The Virtual Brain: building blocks for whole-brain modelling

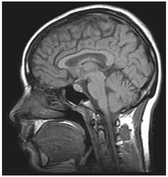
Jan Fousek

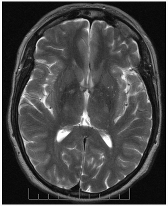
September 11, 2019

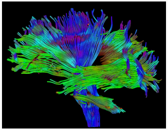
INS/TNG, Aix-Marseille University

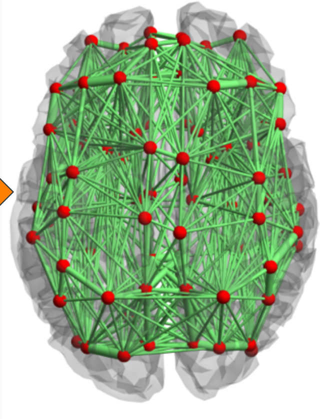
Motivation behind TVB

To use individual structural differences to constraint personalized brain network models.



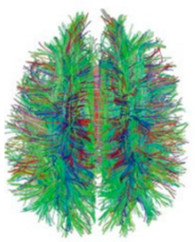


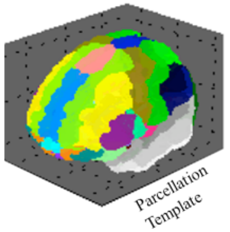




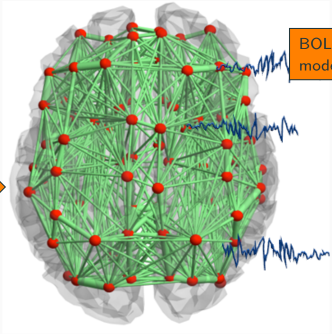
The Virtual Brain Workflow

Neuroimaging data





Brain Network Model



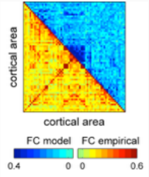
BOLD
model

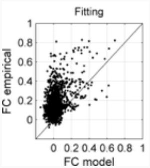
Simulated Neural Activity

local dynamical model

structural connection

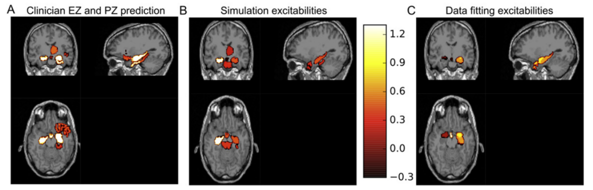
Analysis: e.g. simulated vs empirical FC





Example applications

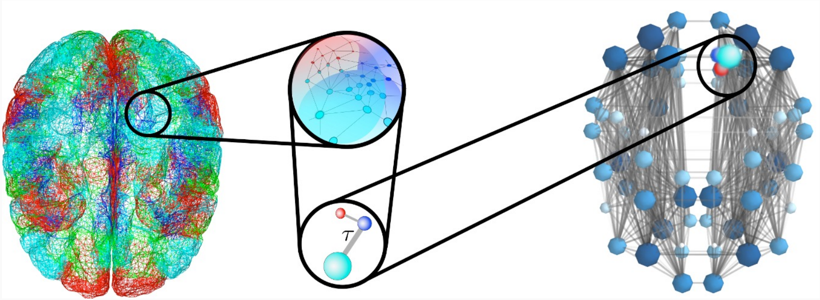
- Lesions: recovery after stroke (doi: 10/czsh)
- Depression: modelling the effect of DBS (Protzner et al., submitted)
- The Virtual Mouse (doi: 10/czsg)
- Epilepsy: estimation of epileptogenic zone by parameter inference



- ...

Brain Network Model

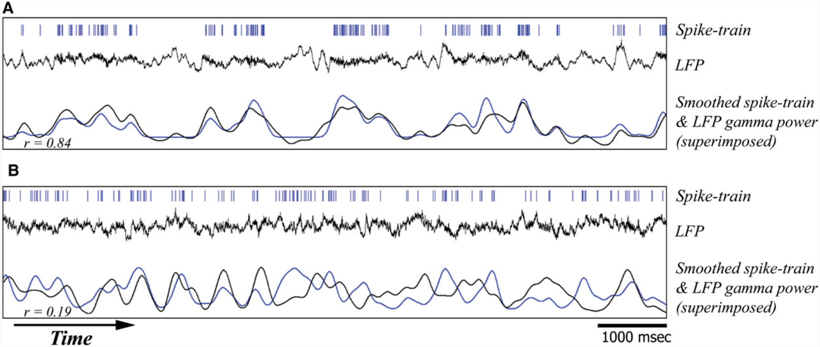
Large-scale brain network



- network node: mean field modeling
- connectome: connectivity and time delays

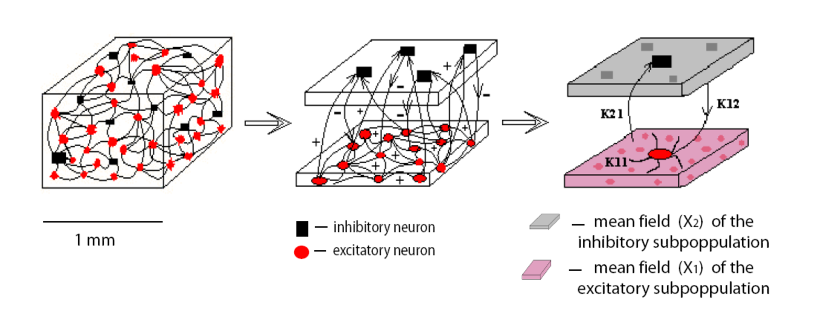
Oscillations in neuronal populations

LFP gamma power vs spiking neurons



Nir et al., Curr. Biol. 2007

Mean field model



Brain Network Model

Basic building block: neural mass models

- mean field description of dynamical behaviour of coupled neuronal populations

Brain network model

- whole-brain network of coupled NMMs
- nodes: brain regions / cortical patches governed by NMM
- edges: white matter connectivity (connectome)

Numerical implementation: The Virtual Brain

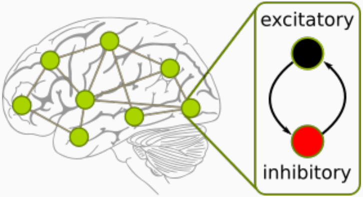
Brain Network Model

- noise
- external stimulus
- population dynamics

$$\dot{\Psi}(x,t) = \boxed{\Lambda(\Psi(x,t))} + \boxed{I(x,t)} + \boxed{\xi(x,t)} + \boxed{\int_\Gamma g_{local}(x,x\prime)S(\Psi(x\prime,t))dx\prime} + \boxed{\int_\Gamma g_{global}S(\Psi(x\prime,t-\frac{|x-x\prime|}{\nu}))dx\prime}$$

- short-range spatially invariant input
- long-range spatially heterogeneous input

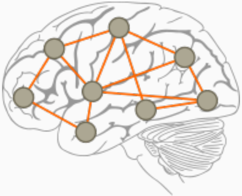
BNM components: node dynamics



Mean field models of neural populations

- deterministic (drift) component of the dynamics
- coupled ODEs: state variables and parameters
- firing rates: Wilson-Cowan, Jansen Rit, Brunel Wang
- local field potentials: Wong-Wang, Stefanescu-Jirsa
- phenomenological: epileptor, planar oscillator

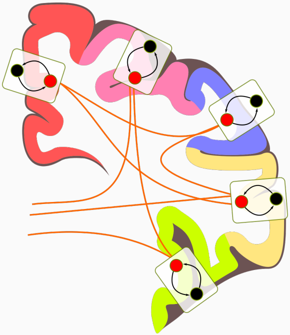
BNM components: connectivity



Coupling / connectivity

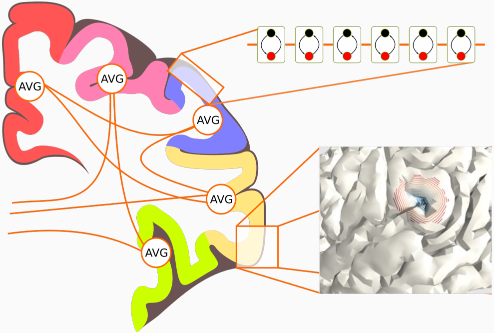
- derived from empirical DWI data
- space-time structure: finite propagation speed $\implies$ delays

BNM components: regional connectivity



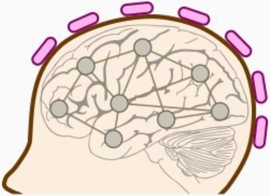
- nodes ≈ 60–300: brain regions defined by cortical parcellation + subcortical areas
- only connectome-based connectivity
- computationally inexpensive

BNM components: surface connectivity



- nodes $> 10^5$: discretized surface + subcortical areas
- connectome-based and surface connectivity

BNM components: forward solution



Biophysical forward solution

- generates simulated data comparable to neuroimaging
- M/EEG, SEEG, fMRI

TVB architecture

TVB overview

`tvb-library`

- the simulator core
- Python + NumPy
- fairly lightweight

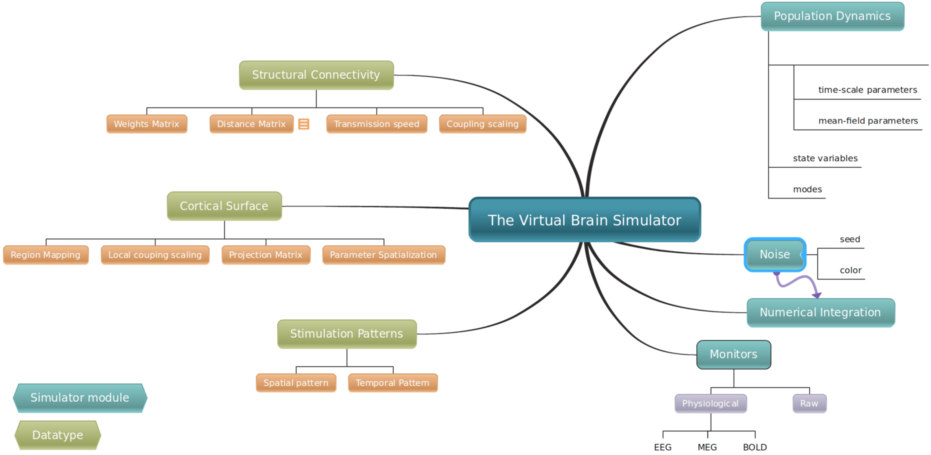
`tvb-framework`

- GUI (HTML5) with visualizers
- data persistence layer
- workflow manager

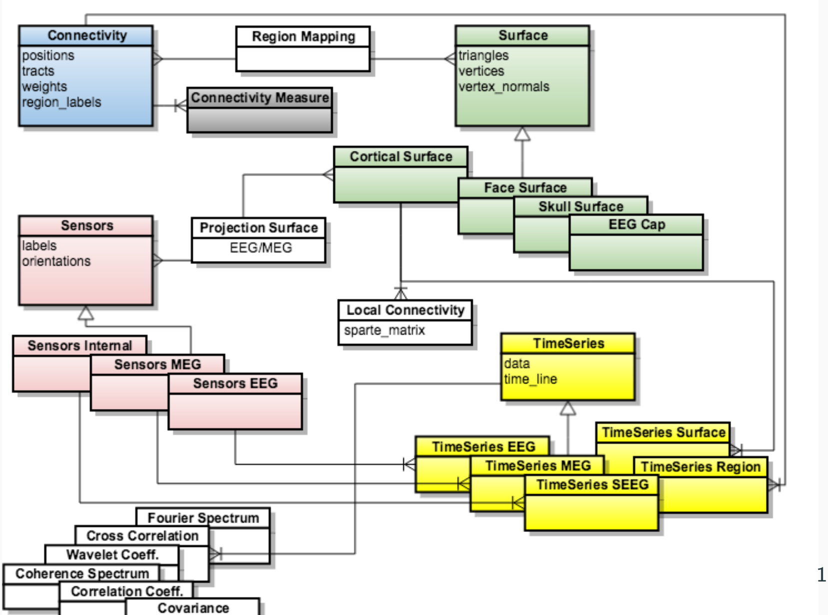
TVB Components: modeling choices

- neural population model
- geometry
    - region based modeling
    - surface based modeling
- connectivity
    - global coupling: connectome
    - local coupling
    - space-time structure (delays)
- noise
- stimulus
- monitors (forward solution)

TVB architecture



TVB datatypes



TVB ecosystem

Application-specific ports

`tvb-epilepsy`

- Bayesian inference in the context of epilepsy
- implementation: STAN probabilistic language

`tvb-hpc`

- C implementation of Wilson-Cowan for parallel parameter sweeps

`tvb-multiscale`

- prototype of TVB-NEST integration

Need for DSL

current Python codebase

- reference implementation
- easily "hackable"

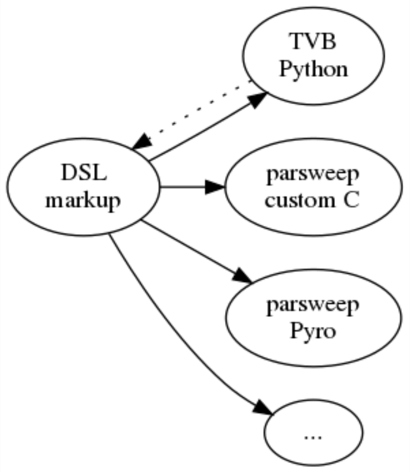
NeuroML(-like) model description

- focus on population models and connectivity
- integrator etc. up to the backends

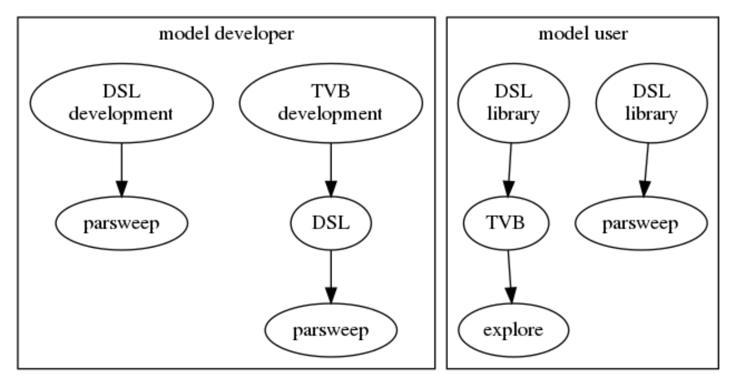
Backends

- translate the DSL
- place for e.g. performance engineering

Backend: a code generator



DSL workflow examples

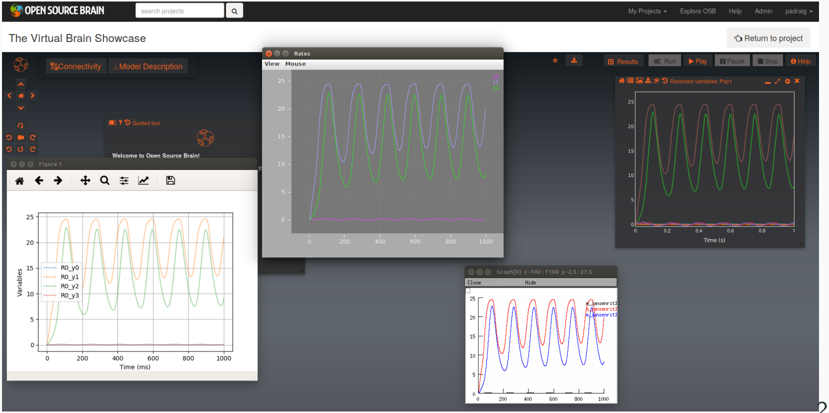


Model example: 2D oscillator

```
class G2DO:
    "Generic nonlinear 2-D (phase plane) oscillator."
    state = 'W V'
    limit = (-5, 5), (-5, 5)
    cvar = 'c_0'
    const = {'tau': 1.0, 'I': 0.0, 'a': -2.0,
             'b': -10.0, 'c': 0.0, 'd': 0.02,
             'e': 3.0, 'f': 1.0, 'g': 0.0,
          'alpha': 1.0, 'beta': 1.0, 'gamma': 1.0}
    drift = (
        'd * tau * (alpha*W - f*V**3 + e*V**2 + ...)',
        'd * (a + b*V + c*V**2 - beta*W) / tau'
    )
    diffs = 1e-3, 1e-3
    obsrv = 'W', 'V'
```

TVB-lems showcase by Padraig

- LEMS representation of a single node
- runs in jNeuroML, NEURON, and OSB



Future directions

Elaborate the NeuroML showcase

- network description
- new population model base type?

Work with backend developers

- FZJ for `tvb-multiscale`
- Charite for `tvb-hpc`
- INS for `tvb-inference`

Have fun with CUDA in free time...

Summary

- Brain Network Model: mean-field, connectome-constrained, personalized, whole-brain model
- The Virtual Brain implements the Brain Network Model
- we are looking for DSL to tie together a growing ecosystem
- NeuroML looks like a good candidate

Thank you for your attention.