

How to use this code?

This repository contains the code to run the model-free and model-based approaches as well as some inputs using the Schaefer 1000 7 RSNs parcellation. The repository is organized in 6 folders: (1) model-free, with the code to run whole-brain turbulent-like dynamics; (2) model-based, with the code to run the Hopf whole-brain model with intact structural connectivity; (3) model-based-lesions, with the code to run the Hopf whole-brain model after the simulated attack approach; (4) CONN_preprocessing, the outputs from the CONN toolbox after denoising; (5) time-series, the derivatives needed for the model-free and model-based approaches; (6) lesion-masks, the code needed for the simulated attack approach. A diagram showing the structure of the code, the helper functions and the batch files needed for the HPC environment are depicted in Figure 1.

Model-free approach: Turbulent-like dynamics measures

The core of the model-free approach consists of 2 steps: (1) getting the derivatives from time-series in the right format after denoising in the CONN toolbox, and (2) calculating the turbulent-like dynamics measures. Step 1 is performed with the scripts *timeseriesforturbulence_ON_HC_Sch1000_tp1.m*, *timeseriesforturbulence_ON_HC_Sch1000_tp2.m* and *timeseriesforturbulence_ON_tbi_Sch1000_tp123.m*. Step 2 requires the outputs from step 1 as inputs (these can be found in the time-series folder) and the scripts *turbulence_empirical_measures_hc_sch1000_tp1.m*, *turbulence_empirical_measures_hc_sch1000_tp2.m* and *turbulence_empirical_measures_tbi_sch1000_tp123.m*.

Model-based approach: without lesions

The code in this folder allows to build a Hopf whole-brain model with Stuart-Landau oscillators for each data set. First we need to calculate the power spectrum peak of each region in the empirical data with the scripts *Compute_Hopf_Freq_hc_tp1.m*, *Compute_Hopf_Freq_hc_tp2.m* and *Compute_Hopf_Freq_tbi_tp123.m*. After this, we need to compute the empirical functional connectivity as a function of the Euclidean distance between equally distant brain regions within the inertial subrange. The user should run the scripts *Empirical_corrfcn_hc_tp1.m*, *Empirical_corrfcn_hc_tp2.m* and *Empirical_corrfcn_tbi_tp123.m*. Next, we can run the Hopf whole-brain model to find the optimal working point. To do so, we run 100 simulations for each of the coupling parameter (G values) in the range (0:3) with 0.01 steps. This is done with the scripts *hopf_DTI_Grange_hc_tp1.m*, *hopf_DTI_Grange_hc_tp2.m* and *hopf_DTI_Grange_tbi_tp123.m*. The output variable *err_hete* contains the difference between the simulated and empirical functional connectivity. This variable is used by the scripts *get_working_point_G_tbi_tp123_hc_tp1.m* and *get_working_point_G_tbi_tp123_hc_tp2.m* to find the optimal fitting of the model for each data set. To perturb the model, we need to use the scripts *pert_infocapacity_susc_hc_tp1.m*, *pert_infocapacity_susc_hc_tp2.m* and *pert_infocapacity_susc_tbi.m*. In this step, we are introducing random changes in the local bifurcation parameter a of each brain region in the range (-0.02:0) and we run 100 simulations for each trial (total trials= 100). The output variables from these scripts contain the information encoding capability and the susceptibility for each trial averaged across simulations. The information encoding capability and susceptibility are the std and mean of the difference between the perturbed and unperturbed modulus of the Kuramoto local order parameter, respectively. These variables are used by the scripts *get_InfoCap_Susc_tbi_t123_hc_tp1.m* and *get_InfoCap_Susc_tbi_t123_hc_tp2.m* to calculate the mean from all trials and export the values for statistical analyses.

Simulated attack approach

The folder lesion-masks contains the binary lesion masks for each TBI patient as provided by the curators of this data set (subfolder lesions_renamed) and the code needed to create the lesion mask array used in the Hopf whole-brain model. The binary lesion masks were normalized in SPM and resampled to the Schaefer parcellation space using the Draw VOI tools in mricron v.1.0.20201102. The normalized resampled binary lesion masks can be found in TBI_ON_turbu_Hopf/lesion_masks/overlap_Schaefer/Schaefer1000_wotbi09/. Note that TBI sub-01 had no visible lesion and that normalization of TBI sub-09 failed and no lesion mask was used for this patient. The normalized binary lesion masks are used by the bash script *get_lesion_overlap_Schaefer1000_mricron.sh* to calculate the overlap between the lesion mask of the patient and each parcel in the Schaefer 1000 parcellation. The outputs from this script include the volume for each of the Schaefer parcel (*Schaefer_volume_combined.txt*) and the overlap of the lesion mask for each parcel and TBI patient (*overlap_volume_combined_wsub-tbixx.txt*). These files are loaded using the *get_lesion_mask_array_Schaefer1000.m* script, which calculates the percent overlap with each Schaefer parcel by dividing the overlap between the lesion and each parcel by the parcel's volume (124-26). This is used to find the nodes to attack based on 1.5, 2, 3 and 4 std from the overlap volume in all patients. Therefore, the binary lesion masks are created by setting to 0 the values for the nodes to attack (155-91), that is, complete deletion. For the weighted approach (193-193), the weight is computed by calculating the number of patients with the lesioned node divided by the total number of TBI patients. Then the value in the lesion mask matrix between that node and the rest of the brain is set to 1 - weight. The resulting lesion mask arrays for each approach and threshold are depicted in Figure 2.

Model-based approach: with lesions

The lesion mask arrays are introduced at 2 steps for the model-based approach by multiplying the structural connectivity matrix (variable C) by the corresponding lesion mask array: (1) calculation of the Hopf whole-brain model and (2) in silico perturbation. Therefore, the scripts calculating the power spectrum peak of each region and the empirical functional connectivity are the same as when using the intact C. This repository contains the code for the 2 thresholds and simulated attack approaches reported in the paper. This code can be found in the folder model-based-lesions. The subfolders contain the scripts that were specifically modified to account for the effect of the lesion when computing the Hopf whole-brain model as well as the lesion mask array. For example, the subfolder TBI_ON_Lesions_mask1halfSD_bin contains the scripts for: (1) fitting the model (scripts *hopf_DTI_Grange_tbi_lesions_mask1halfSD_bin.m* and *get_working_G_tbi_mask1halfSD_bin_hc_tpx.m*) and (2) in silico perturbations (scripts *pert_infocapacity_susc_tbi_lesions_mask1halfSD_bin.m* and *get_InfoCap_Susc_tbi_mask1halfSD_bin_hc_tpx.m*). The rest of the subfolders are structured following this logic.

Visualization

So far, we have presented the code to get the results in this paper. The visualization folder contains the Matlab and Python code that were used to create Figures 1-5 and FigS1. Note that the plots for FigS2 were obtained with the code in the lesion-masks folder.

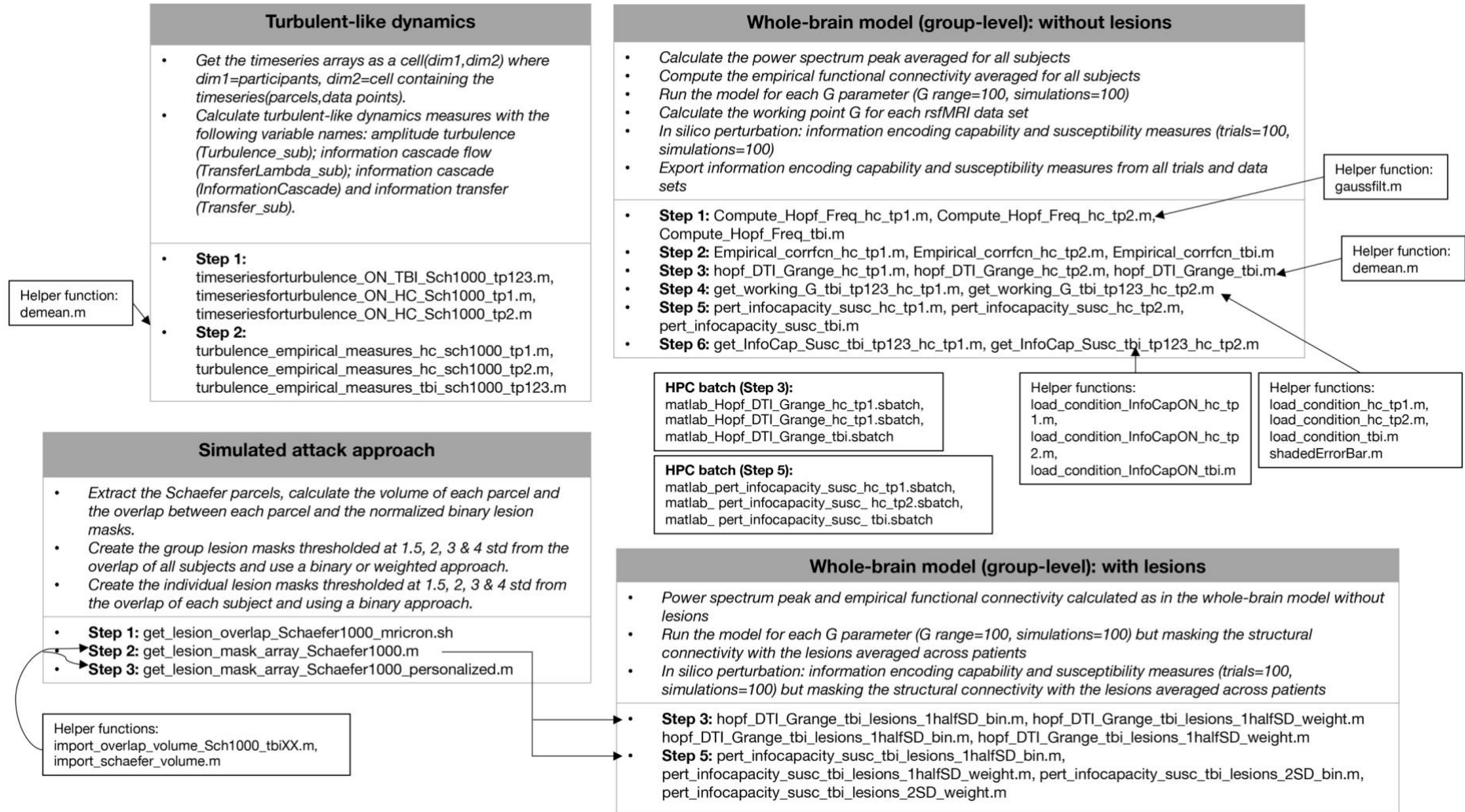
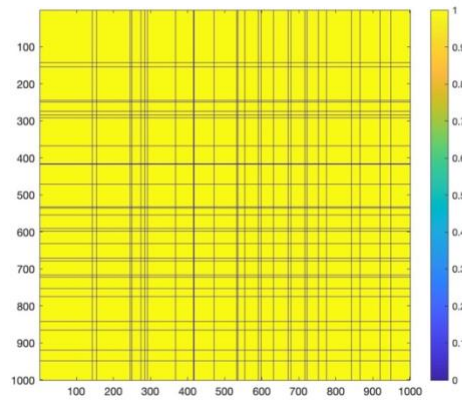
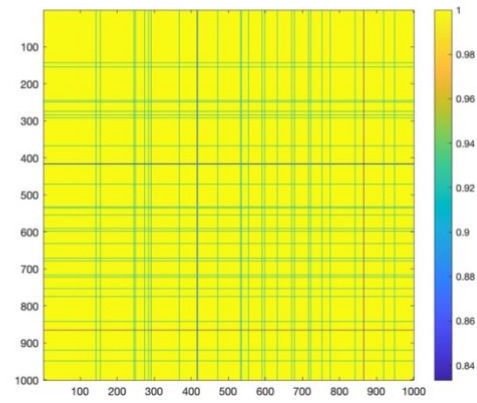


Figure 1. Diagram with the structure of the code in this repository.

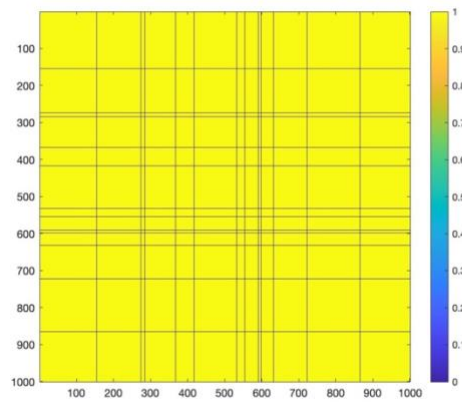
Lesion mask array 1.5 SD binary approach



Lesion mask array 1.5 SD weighted approach



Lesion mask array 2 SD binary approach



Lesion mask array 2 SD weighted approach

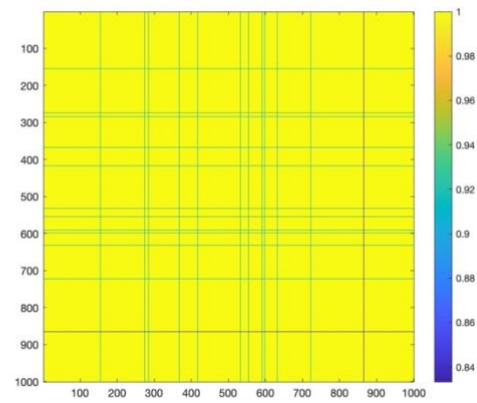


Figure 2. Lesion mask arrays used for the simulated attack approach.