is **2000 ms**.

These two simulations are not included in this project, but you will need them if you want to reproduce the simulations described below.

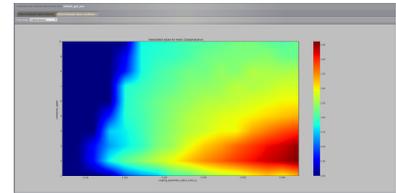


Figure 17: Global variance map from \circ *Default_g2d_pse*

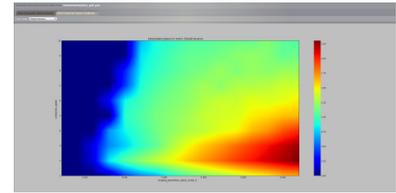


Figure 18: Global variance map from \circ *NoInterhemispheric_g2d_pse*

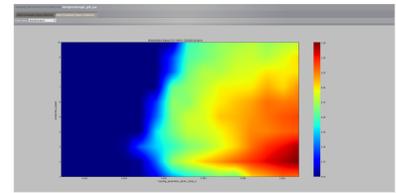


Figure 19: Global variance map from \circ *NoHighInStrength_g2d_pse*

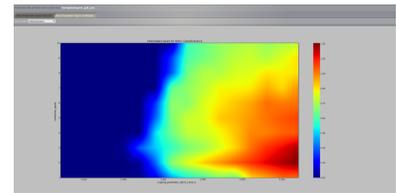


Figure 20: Global variance map from \circ *NoHighInDegree_g2d_pse*

- Using the branching mechanism, launch **4 min** simulations from the two previous ones.

These are *Control_g2d_init_bold_branch1* and *Stroke_g2d_init_bold_branch1*.



Figure 21: Add and apply a filter.

- Go to ANALYSIS →FOURIER SPECTRAL ANALYSIS.
- Add and apply a filter to find the time-series tagged with *ThisPoint*. (Fig. 21). Select **All**.
- Select a **Hamming window** with a length of **500 ms**. Launch the analyzer. (See Fig. 23).
- Go to PROJECTS→OPERATION DASHBOARD and have a look at the spectra of the time-series.

Region-based Simulations Of A Control And Stroke Connectome Using A Higher-dimensional Model

- Here, we will use the Stefanescu-Jirsa 3D (sj3d) presented in the previous talk.
- Select the **control connectivity, long range linear coupling** with a range between **0.002** and **0.02**, step **0.002**. Set the conduction speed in the range **1 mm/ms** and **10 mm/ms**, step **1 mm/ms**.
- The model parameters are specified in Table 2.
- In Variables watched by monitors select only **xi**.
- The integration scheme is **HeunDeterministic** with **dt=0.01 ms**.
- Launch two PSEs. The results are shown in Figs. 24 and 25 and correspond to simulations *Control_sj3d_pse* and *Stroke_sj3d_pse*.
- Select the point corresponding to a speed of **7 mm/ms** and linear coupling strength of **a=0.008** from each PSE map. Tag the corresponding time-series with Tag 2: sj3d and Tag 3: A.



Figure 22: Select points



Figure 23: Select points

Model parameter	Value
r	0.006
a	1
b	3
c	1
d	5
s	4
x_0	-1.6
K_{11}	0.5
K_{12}	0.15
K_{21}	0.15
σ	0.3
μ	3.3

Table 2: Stefanescu-Jirsa 3D parameters

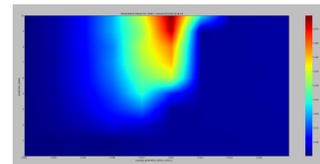


Figure 24: Variance map of *Control_sj3d_pse*.

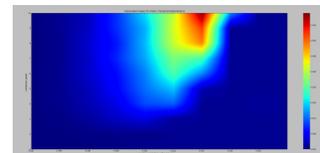


Figure 25: Variance map of *Stroke_sj3d_pse*.

- Then, using the parameters from the previously tagged points, launch two **60 s** long simulations. This time, in addition to the variable **xi**, select also the variable **alpha**.
- Use the **BOLD monitor** and the **Mixture of Gammas**

HRF kernel with its default parameters. The sampling period of the monitor is **2000 ms**.

These two simulations are not included in this project, but you will need them if you want to reproduce the simulations that are described below.

10. Using the branching mechanism, launch **4 min** simulations from the two previous ones. These are *Control_sj3d_init_bold_branch1* and *Stroke_sj3d_init_bold_branch1*.

11. Go to From PROJECTS → OPERATION DASHBOARD compute the Pearson correlation coefficients matrix of the resulting spatially averaged time-series.

The region pair-wise Pearson correlation coefficients matrix is commonly referred to as to functional connectivity. It is often computed from BOLD time series. Here, for simplicity and comparative purpose we compute the matrix using spatially averaged time-series of neural activity. The results are exhibited in Figs. 26 and 27.

More Documentation

For peer reviewed documentation on the software The Virtual Brain, please see the following articles [Sanz-Leon et al., 2013, Woodman et al., 2014]

Support

The official TVB website is www.thevirtualbrain.org. All the documentation and tutorials are hosted on [the-virtual-brain.github.io](https://github.com/the-virtual-brain). You'll find our public GIT repository at <https://github.com/the-virtual-brain>. For questions and bug reports we have a users group <https://groups.google.com/forum/#!forum/tvb-users>

References

P. Sanz-Leon, S. A. Knock, M. M. Woodman, L. Domide, J. Mersmann, A. R. McIntosh, and V. K. Jirsa. The virtual brain: a simulator of primate brain network dynamics. *Front Neuroinform*, 7:10, 2013. DOI: 10.3389/fninf.2013.00010. URL <http://dx.doi.org/10.3389/fninf.2013.00010>.

M. M. Woodman, L. Pezard, L. Domide, S. A. Knock, P. Sanz-Leon, A. R. McIntosh, J. Mersmann, and V. K. Jirsa. Integrating neuroinformatics tools in thevirtualbrain. *Frontiers in Neuroinformatics*, 8:36:–, 2014. DOI: 10.3389/fninf.2014.00036 - See more at: <http://journal.frontiersin.org/Journal/10.3389/fninf.2014.00036>.

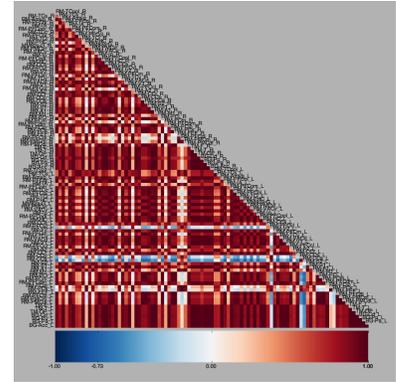


Figure 26: Pair-wise Pearson correlation coefficient matrix computed over the long BOLD time-series. Control matrix.

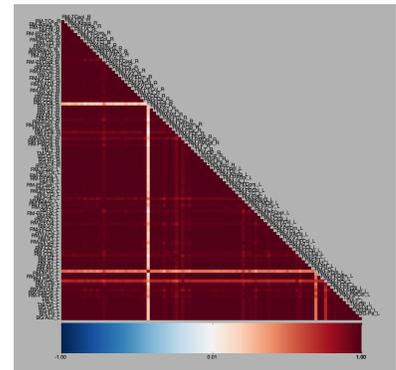


Figure 27: Pair-wise Pearson correlation coefficient matrix computed over the long BOLD time-series. Stroke matrix.

NODE #1
HAMBURG



THE VIRTUAL BRAIN.

Hands-on Session #4

Stimuli Generation and Heterogeneous Models

June 7, 2014

Virtual stimulation will allow researchers to directly apply an external perturbation to cortical and subcortical regions. The induced activation propagates through long-range connections to other brain areas. Here, we focus on the generation of simulation trains to be able to study for instance changes in functional connectivity.

Objectives

THIS TUTORIAL DESCRIBES the process of generating a virtual stimulus at the region and surface. This spatiotemporal pattern will be applied to the brain network model. Furthermore, we will explore what happens when the local dynamics are not homogeneous.

Project: *Session_IV_HeterogenousModelsAndStimulation*

IN THIS SESSION, we'll show you how to build stimulation patterns for your region or surface simulations. Also, we'll go beyond homogeneous models, that is when all the nodes have the same initial local dynamics, and define different dynamics for specific regions or patches of our brain network model. All the data were already generated. We'll only go through the necessary steps required to reproduce some simulations listed in Table 1. You can always start over, click along and/or try to change parameters.

Region-based Stimuli Generation

WITHIN TVB, STIMULI can be specified at either the region or surface level, with the latter only applying to the case of simulations including a mesh representation of the cortical surface. Here, we will first define a basic stimulus at the region level and apply it to a region-based simulation. We'll use a deterministic integration scheme so that the effects of the stimuli are very clear and after that we'll do the same using a stochastic integration.

This example has been published in [Sanz-Leon et al., 2013], which is merely a toy example for understanding evoked responses.

The basic process consists in defining the temporal profile of the stimulus, selecting some nodes and defining the weighting of the stimuli coming into those nodes.

Name
Region-based Stimuli Generation
Surface-based Stimuli Generation
Stimulus-driven simulations
<i>EvokedResponses_init</i>
<i>EvokedResponses_init_branch1</i>
<i>EvokedResponses_init_stochastic</i>
<i>EvokedResponses_init_stochastic</i>
<i>EvokedResponsesSurface_init</i>
<i>EvokedResponsesSurface_init_branch1</i>
<i>EvokedResponsesSurface_init_branch2</i>
Heterogeneous models
<i>HomogeneousRegionModel</i>
<i>HeterogeneousRegionModel</i>
<i>HomogeneousSurfaceModel</i>
<i>HeterogeneousSurfaceModel</i>

Table 1: Simulations in this project.

1. Go to **STIMULUS** → **REGION STIMULUS** area.
2. Selecting an equation, give a name to the stimulation pattern and set its parameters as desired.
3. Here we'll just take the default Pulse Train. We'll shift the **onset** to **500 ms**, **tau=5 ms** is the duration of the pulse, and **T=500 ms** is the repetition period (equivalent to 2 Hz). (See Fig. 1).

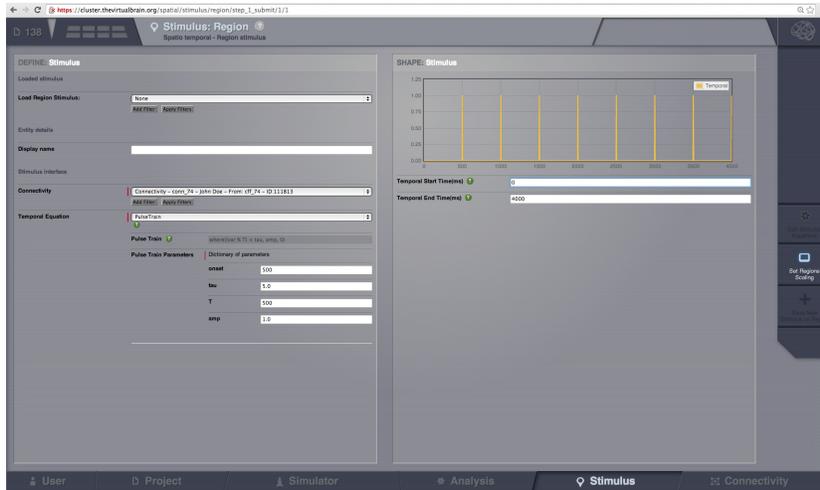


Figure 1: Region-based stimulus generation

4. On the right column you can change the end time to visualize the stimulus. See Fig. 2.

To define which and how much nodes will be affected by a stimulus, a scaling factor per node can be defined.

5. Click on Set Region Scalings.
6. Unselect all the nodes. Then, select the primary visual cortices, left and right V1, save the selection for later use. (Figs. 3 and 4)

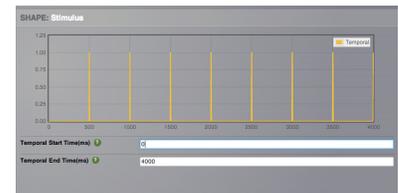


Figure 2: Square pulse train

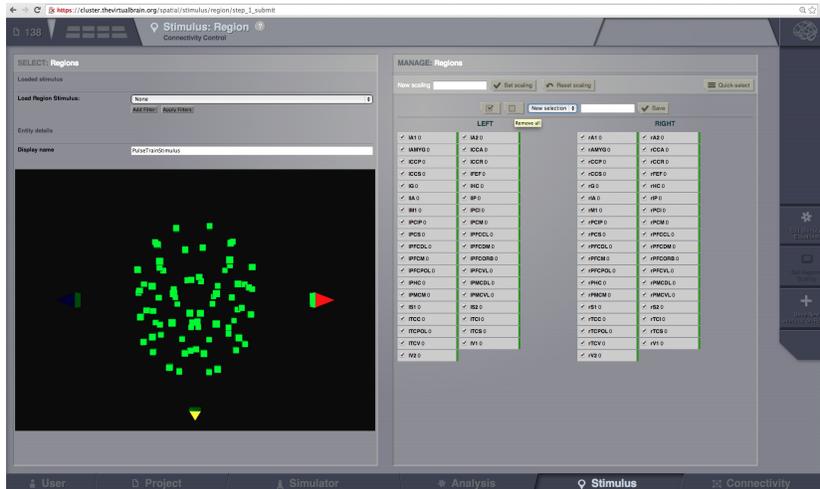


Figure 3: Unselect all the nodes

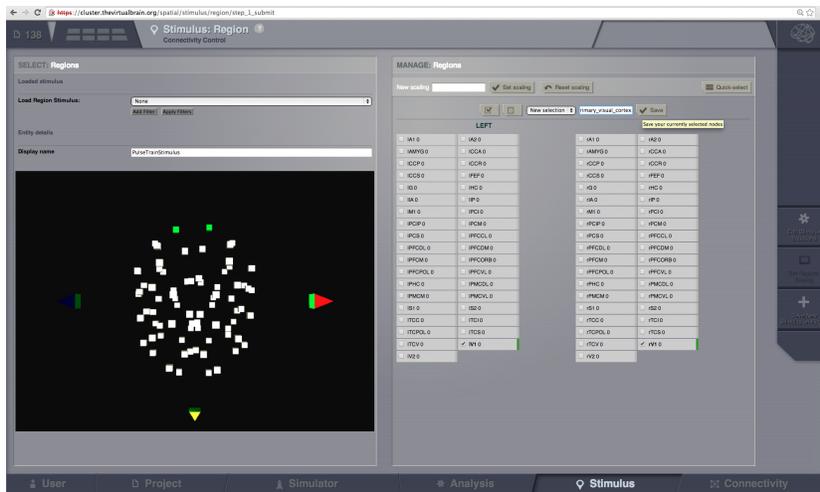


Figure 4: Region-based stimulus generation

7. Set the new scaling to 3.5 – only those nodes will receive stimulation.
8. Select all the nodes again and save the stimulus. (Figs. 5 and 6).

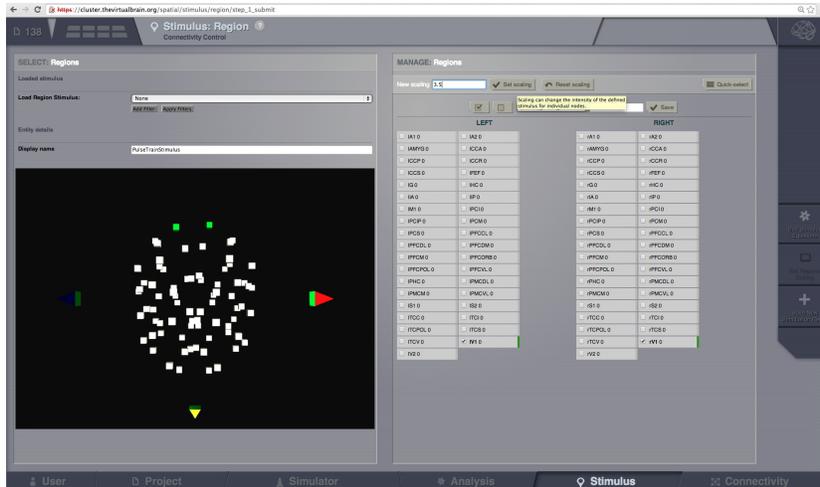


Figure 5: Unselect all the nodes

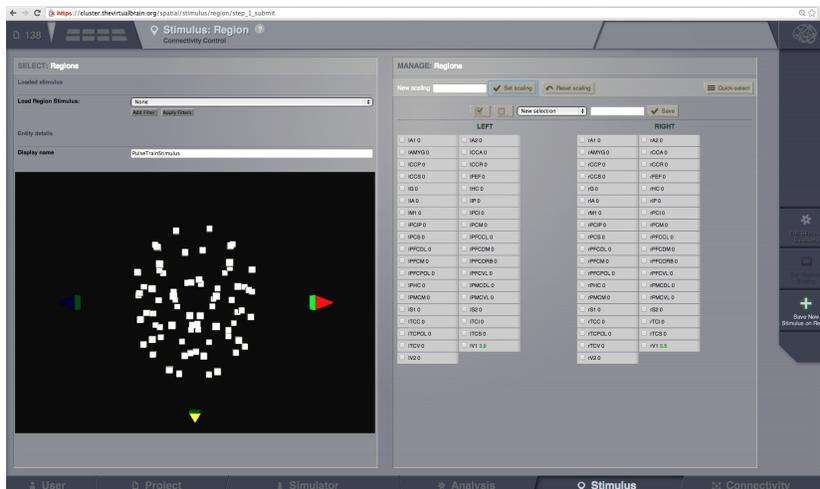


Figure 6: Region-based stimulus generation

Surface-based Stimuli Generation

1. Go to STIMULUS → SURFACE STIMULUS area.
2. As with the region level stimuli, we use an equation to define the temporal profile. The temporal component of the surface stimulus will be the same square pulse train as before, except that we'll set the onset and the repetition period to **50 ms**.
3. However, unlike in the region level case, we also use an equation to define the spatial profile of the stimulus. For the spatial profile we'll use a Sigmoid function with the only change being the width of the kernel, sigma, that we set to **10 mm**. Here, as with the local connectivity, we must use an equation

which drops toward zero with increasing distance.(Fig. 7)

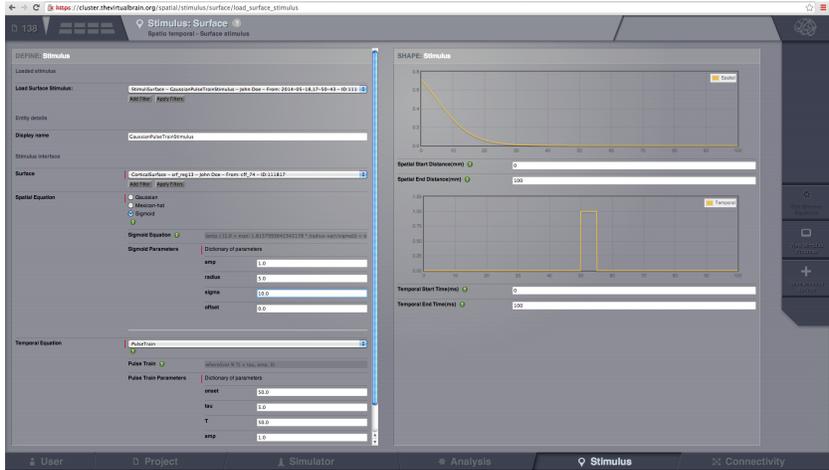


Figure 7: Surface-based stimulus generation

- We also need to specify one or more focal points (vertices on the cortical surface), about which the spatial equation will be evaluated. To do this we go to Edit Focal Points and View where we can select a few focal points by clicking on the cortical surface and then on the Add focal point button. (Fig. 8)

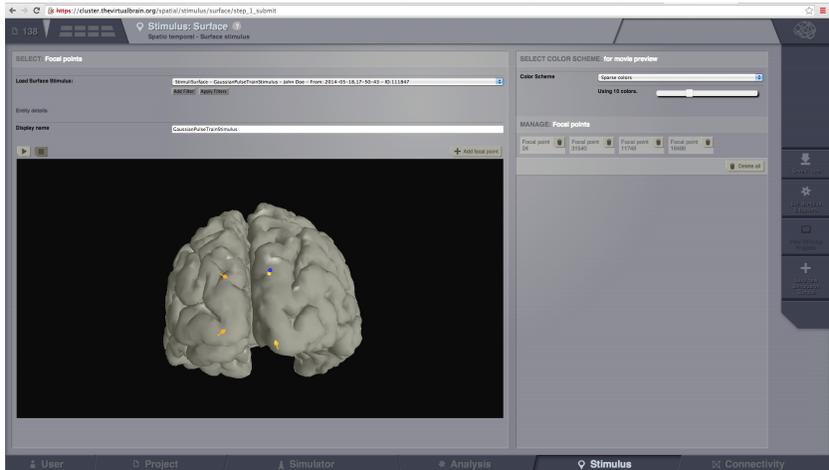


Figure 8: Add focal points

- We save the stimulus and then we can visualize it as a movie. This was already done for you, so you can directly click on . (Fig. 9).

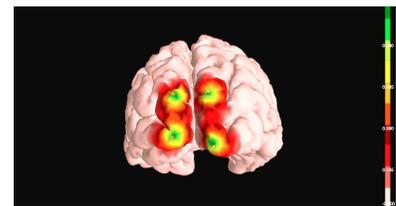


Figure 9: Have a look at the spatiotemporal profile of the stimulus.

Stimulus-driven simulations

Next, we go back to the SIMULATOR area, and we'll set up the local dynamics using the generic 2D oscillator.

1. Now we'll build the model using the Connectivity, Model, Integrator and Monitors as in [Sanz-Leon et al., 2013]. You can check the local dynamics using the phase plane interactive tool. The model parameters are specified in Table 2.
2. Remember the branching mechanism? Before applying the stimulus to our model we'll let the system reach a steady state first, to avoid the initial transient to mask the effect of the stimulation. This is what you have in *EvokedResponses_init*
3. Run a simulation for **1000 ms**, with HeunDeterministic and **dt = 0.1 ms**; use the Temporal Average monitor with a sample period equal to **1 ms**
4. Next, we add a stimulation pattern (Fig. 10) and continue the simulation for another **4000 ms**.

Results are in *EvokedResponses_init* and *EvokedResponses_init_branch1*.

5. Finally, repeat steps 1-4 using a stochastic integration: HeunStochastic; additive noise with **D = 0.005 au** and **dt = 0.1 ms**. These are the simulations in *EvokedResponses_init_stochastic* and *EvokedResponses_init_stochastic_branch1*.

Model parameter	Value
a	-0.5
b	-15
c	0
d	0.02
e	3
f	1
g	0
I	0
α	1
β	1
γ	1
τ	1

Table 2: In this configuration the topology of the phase portrait features a stable fixed point (a stable spiral) with a characteristic frequency of approximately 10 Hz.

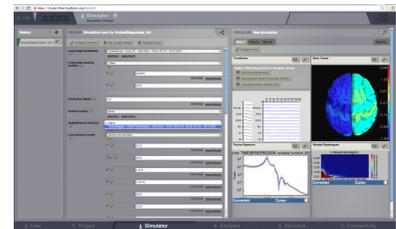


Figure 10: Add a stimulation pattern

Now, we'll have a look at the resulting time-series.

1. From the OPERATION DASHBOARD, select the time-series  from *EvokedResponses_init_branch1*. Then, from the Visualizer tab launch the Animated time series visualizer. (Fig. 11).
2. The node selection we saved when we created the stimulus is available. You can directly select the nodes where stimulation was applied. (Fig. 12). You'll have to change the page size, that is how many time points are displayed. (Fig. 13).
3. The stimulus was applied to V1, however if we select left V2 and zoom in we can see the effect of the stimulus. (Fig. 14)

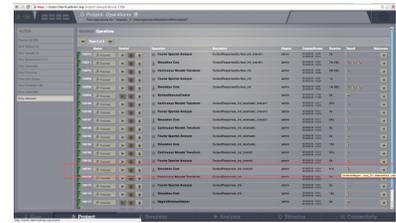


Figure 11: Select a time-series

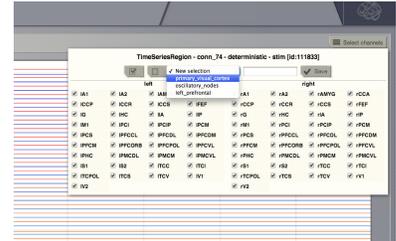


Figure 12: Use the node selection *primary_visual_cortex*

1. For surface simulations, select the default cortical surface and the local connectivity.
2. In addition we set up a few Monitors: EEG, Spatial Average and Temporal Average.
3. As with the region simulations we'll first run a short simulation, **200 ms** to clear out the initial transient.
4. Add the Surface stimulus and continue the simulation for **4000 ms**. This is a full brain network model making use of all the components of TVB. Results are in *EvokedResponsesSurface_init* and *EvokedResponsesSurface_init_branch1*.



Figure 13: Change the page size



Figure 14: Left V2: zoom in

Note that surface simulations can also be run with stimuli defined at the region level. In this case, all the vertices that belong to one region will receive the same stimulus. This is what we did in *EvokedResponsesSurface_init_branch2*. The differences are shown in Figs. 15 and 16

Heterogeneous models

WE HAVE ASSUMED THAT every node either in a region or surface simulation has the same dynamics. TVB allows for the "spatialization" of parameters, that is, every node can have different dynamical regimes by varying the models parameters. This feature becomes relevant when modelling regions or patches of the brain that have different dynamics. A specific case will be presented in the next session about modelling epilepsy in TVB.

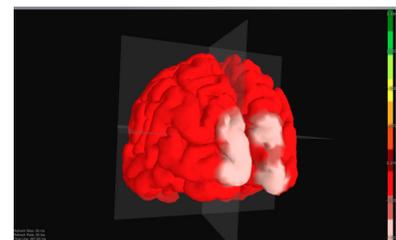


Figure 15: Surface simulation with a stimulus defined at the surface level.



Figure 16: Surface simulation with a stimulus defined at the region level.

For region simulations the mechanism is similar as the one for creating Region Stimulus and for surface simulations a spatial profile determines the variation of the parameter according to a smooth function.

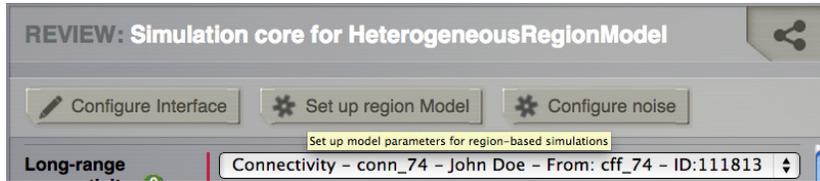


Figure 17: Click on Set Up Region Model.

1. For region simulations, set up a simulation using the generic 2d oscillator with the parameters from Table 2 found in the BUILDING YOUR OWN BRAIN NETWORK MODEL hand-out.
2. Run a simulation for **1000 ms**, with HeunDeterministic and **dt = 0.1 ms**; use the Temporal Average monitor with a sample period equal to **1 ms**. This is simulation *Homogeneous-RegionModel*.
3. Next, configure a second simulation with the same parameters as before (e.g., by coping the previous simulation).
4. Click on Set Up Region Model (see Fig. 17). We'll change the value of the model parameter **b** for a selection of nodes. This selection is already available under the name *oscillatory_nodes*. (Fig. 18)

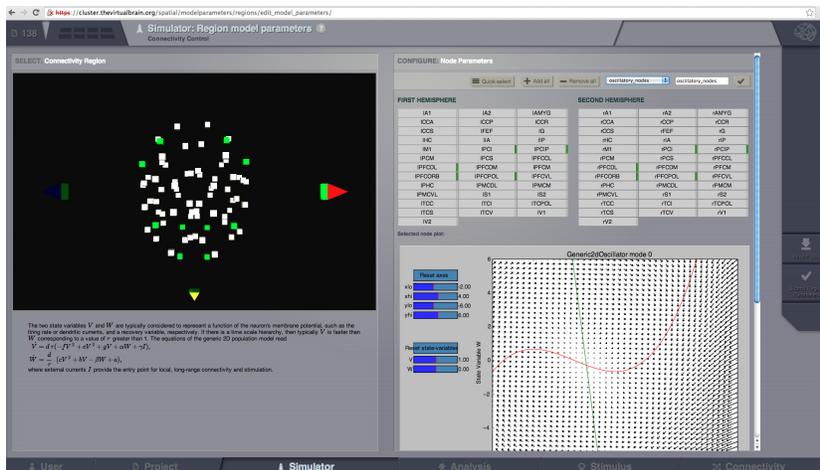


Figure 18: Select the nodes that will have a different value.

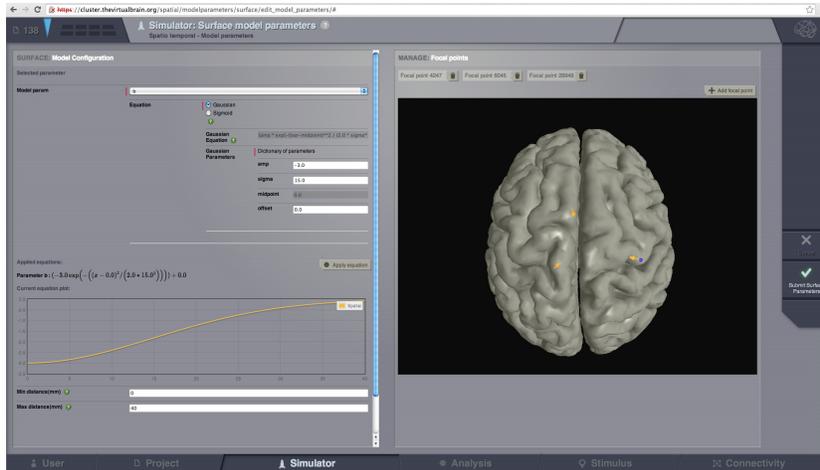


Figure 21: Describe a parameter spatial variation with a function.

9. You can now launch the simulation and observe the difference with respect to the homogeneous version. (Fig. 22).

More Documentation

FOR MORE DOCUMENTATION on The Virtual Brain, please see the following articles [Sanz-Leon et al., 2013, Spiegler and Jirsa, 2013, Woodman et al., 2014, Jirsa et al., 2010]

Support

The official TVB website is www.thevirtualbrain.org. All the documentation and tutorials are hosted on [the-virtual-brain.github.io](https://github.com/the-virtual-brain). You'll find our public GIT repository at <https://github.com/the-virtual-brain>. For questions and bug reports we have a users group <https://groups.google.com/forum/#!forum/tvb-users>

References

V. K. Jirsa, O. Sporns, M. Breakspear, G. Deco, and A. R. McIntosh. Towards the virtual brain: network modeling of the intact and the damaged brain. *Arch. Ital. Biol.*, 148(3):189–205, Sep 2010.

P. Sanz-Leon, S. A. Knock, M. M. Woodman, L. Domide, J. Mersmann, A. R. McIntosh, and V. K. Jirsa. The virtual brain: a simulator of primate brain network dynamics. *Front Neuroin-*

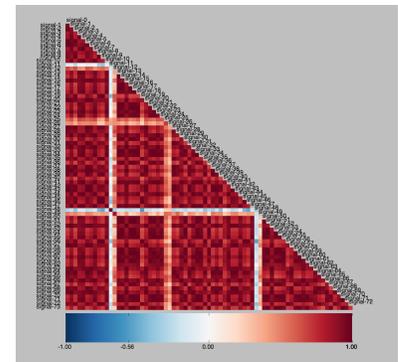


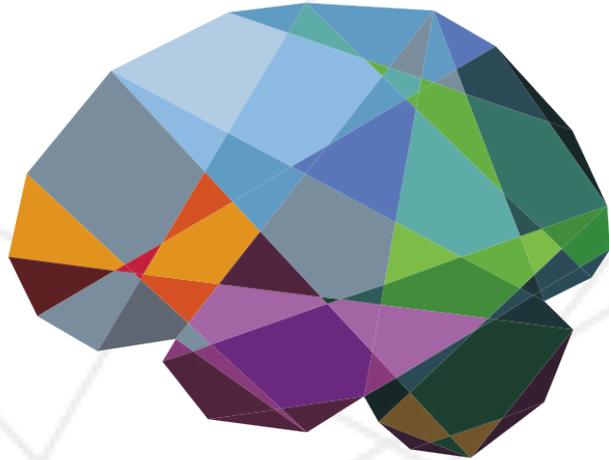
Figure 22: Pearson correlation coefficients computed from *HeterogeneousSurfaceModel*.

form, 7:10, 2013. DOI: 10.3389/fninf.2013.00010. URL <http://dx.doi.org/10.3389/fninf.2013.00010>.

A. Spiegler and V. Jirsa. Systematic approximations of neural fields through networks of neural masses in the virtual brain. *Neuroimage*, 83C:704–725, Jun 2013. DOI: 10.1016/j.neuroimage.2013.06.018. URL <http://dx.doi.org/10.1016/j.neuroimage.2013.06.018>.

M. M. Woodman, L. Pezard, L. Domide, S. A. Knock, P. Sanz-Leon, A. R. McIntosh, J. Mersmann, and V. K. Jirsa. Integrating neuroinformatics tools in the virtual brain. *Frontiers in Neuroinformatics*, 8:36:–, 2014. DOI: 10.3389/fninf.2014.00036 - See more at: <http://journal.frontiersin.org/Journal/10.3389/fninf.2014.00036>.

NODE #1
HAMBURG



THE VIRTUAL BRAIN.

Hands-on Session #5
Modelling An Epileptic Patient
June 7, 2014

It is possible to use TVB to model a specific subject such as an epileptic patient. Using relevant neural mass models, TVB allows to ask multiple questions such as the localisation of the epileptogenic zone or the validity of different neuroimaging modalities to assess the epileptogenicity of a brain structure. Here we will present an example of such a modelisation.

Objectives

THE MAIN GOAL of this session is to provide a clear understanding of how we can reproduce clinically relevant scenarios such as stimulation, modelisation of forward solution of sEEG and EEG during a seizure; modelisation of forward solution of MEG with interictal spikes; and, modelisation of surgery with resection of a part of the brain.

Another important step is to process data from neuroimaging modalities into TVB. We are currently developing a pipeline (github.com/timpix/scripts) that will let you easily prepare your data in a format suitable for import into TVB. All you will need for this pipeline will be a T1 MRI and a Diffusion MRI from the subject. You can find these data in database such as the Human Connectome project. At the end of the pipeline, you will have a connectivity, a surface and a region mapping, i.e. you will be able to do region based and surface based simulations. We will use the different functionalities that you have learned in the previous sessions.

Project: *Session_V_ModellingEpilepticPatient*

IN THIS PROJECT, all the data were already generated. We'll only go through the necessary steps required to reproduce the simulations listed in Table 1, along with the relevant outline. You can always start over, click along and/or try to change parameters. We will use the default subject connectivity matrix and surface.

Exploring the Epileptor model

BEFORE DOING any simulations, we would like to have a look at the phase space of the Epileptor model to understand better its dynamics. We will use the interactive tool that you have seen in Session #4.

Name
Exploring the Epileptor model
Region based simulation of an epileptic patient <i>Region_TemporalLobe</i> <i>Region_TemporalLobe_sEEG_EEG</i>
Surface based simulation of an epileptic patient <i>Surface_TemporalLobe_sEEG_EEG</i>
Applying a stimulus to trigger seizures <i>Surface_Stimulation</i>
MEG forward solution for interictal spikes <i>Surface_InterictalSpikes_MEG</i>
Modeling surgical resection <i>Surface_Resection</i>

Table 1: Simulations in this project.

1. Go to SIMULATOR and select the **Epileptor** model
2. Click on Set Up Region Model.
3. Look at the phase space (Fig. 1). We have here the first population (variables y_0 in abscissa and y_1 in ordinate). The left most intersection of the nullcline defines a stable fixed point whereas the rightmost intersection is the center of a limit cycle. Both states are separated by a separatrix, as you can see by drawing different trajectories in this phase space (left click on the figure).
4. You can also look at other variables in the phase space, such as y_2/y_0 (slow-fast subsystem) or y_5/y_4 (second population), and change parameters to see what is the effect on the nullclines.

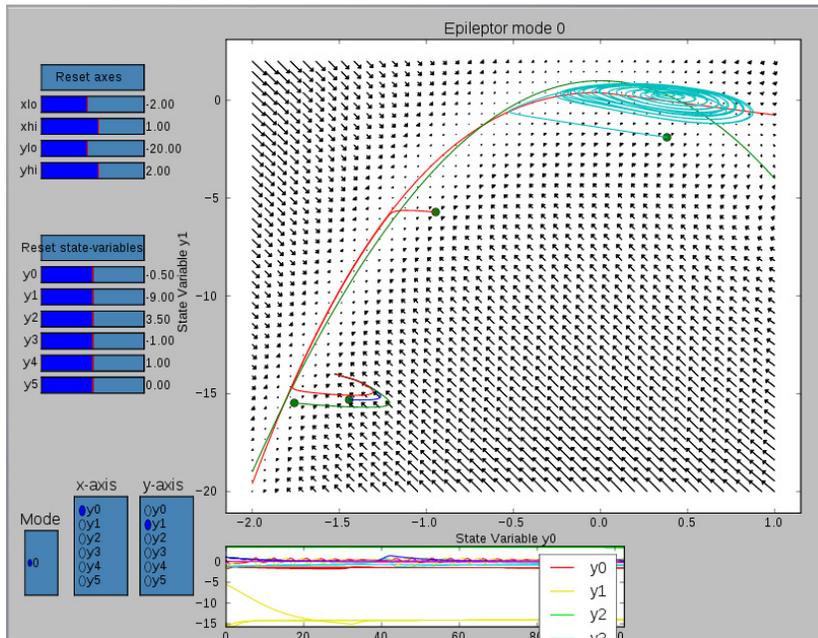


Figure 1: phase space for the first population

Region based simulation of an epileptic patient

WE ARE GOING to model a patient with temporal lobe epilepsy (TLE). We will choose different values of epileptogenicity (x_0 parameters in the Epileptor) according to the region positions, thereby introducing heterogeneities in the model parameters (Session #4). We set the right limbic areas (right hippocampus (rHC), parahippocampus (rPHC) and amygdala (rAMYG)) as epileptic zones. We also add

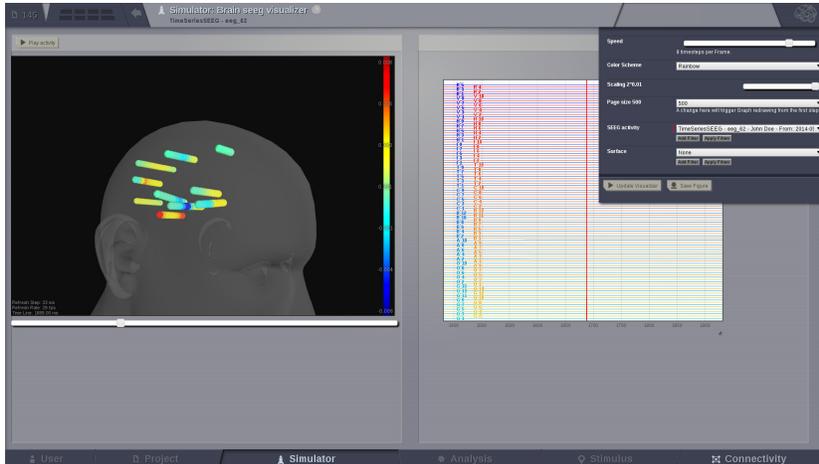


Figure 6: EEG 3d/2d visualizer

have a more accurate representation of the MEG/EEG/sEEG signal.

1. Copy the former simulation.
2. Add a Brain Viewer visualizer.
3. Choose the TVB's **default surface**, the corresponding **local connectivity** and a **Local coupling strength** of $\alpha = 0.2$.
4. Add a **Spatial average** monitor with a **sampling period** of **1 ms**
5. Click on Set up surface model (Fig. 7). Choose working parameters (**Model parameter** x_0 , **Equation** : **Gaussian**, **amp** = 0.6, **sigma** = 10., **offset** = -2.2) and click on Apply equation. Click on a location on the brain where you want the parameter setting to apply, then click on Add focal point. Then click on Submit Surface parameters. You can see that the field of the x_0 parameter is updated according to the selected equation.

The results are already computed for you in *Surface_TemporalLobe_sEEG_EEG*

6. Click on Results, click on the TimeSeriesSurface and visualize them in the Brain Activity Visualizer (Fig. 8). You can use the arrows of the numpad to rapidly move the brain.
7. Go back, click on TimeSeries and visualize them with the Time Series Visualizer. Change the scaling, change the selected channels and zoom in to see the seizure (Fig. 9).
8. Go back, click on TimeSeriesSEEG and visualize them with the 3d/2d visualizer (Fig. 10). Note the difference with the former region-based simulation.

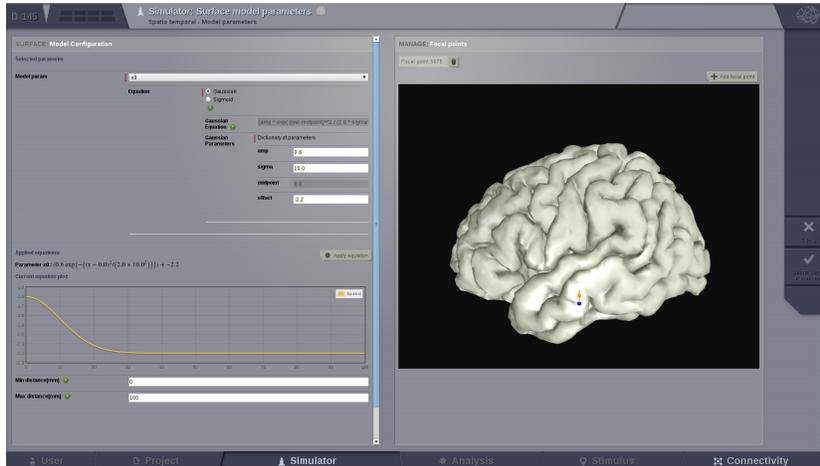


Figure 7: Set up the surface parameters

9. Go back, click on the EEG time series and visualize them with the 3d/2d visualizer (Fig. 11). Note the difference with the region-based simulation.

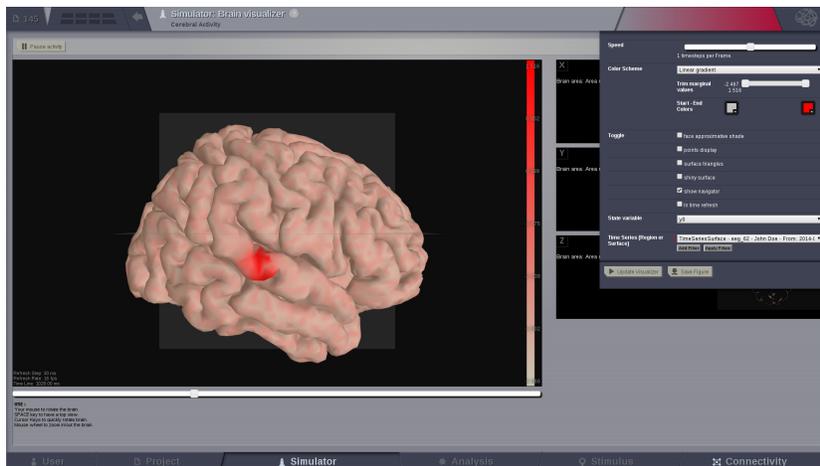


Figure 8: Spatial averaged time series for a surface simulation

Applying A Stimulus To Trigger Seizures

Now we are going to simulate a stimulation. We set the whole brain to non-epileptogenic but close to the threshold

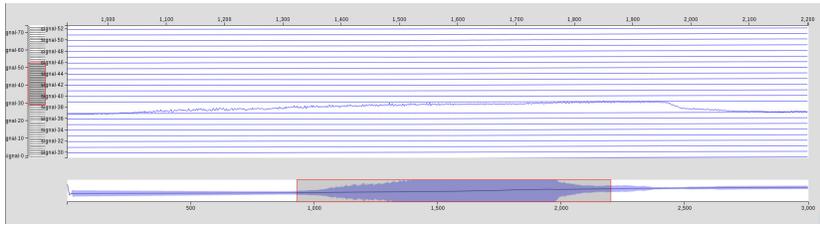


Figure 9: Spatial averaged time series for a surface simulation

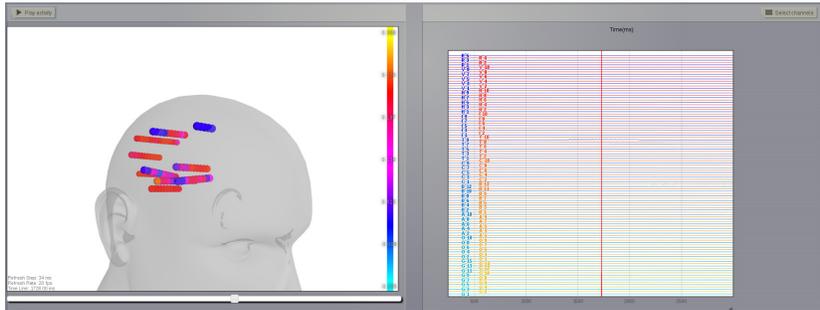


Figure 10: sEEG for a surface simulation

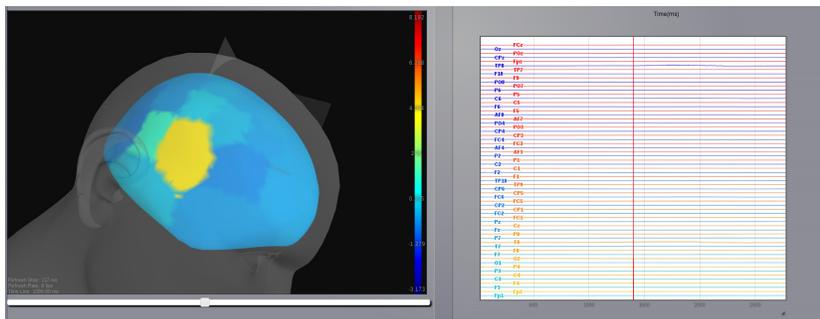


Figure 11: EEG for a surface simulation

1. Go to **STIMULUS** → **SURFACE STIMULUS**
2. Give a name to the new stimulus
3. Choose a Sigmoid stimulation in space with the parameters given in Table 3
4. Choose a PulseTrain stimulation in time with parameters given in table 3 (Fig. 12)
5. Click on Edit Focal Points and View
6. Choose a focal point (Fig. 13)
7. Save the new stimulus on surface

Space stimulation parameters	Value
<i>amp</i>	1.0
<i>radius</i>	5.0
<i>sigma</i>	1.0
<i>offset</i>	0.0
Time stimulation parameters	Value
<i>onset</i>	2000.0
<i>tau</i>	20.0
<i>T</i>	4000.0
<i>amp</i>	10.0

Table 3: Space and time parameters for the stimulus

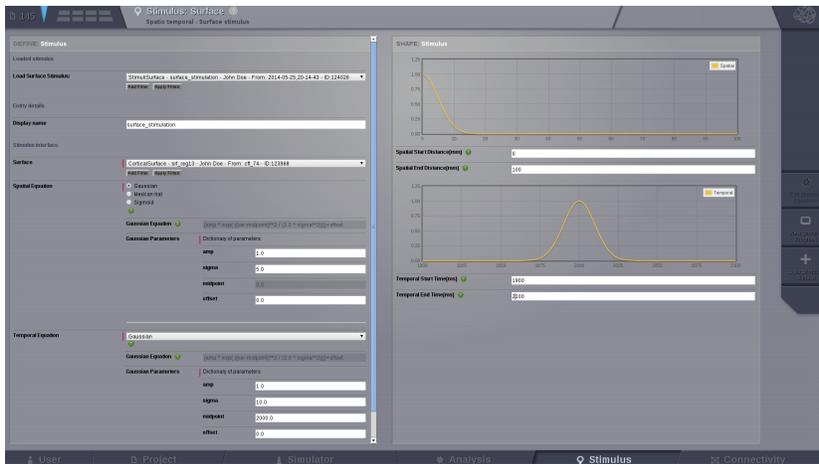


Figure 12: Spatio temporal pattern of the stimulus

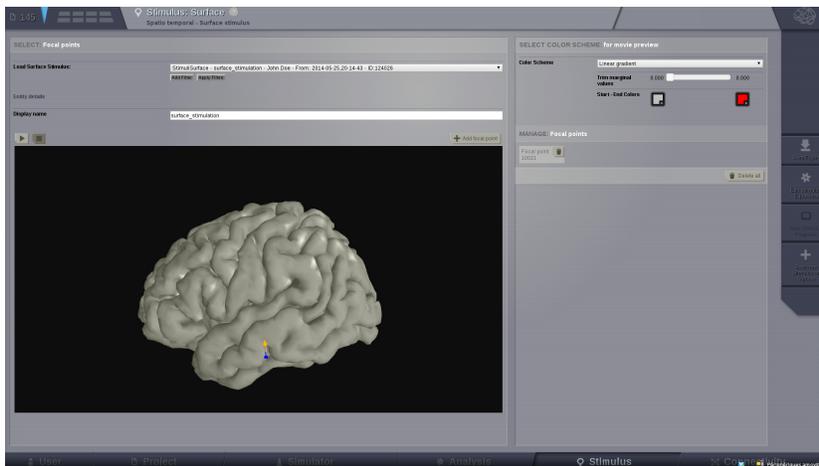


Figure 13: Focal point for a surface stimulation

The stimulus was already set for you under the name *Surface_SquareStimulus*

8. Go to **SIMULATOR** and copy the former simulation.
9. Choose the *Surface_SquareStimulus* stimulus.
10. Set the parameter x_0 to -2.1
11. Choose only a **Spatial average** monitor.
12. Set the **Simulation Length** to **4000 ms**.

You can see the result of this simulation in *surface stimulation*

13. Click on **Results**, click on the TimeSeries and visualize the Surface Average Time Series with the **Animated Time Series** visualizer.
14. **Select channel #67 to #73** (Fig. 14). Increase the page size in  to include time **2000 ms** (Fig. 16) and better see the effect of the stimulation (Fig. 15).

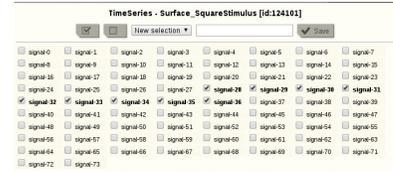
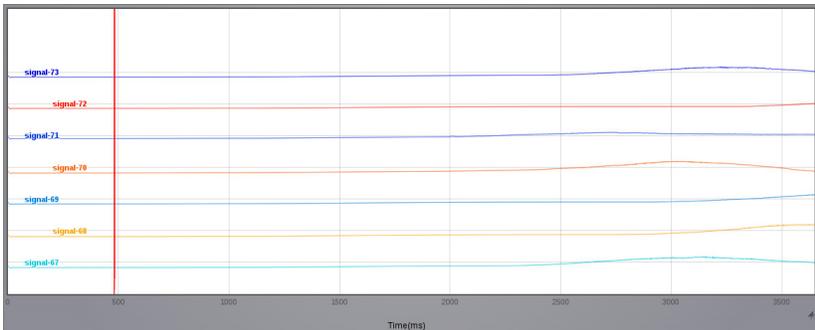


Figure 14: Channel selection menu: you can choose the channels of interest.

Figure 15: Time Series for a stimulation

MEG forward solution for interictal spikes

MEG IS RARELY recorded during a seizure and usually we have only access to interictal recordings. Therefore it is interesting to simulate interictal spikes in order to test forward and inverse model in MEG. We are going to set the parameter x_0 in such a way that the Epileptor model will not trigger seizures, but only interictal spikes, i.e. the model will be set close to the threshold between ictal and interictal states. We will also add noise to trigger the interictal spikes.

1. Copy the former simulation.
2. Set the **Spatiotemporal Stimulus** back to the **None** value.



Figure 16: Brain menu: Increase the scaling to see the time 2000[ms].

3. As we did earlier, go to Set Up region Model and set the x_0 parameter to -3.0 for all nodes except for the selection *right_amy_hip parahip* for which you set the value -2.1 and for the selection *right_orbital_temporal* for which you set the value -2.8 .
4. Choose a **HeunStochastic** integration scheme with a **integration step-size** of **0.05**. We choose a smaller integration time step than previously to be sure that the integration scheme stays stable even in presence of noise. We will put noise in the two variables of the second population which are responsible for triggering interictal spikes: set **D** to **[0., 0., 0., 0.002, 0.002, 0.]**.
5. Choose the **MEG** monitor with a sampling period of **1 ms**

The results are given in *Surface_InterictalSpikes_MEG*.

1. Go to Results, select the MEG Time series and visualize them with the Time Serie Visualizer. Choose channels #15, #73, #84 and #150 and increase the scaling, you should see interictal spikes on this MEG simulation (Fig. 17).

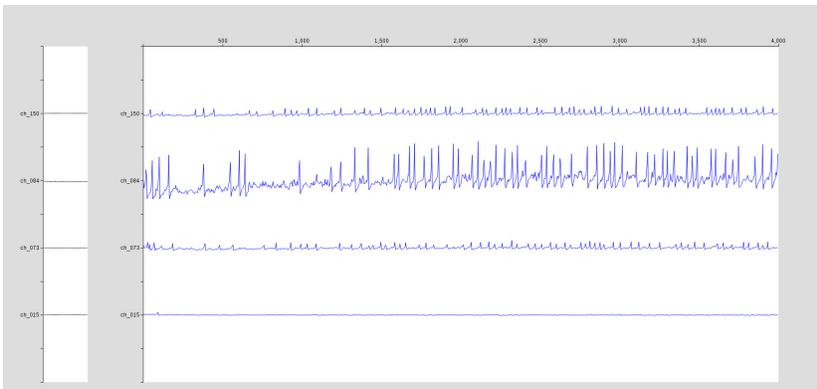


Figure 17: MEG Time Series showing interictal spikes

You have seen that the time series of different channels generated by the MEG spherical forward solution have very different amplitudes. This is caused by our choice of the spherical forward solution which is not a really good model. A better approach is to use the MEG leadfield matrix, such as the one we used for EEG. Such a matrix will soon be available in TVB.

Modeling surgical resection

SURGICAL RESECTION is used for around 20% of epileptic patient whose seizures are drug-resistant. We can imagine in this case that a small part of the brain of the patients has been resected (for instance the hippocampus, the amygdala and the parahippocampal cortex) and we are going to simulate that.

1. Go to CONNECTIVITY → LARGE SCALE CONNECTIVITY.
2. As you learned in session #3, select the nodes IAMYG, IHC and IPHC via your former selection, delete all their in-out and out-in connections with the other nodes, give a name to the selection and save it. (Fig. 18)
3. As we choose the same regional parameters that in the first simulation, we are going to copy them directly from our former setting. Go to the *Region_TemporalLobe* simulation, select and copy the values for the parameter x_0 .
4. Copy the surface simulation *Surface_TemporalLobe_sEEG_EEG* and paste the array of values in the parameter x_0 . With this direct copy-paste method, we avoid to redo the work previously done via the Set Up Region Model panel. You can also use you own vector, as long as the length of this vector correspond to the number of regions. Change the values -1.6 by -2.2 in this array (i.e. we replace the dynamics of the resected node by a stable node).
5. Choose only a **Spatial average** monitor with a **sampling period** of **1 ms**.
6. Remove the Brain Viewer visualizer.
7. Do not forget to choose the right connectivity matrix and you are ready to launch the simulation.

The results are given in *Surface_Resection*.

6. Click on Results, then TimeSeries and visualize the spatial average time series with the a Time Series visualizer. Don't forget to increase the Scaling and Select all channels. (Fig. 19)

More Documentation

For more documentation on The Virtual Brain platform, please see the following articles [Sanz-Leon et al., 2013, Woodman et al., 2014].

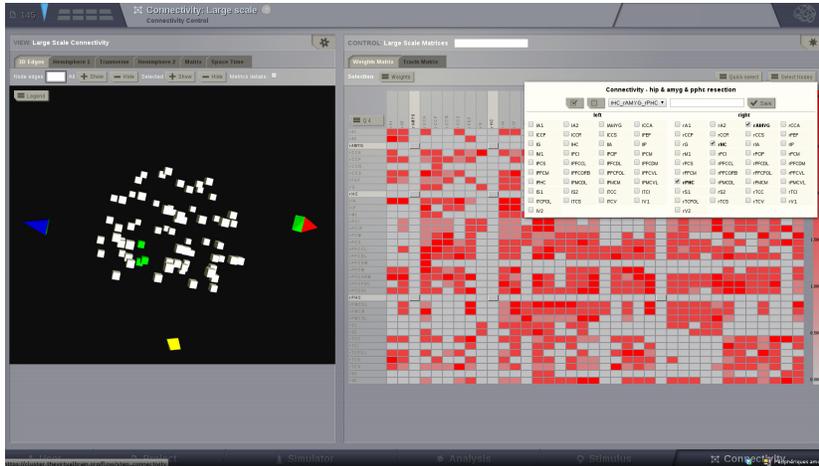


Figure 18: Focal point for a surface stimulation

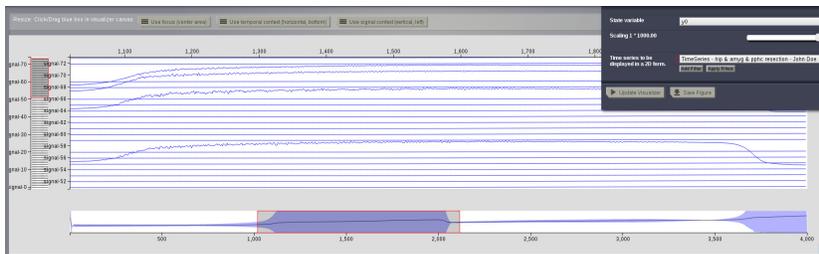


Figure 19: Time Series after a resection

For more details about the **Epileptor** model, see [Jirsa et al., 2014].

Support

The official TVB website is www.thevirtualbrain.org. All the documentation and tutorials are hosted on the [-virtual-brain.github.io](https://github.com/the-virtual-brain). You'll find our public GIT repository at <https://github.com/the-virtual-brain>. For questions and bug reports we have a users group <https://groups.google.com/forum/#!forum/tvb-users>

References

Viktor K. Jirsa, William C Stacey, Pascale P Quilichini, Anton I Ivanov, and Christophe Bernard. On the nature of seizure dynamics. *Brain*, (In Press):-, 2014.

P. Sanz-Leon, S. A. Knock, M. M. Woodman, L. Domide, J. Mersmann, A. R. McIntosh, and V. K. Jirsa. The virtual brain: a simulator of primate brain network dynamics. *Front Neuroinform*, 7:10, 2013. DOI: 10.3389/fninf.2013.00010. URL <http://dx.doi.org/10.3389/fninf.2013.00010>.

M. M. Woodman, L. Pezard, L. Domide, S. A. Knock, P. Sanz-Leon, A. R. McIntosh, J. Mersmann, and V. K. Jirsa. Integrating neuroinformatics tools in the virtual brain. *Frontiers in Neuroinformatics*, 8:36:–, 2014. DOI: 10.3389/fninf.2014.00036 - See more at: <http://journal.frontiersin.org/Journal/10.3389/fninf.2014.00036>.