SEEG SIGNAL SIMULATOR

Here we describe the different functionalities of the software that was developed in order to simulate SEEG signals for the many parameters and factors that intervene in the relationship between the activity of neuronal sources and the signals collected at the level of electrode contacts. Special attention was paid to the development of user-friendly graphical interfaces that allow users to visualize the main simulation elements (virtual brain, electrode, epileptic patch) and easily set the simulation parameters.

1.1. Introduction

The past decades have witnessed an increasing interest in scanning brain activity accompanied by a vast development in the invasive and noninvasive recording technologies. Epilepsy profited from this advancement and epileptiform events can now be recorded from several levels, starting from microscopic single neuron passing through mesoscopic groups of neural population to the global macroscopic scanning of magneto/electrophysiological signals produced by activation of large-scale brain regions. In order to understand the basic mechanisms of the underlying sources of the recorded signals and/or scanned images, studies confirmed that reliable models inspired from the biological and biophysical features can immensely assist in developing new hypotheses that can guide experiments and ultimately lead to new discoveries (Destexhe, 1998).

In response to this demand, several toolboxes and software have been developed for different levels of neuronal simulations. Some are focused on simulating extracellular local field potentials (LFPs) from multi-compartmental detailed level model (Lindén et al., 2013), others took into consideration the spatial and frequency filtering characteristics of the extracellular potentials (Tomsett et al., 2015). More convenient modeling approaches to simulate macroscale signals starting from interconnected mesoscale models were provided as a neuroinformatics platform that comprises a graphical user interface and an ability to switch to programming interface (Matzke et al., 2015; Melozzi et al., 2017). One important type of simulation in epilepsy is replicating electrophysiological brain activity recorded by clinical electrodes, this requires a biophysical extension for the mesoscale model generating the temporal dynamics. Despite its importance in clarifying the spatiotemporal link between recorded signals and their corresponding sources (Cosandier-Rimélé et al., 2007; Cosandier-Rimélé et al., 2008), an effective and easy-to-use software remains to be developed.

Here, we present an interactive graphical user interface integrated in a single 3DSlicer Python module that will assist in simulating intracerebral signals. Exploiting 3DSlicer (Pieper et al., 2004) capabilities in visualization of 3D meshes, MRI and CT scan volumes we were able to implement a flexible user interface that facilitates (1) inserting virtual intracerebral electrodes of realistic shape and geometry at the desired position and orientation, (2) delineating the cortical neural field model responsible for the generation of pathophysiological temporal dynamics (3) solving the forward problem by computing the contribution of each neural mass on each contact of every electrode and concatenating the result in in a Lead Field Matrix, (4) plotting the simulated signal on any desired contact of each electrode.

The module named "SEEG Signal Simulator" (SSS, Triple-S) has proved its ability to replicate several epileptiform events such as interictal spikes, high frequency oscillations of both criteria ripples(80 Hz – 200Hz) and fast ripples (>200Hz) as well as ictal activity. A key contribution to the features provided by 3D slicer is the interactive modifications that the user can directly execute onto the virtual scene without even touching the graphical interface components. This will allow researchers with moderate computer skills to easily handle it and consequently exploit its capabilities to the maximum. We will explain different features of the module, theoretical basis behind the neural field simulation and finally evaluate its performance by comparing the resulting simulated signals with real SEEG signals.

1.2. Biophysics behind Triple-S

The signals recorded on intracerebral electrodes stem from postsynaptic potentials generated by neuronal populations, mainly at the level of pyramidal cells. Studies have shown that pyramidal cells of the neocortex are aligned parallel to each other and orthogonal to their corresponding surface forming a source/sink configuration similar to an active current dipole (Niedermeyer and da Silva, 2005) thus creating an cortical dipole layer. This spatial organization of pyramidal cells in the cerebral cortex enables distant electrodes, as in the case of clinical electrodes, to detect variations of the generated extracellular potentials. Moreover, some studies indicate that action potentials generated by the pyramidal cells play a role in shaping the collected signal especially on microelectrodes (Worrell et al., 2008), however their contribution is considered to be minor in case of macroelectrodes and could be neglected.

To simulate PSPs produced by the neuronal populations neural mass models based on biologically-inspired parameters can be used to simulate and analyze pathophysiological mechanisms. Then, signals can be generated by projecting PSPs onto electrode contacts while taking volume conduction factors into account. In this process one should solve what so called the forward problem and take into account the

relative spatial geometry between the dipole sources and the observation points (electrode contact) as well as the conductivity of different tissues of the volume conductor. Here we will give a brief overview on the neuronal mass model used to simulate the temporal dynamics although it is not implemented in Triple-S module, as well as various available neuronal mass models whose output is compatible with this module as long as formatting considerations are applied. Then we provide a detailed description of the implemented forward problem solver and Lead Field matrix fabrication.

1.2.1. Neural Mass modelling of PSPs

Computational models at various spatial and temporal scales help us to understand the essential mechanisms and fundamental configuration of neuronal sources as well as their neurophysiological processes which reflect in as images (MRI, fMRI) and signals (MEG, EEG, iEEG, single cell recording) captured by many types of available recording technologies. On the mesoscopic scale, lying between microscopic (single neurons recordings) and macroscopic (whole brain dynamics), neural mass models (NMMs) account for the neural interactions among different types of neuronal elements (pyramidal cells, interneurons, thalamic input) and may consider several physiological (collateral excitation, feedback and feedforward inhibition) and pathophysiological mechanisms some of which are related to neurological disorders (depolarizing $GABA_A$, hyper-excitability).

Generally, the basic properties of elementary interactions between building blocks of NMMs are not explicitly described, rather a statistical representations of collective actions of populations of neurons are implemented. The basis for NMMs dates back to the seventies with Lopes da Silva et al. and Freeman et al. (Da Silva et al., 1976) then these models were enhanced and enriched by several contributions and are now widely used to simulate all types of brain activity (see Table 2).

In the model used in this research work, the average value of extracellular PSPs is considered as the state variable and parameters like average connectivity constants, average amplitude of density of action potentials and average firing rate of different elementary types of neurons are taken into account. The model described uncoupled cortical columns in which pyramidal cells, i.e. the main component, somatic (fast) and dendritic (slow) targeting interneurons interacted via synaptic transmissions. Thalamic input, depolarizing GABA and feedforward inhibition mechanisms were also integrated within the model. The set of differential equations describing the time variation of PSPs of different component were solved using Runge-Kutta numerical method.

In general, any NMM that produces time varying PSPs of pyramidal cells as an output is appropriate to be used as a neuronal source of the Triple-S module under the condition that the file format and header is in accordance with the described formats in the "Input Files" section. Different modelling approaches already exist where some are used to explain the mechanisms underlying the generation of ictal HFOs (Demont-Guignard et al., 2012; Ibarz et al., 2010), others were designed to study absence seizures and tonic-clonic seizures (Breakspear et al., 2005) and to investigate potential mechanisms underlying generalized seizures (Yan and Li, 2011). Finally, it is worth to mention that the compatibility with the presented module also depends on the anticipated goal of the study, i.e. if PCs contribution on intracerebral electrodes is the main addressed subject.

1.2.2. From extracellular PSPs to LFPs

Given the time varying momentum, position and orientation of the dipoles, their electric contribution could be calculated on an observation point using the volume conductor theory. Taking into consideration several justifiable approximations simplifies the calculations needed to solve the forward problem. For instance, using the raw Maxwell's equations to determine how the electric field produced by the dipoles project on the electrodes may become extremely complicated and can be effectively avoided if the quasi-static approximation is assumed. This implies that the delay between the instant of formation of the target electric field and the actual time at which the electric field is detected by the electrode contacts is neglected due to the slow nature of neural activity (see (Hämäläinen et al., 1993) for argument). Another essential approximation is to consider the volume conductor medium, mainly gray and white matter, to be isotropic homogenous mediums i.e. conductivity is the constant in every direction, and that capacitive and inductive characteristics of the medium are also ignored i.e. conductivity will be a constant real scalar. These approximations seem to be applicable as long as the recorded signals are of low frequencies, typically less than few thousand hertz.

Mathematically speaking, each current dipole is characterized by its location z at the barycenter of each triangular element and its orientation perpendicular to its corresponding surface mimicking the real alignment of pyramidal cells in the outer cortical layers (see Fig 39). In Triple-S module, the barycenter coordinates, the director vector of the normal associated to each triangle and its corresponding area is automatically calculated once the user confirms the delineated patch. At each location z, the time varying moment of each dipole is given by $m_i(z,t) = M(z) \cdot X_i(t)$. Where $M(z)$ is the dipole moment surface density obtained by multiplying an average value of the moment surface density of cerebral cortex (525

nA.mm (Cosandier-Rimélé et al., 2010)) by the surface area of each neural population i.e. surface area of each elementary triangle. $X_i(t)$ denotes the time varying PSPs of the pyramidal cells of each neural population obtained from a NMM, and $m(z,t)$ is the overall spatio-temporal moment characterizing the electric activity of population i, where i varying from 1 to N and N is the total number of populations. The LFPs collected at one contact j (j =1...15 considering that the electrodes are modeled with 15 contacts each) located at distance r from a patch composed of N neuronal population is given by:

$$v_j = \sum_{i=1}^{N} \frac{\left\langle m_i \middle| u_{r(i,j)} \right\rangle}{4\pi\sigma r_{i,j}^2} \tag{12}$$

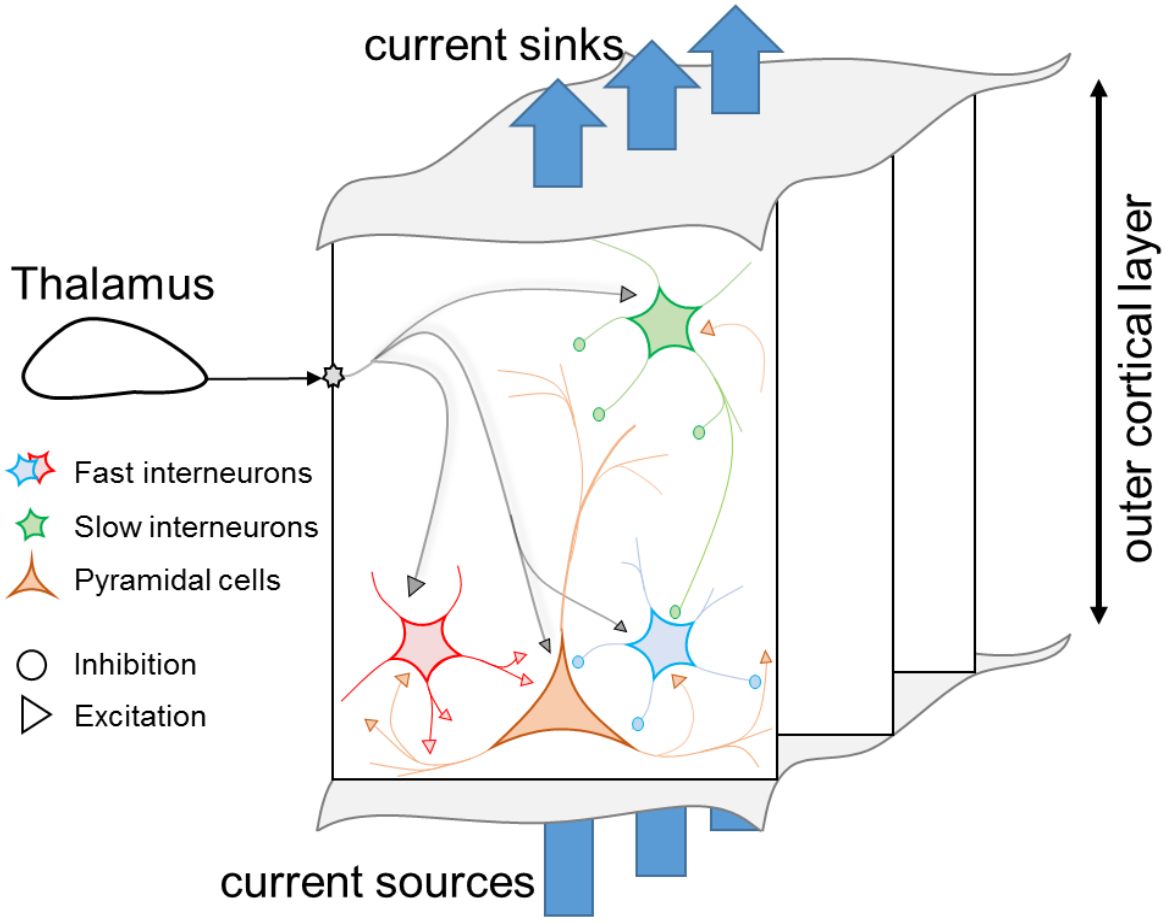

Figure 1. Schema representing the outer cortical layer. Current dipoles are arranged parallel to the pyramidal cells (orange). Three types of interneurons are presented: slow inhibitory (green), fast inhibitory (blue), fast excitatory (red). It is worth to mention that the inhibitory/excitatory characteristic is a property of the synaptic connections, but for simplicity it is accounted for interneurons. Each neuronal population and each element in the population receives an afferent input from the subcortical regions (thalamus for instance)

Where $\langle \ | \ \rangle$ denotes the scalar product operation, m_i is the spatio-temporal vector representing the dipole of population i and $u_{r(i,j)}$ denotes the unit vector directed from dipole i to sensor j

Generally, to save computational cost the value (weight) describing effect of each dipole on each contact is calculated without the need for the variable temporal dynamics and is concatenated into what so called "Lead Field Matrix" (gain matrix). This will allow the recurrent use of the same spatial configurations i.e. same lead field matrix, but each time with a different time varying PSPs. To do so, only the constant spatial vector component of m_i is taken into account and the values obtained from each potential equation will be a weight (denoted by $a_{i,j}$) that defines the linear relationship between neuronal sources and collected LFPs. A typical Lead Field matrix is shown below:

$$\begin{array}{c} \\ C_1 \\ C_2 \\ \vdots \\ C_{15} \end{array} \begin{array}{c} \begin{array}{cccc} NP_1 & NP_2 & & NP_N \end{array} \\ \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & & a_{2,N} \\ \vdots & & \ddots & \vdots \\ a_{15,1} & a_{15,2} & \cdots & a_{15,N} \end{pmatrix} \end{array}$$

Where NP_i stands for neural population of index i. Cj stands for contact of index j. $a_{i,j}$ is the coefficient (weight) that determine the contribution of signal generated by population i on contact j. To compute the LFPs as observed on each contact it is sufficient to multiply the Lied Field matrix by the temporal dynamics matrix.

1.3. Triple-S Functionalities

In this section we provide a detailed description on different interactive and graphical features embedded in Triple-S and concisely clarify how to benefit from them. Briefly, Triple-S is designed to assist in three main phases while simulating intracerebral EEG: (1) insert intracerebral electrodes in desired location, (2) delineate cortical patch that will be used as a source for brain activity and (3) solve forward problem and simulate LFPs.

1.3.1. Insertion of Electrodes

In clinical recordings, the relative position of the intracerebral electrodes to the source generating pathological activity plays a significant role in interpreting the recorded signals and consequently

pinpointing the epileptogenic zone to be removed by a resective surgery. Triple-S module is designed for two main purposes, (i) simulate real case scenarios where the coordinates of electrode contacts is known in the real recordings; (ii) simulate purely virtual cases where the user defines the position and orientation of the intracranial electrodes.

(i) In the first case, the coordinates of the electrodes contacts could be extracted from CT scan (see Fig. 40A) performed during intracerebral recordings as MRI can't be done due to the metallic nature of the contacts. The procedure that enables the extraction of the contacts coordinated from CT scans is rather complicated, but there is some software and toolboxes especially designed to facilitate this task (Azarion et al., 2014). However, the difference in coordinate systems and number of registered slices between

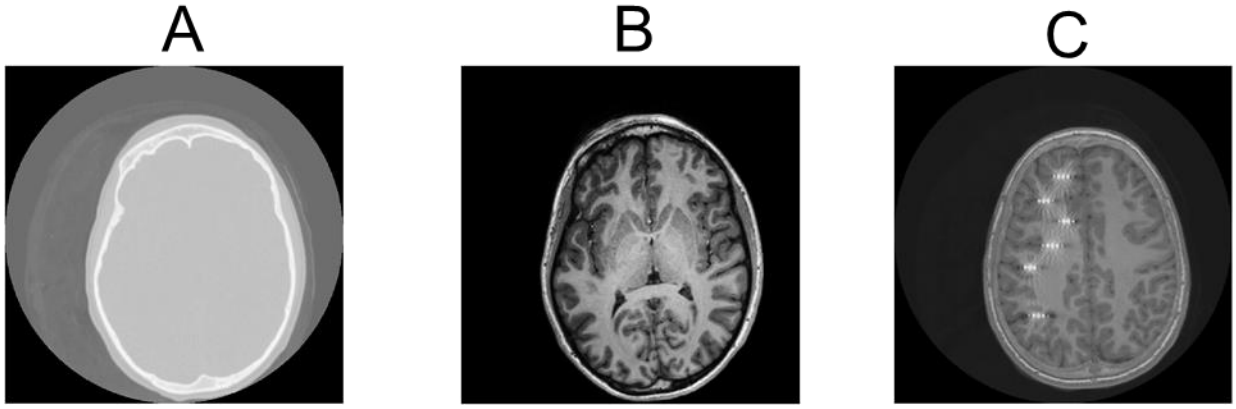

Figure 2. Coregister MRI and CT scans. (A) Axial view of a typical CT scan. **(B)** Axial view of a typical MRI scan. **(C)** Coregistered MRI and CT scans, electrode contacts are visible as bright dots inside the brain structure.

different systems used to generate MRI and CT scan images hinder the procedure of placing electrode contacts in the correct location. This incompatibility comes from the fact that the brain mesh can't be extracted from CT scan (X-ray imaging is not suitable to vividly capture the details of the soft tissue forming the cortex) and it can be only obtained after segmentation and triangulation of MRI images performed as a post-surgical evaluation (see Fig. 40B). To solve this issue, one must use co-registration algorithms that are implemented in several freely available toolboxes (Arnulfo et al., 2015; Narizzano et al., 2017) (see Fig. 40C).

After obtaining the coordinates of the contacts in the co-registered reference system, the user can enter the position and orientation of each electrode in the available table in the following manner:

- (x_0,y_0,z_0) the Cartesian coordinates of the deepest contact of the electrode
- (ψ, θ) the azimuth angle and the polar angle respectively, these angles can be calculated only if the electrode is considered to be a straight segment composed of different linear points (contacts).

Figure 41A shows a typical intracerebral electrodes where its tip (deepest contact in green) has a fixed position characterized by the set (x_0,y_0,z_0) and with the azimuth and polar angle (ψ, θ) represented in the system of reference. Figure 41B shows an example of the provided table in the graphical user interface where the user can enter the above mentioned coordinates for any number of electrodes. In the given example the coordinates of 5 electrodes are provided and in Fig. 41C the considered 5 electrodes are shown in different slices (parietal, coronal, and axial) of the brain as illustrated by 3D slicer.

(ii) In the second case, the coordinates of the electrodes are not extracted from real recordings, hence the user will place the electrode at a position that most suits the case of the target simulation. Since visual inspection and direct interaction with the virtual environment is much easier then blindly guessing the coordinates we modified the interactive features permitted by the virtual environment of the 3D slicer. Usually, the user is not allowed to directly modify the relative positions of the elements existing within the 3D environment without accounting for a linear transformation i.e. the relative positions of the elements in the virtual scene are not modified except from the control interface. Here we enhance the

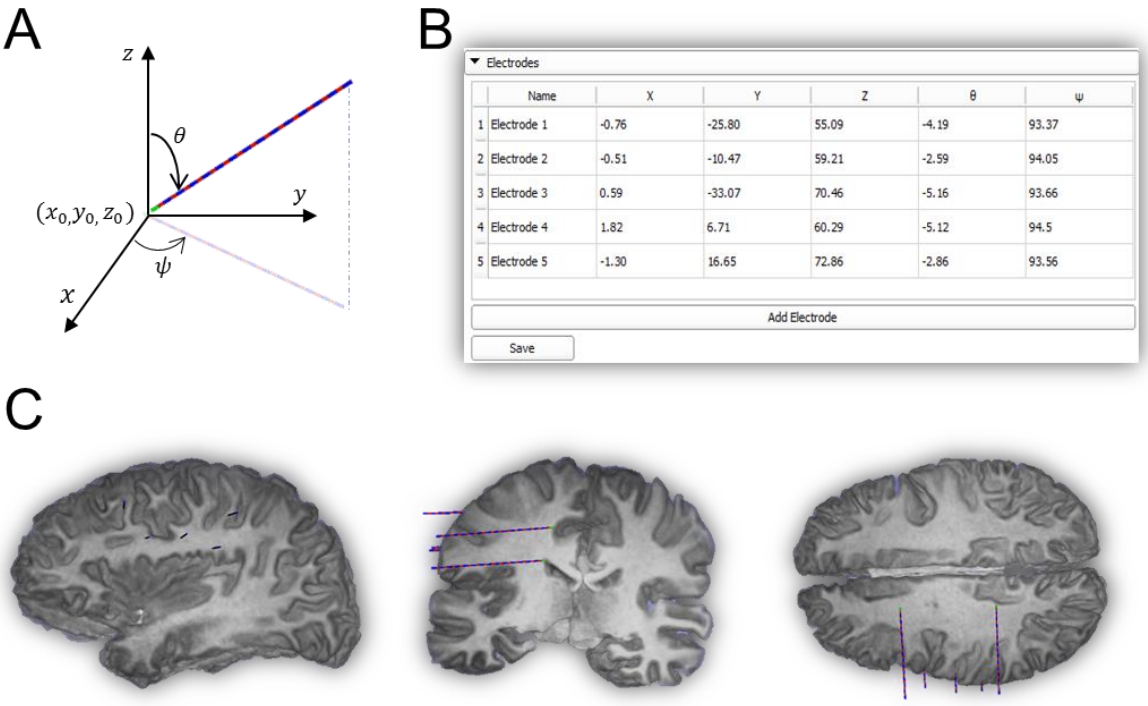

Figure 3. Virtual electrode insertion in virtual brain environment. (A) A schematic representation of a typical intracerebral electrodes with its corresponding coordinates. **(B)** Electrode table provided by the interface where the user can change the coordinates of the electrodes whose position and direction will be updated simultaneously in the virtual environment. Add electrode button is also shown and the save button as well. **(C)** Sagittal, coronal and axial views the brain tissue are shown with the 5 intracerebral electrodes (blue-red).

interactive functionality in a way that the user could handle elements in the 3D scene via the mouse only.

When the button “Add electrode” is pressed an electrode is added at the default position (all coordinates are zeros) and then the user can change the electrode relative position with respect to all other elements in the following manners:

(ii-1) Using left mouse button: the user can select any electrode to modify its orientation by pressing the left mouse button while the cursor is pointing on it. After the selection of the electrode the user can change its orientation while keep holding the mouse button and moving the cursor. In this case the deepest contact of the electrode remains stationary i.e. only angles (ψ, θ) change (see Fig. 42A).

(ii-2) Using mouse wheel: after selecting the electrode, the user can modify its position while keeping its orientation unmodifiable. When the mouse wheel is moved forward or backward, the electrodes move deeper (in the direction of the deepest contact) or outward (opposite to the direction of the deepest contact) respectively (see Fig. 42B). Here the electrode can only move on a straight line.

(ii-3) Using mouse middle button: here the user can translate the selected electrode from one position to another by pressing the middle button and hold it while moving the electrode throughoutt the 3D scene. As depicted in Fig. 42C, the electrode could be translated in any direction and not necessarily on a straight line while keeping the orientation unchanged.

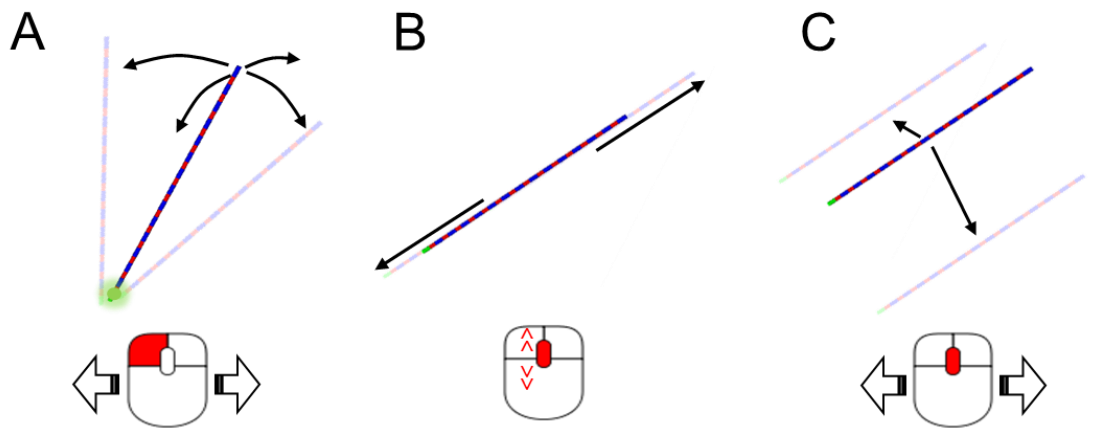

Figure 4. Mouse actions allowing for manipulation of intracerebral electrodes. (A) Fixed tip (green) and 360 degrees of freedom in assigning the electrode orientation by using the left mouse button. **(B)** Fixed orientation, and only one axis of freedom, along the direction of electrode, for specifying the position of electrode by scrolling the middle mouse button. **(C)** Fixed orientation, and two axis of freedom, perpendicular to the direction of electrode, for specifying the position of electrode by holding down the middle mouse button.

The values in the tables of coordinates of each electrode is linked in real time to the position and orientation of each electrode. Hence, when the location of an electrode is changed, the values in the table

for the corresponding electrode is updated accordingly and vice versa. Moreover, the user is able to save the Cartesian coordinates (x,y,z) of the contacts of each electrode as a text file. Unquestionably, we developed the interface to be consistent with the standards of user interface design, for example the addition of a wrong electrode could be corrected by an easy delete of the row of the table corresponding to that electrode or by pressing “d” button while the cursor is pointing to the faulty electrode.

1.3.2. Delineation of cortical patch

During pre-surgical examination, the location and extent of the epileptic zone should be accurately identified as its removal or disconnection from other regions could yield to seizure control. This precise delineation of epileptic zone could be deduced from studies involving simulations by comparing signals obtained from real case intracerebral recordings with signals obtained from simulations where the position of electrodes matches that of the actual case but the source configuration (location, area) varies at each trial. Otherwise, simulation studies could involve situations in which the effect of different source configurations (extent, geometry, multi-patches) on the recorded signals is investigated. Here, we present three different ways to accurately delineate a brain surface area on the cortical mesh:

(1) Rubber band selection: it is the simplest way for defining the cortical patch where the user needs to click the left mouse button when the pointer is over the desired location and drag the pointer over the region to be delineated. A rectangular outline is traced behind the pointer, all the elementary triangular surfaces touched or enclosed the rubber-band outline are selected (see Fig. 43A). The resulting selection accounts only for the front (visible) triangular elements.

(2) Area selection: using this method the user can predefine the desired area of the patch and then click on the cortical surface at the element that will be at the center of the selection. A fully automatic algorithm is then executed to determine the neighboring triangles and include them in the selection and the area increases gradually until the desired area is reached. This method permits the inclusion of triangular surfaces that are in hidden layers and almost impossible to be reached by cursor (see Fig. 43B).

(3) Free selection: this type of selection is the most precise. Here the user clicks on the cortical mesh at any triangular element and holds the button down and moves the cursor over the desired elements in any direction. The color of each selected element changes instantaneously to orange (default) in such a way this method appears as an act of “coloring” the desired area. The user is then free to change the patch color as preferred (Fig. 43C).

More complementary options related to patch delineation were also implemented. For example the user can change the color and transparency of each patch through controls provide in the patch table (see Fig. 43D), the number of populations i.e. elementary triangles, is also given. The visibility of the dipoles (arrows perpendicular to each triangular surface) could also be toggled as well as altering the arrows size and color. The 3D slicer environment provides a unique feature of showing the intersection of the selected brain region, using any of the above described methods, with the MRI volume of the patient.

This allows the user to track, in details, the selected cortical regions on MRI slices in various views (parietal, coronal, axial) as shown in Fig. 43E.

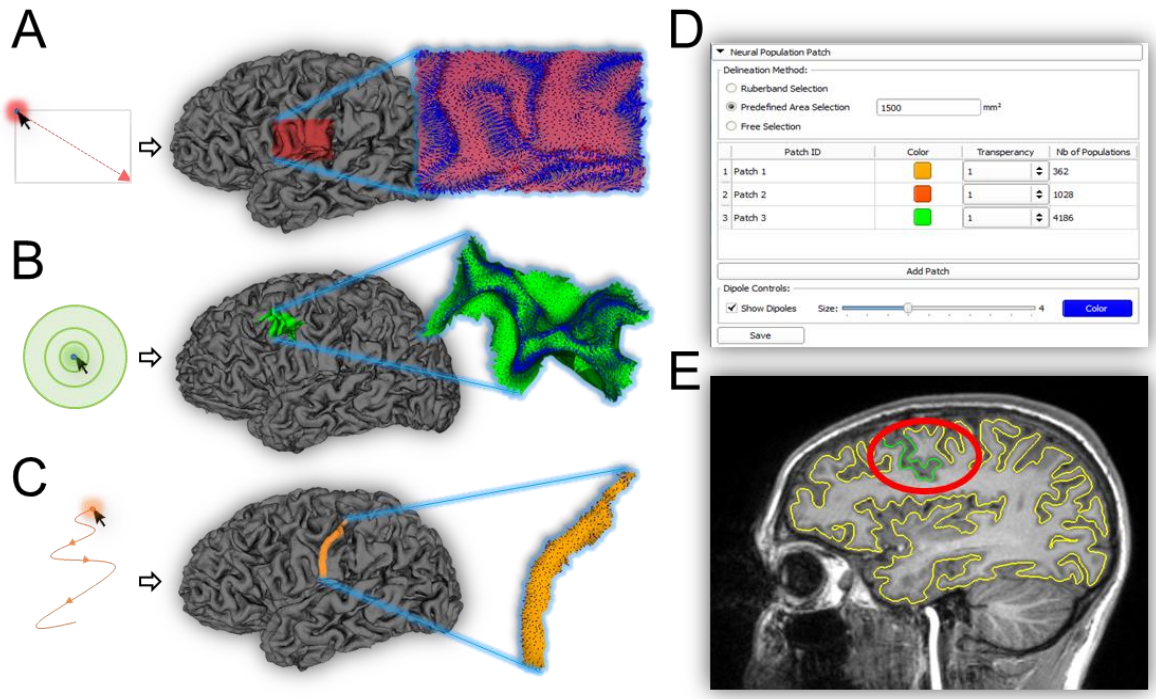

Figure 5. Delineation of the source patch. (A) Rubber band selection method is presented where the user can select the neuronal sources by selecting rectangular area over the cortex. **(B)** Area selection is presented where the user predefine the area of the desired patch and pinpoint the location of its center on the cortical patch. The required patch will be automatically delineated. **(C)** Free selection is represented where the user select the neuronal sources by dragging the pointer over the chosen elementary triangles**. (D)** The interface panel designed for delineating the patch is illustrated. The patch table showing graphical and statistical details about each patch is also presented. The bottom part of the panel is related to the dipoles visibility options (show dipoles, color and size). **(E)** The projection of the green patch is shown on the MRI data of the considered patient.

1.3.3. Simulating LFPs

The morphological and spectral features of the LFPs collected on real intracerebral electrodes hold crucial information determined by the geometrical and temporal dynamics of the sources generating them. In

case of successful simulations of LFPs that accurately replicate the real collected signals we can assume that the spatiotemporal configurations of the simulated epileptic patch resemble those of the real case. This advantage provided by simulation helps in gaining insights into cellular, network or even global mechanisms that contribute to the generation of epileptiform events. Generally, the complexity and precision of the needed simulations varies from one study to another, here we provide 3 modes of projecting PSPs onto intracerebral electrodes:

(1) Temporal Averaging: In this case the temporal dynamics of all neuronal sources is averaged so that all populations will be considered to generate the resulting averaged signal as an output. Here the spatial geometry (location and direction of dipoles) is conserved for each population (see Fig. 44).

(2) Spatial Averaging: In this case, each neuronal dipole will have its own electrophysiological contribution, but the direction of the dipoles is averaged so that all neuronal populations dipolar orientation will be the

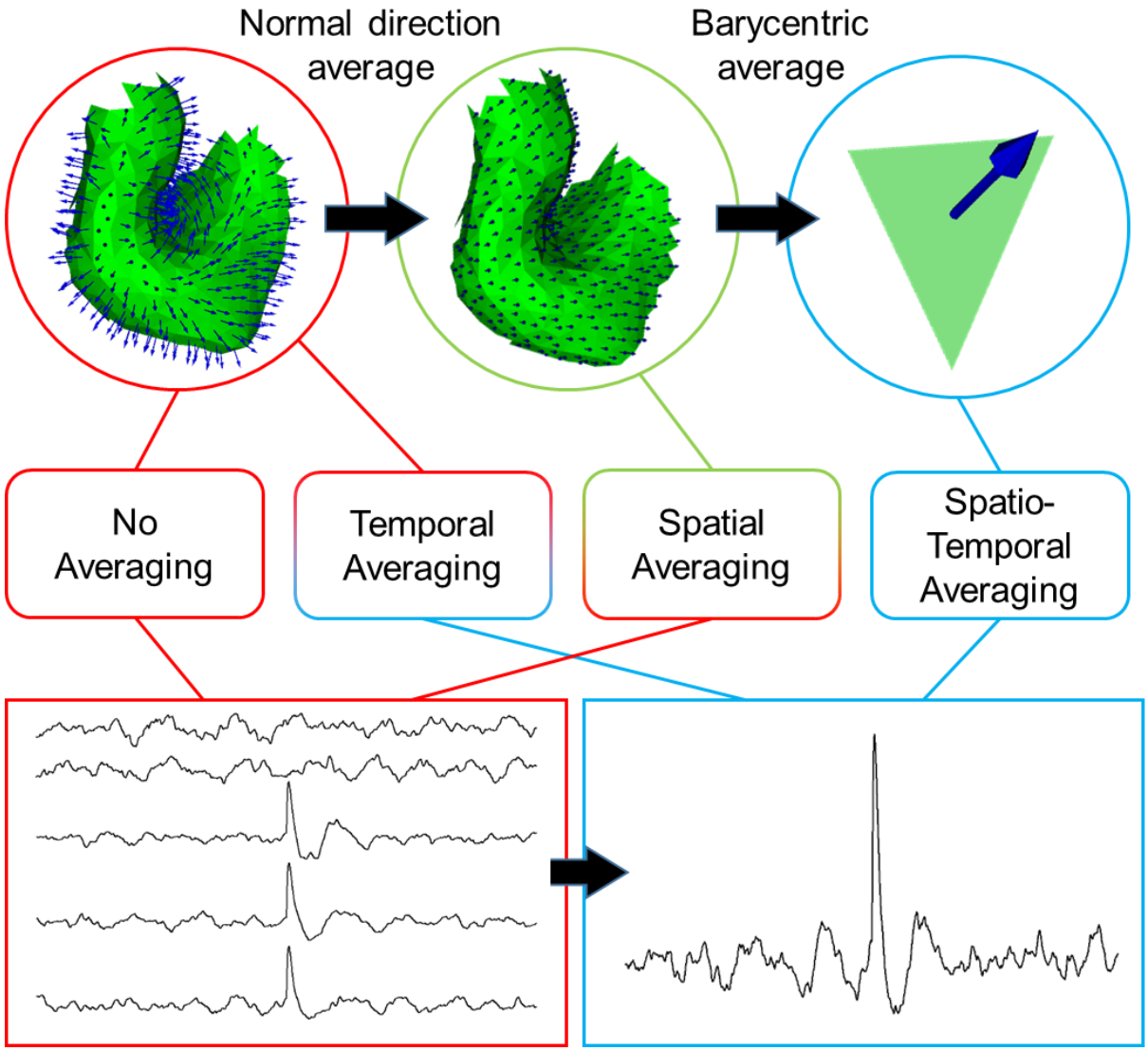

Figure 6. Spatial, temporal and spatiotemporal averaging. When the simulation is done on a raw patch (without any averaging) but each neuronal population get the average of the temporal dynamics (blue square) then it is denoted as temporal averaging. When the simulation is done on spatial patch whose dipole orientations are averaged (green circle) but each neuronal source get a unique temporal dynamics is known as spatial averaging. When the simulation is done on one element representing the whole patch (blue circle) i.e. location and direction averaging, and it takes the average of the temporal dynamics (blue square) then this is known as spatiotemporal averaging.

same (see Fig. 43) but with distinct locations.

(3) Spatio-temporal Averaging: This is the simplest configuration for both, spatial geometry and temporal dynamics, where all populations are reduced to a single neuronal population located at the barycenter of the considered patch and oriented according to the average director vector of all dipoles. (see Fig. 44)

Although several spatial and temporal averaging modes are present, the user may consider to use the raw material without any type of averaging, thus the results will account for neuronal populations with distinct geometrical (location, orientation) and temporal parameters. Since each neuronal population will be taken as a distinct element with its own unique spatial and temporal characteristics then we will gain precession to the cost of extra computational effort.

After choosing the averaging method and uploading the temporal dynamics that comply with Triple-S, the

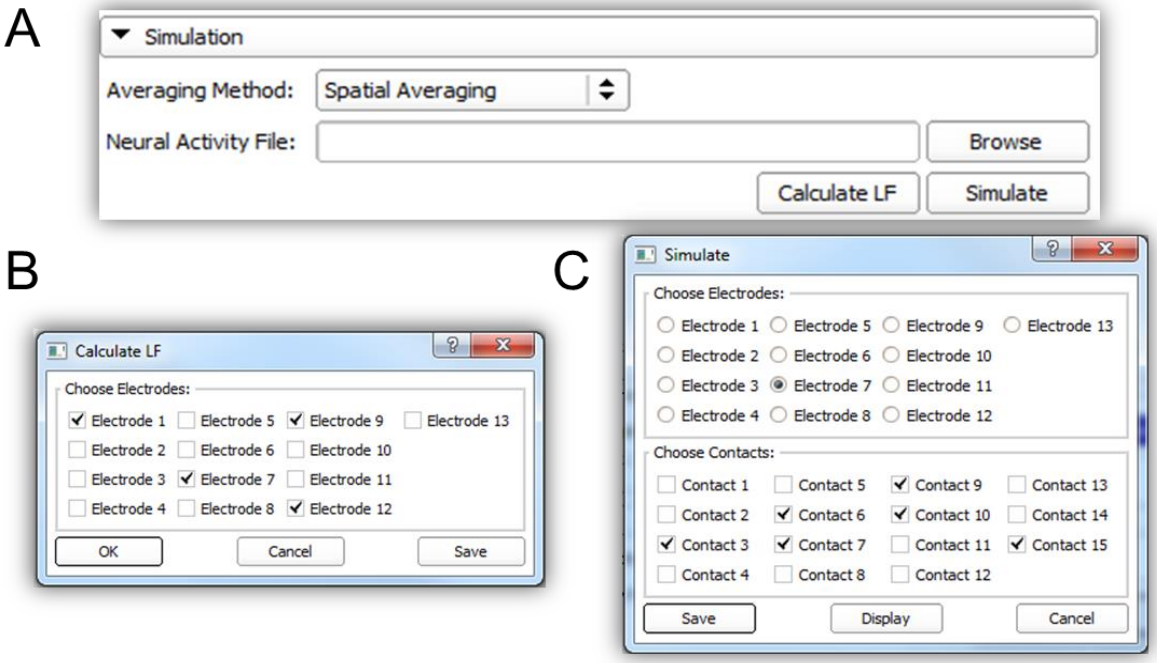

Figure 7. Simulation panels. (A) Panel designed to specify the averaging method and import the neuronal sources temporal dynamics. **(B)** Subpanel that allow the user to specify the electrodes on which he would like to solve the forward problem and calculate the corresponding lead field matrix. **(C)** Subpanel that allow the user to choose one electrode at a time with several contacts and simulate the resulting signals.

user can solve the forward problem and save the result as a Lead field matrix. Finally the user is able to simulate the signals for several contacts of his choice but for only one electrode at a time (see Fig. 45).

1.4. Example: Real Case Simulation

In order to validate the overall quality of the Triple-S module, we used it to simulate two types of epileptic activity appearing on interictal electrodes of 2 different patients suffering from focal drug resistant epilepsy. MRI and CT scan were used to fabricate the 3D cortical mesh and extract position of electrodes contacts. The SEEG recordings of both patients were examined to identify the contacts showing epileptiform events. In the first patient showed high frequency oscillations in the ripple (100-200Hz)) and

fast ripple (>200 Hz) band in the hippocampus. The second patient showed interictal epileptic spikes (broadband frequency) in the temporal lobe region. Triple-S module was then used to simulate HFOs and interictal spikes on the electrodes exhibiting the clearest and highest amplitude of epileptic events (HFOs, spikes). Relying on results obtained from the previously discussed study, the delineated patch comprised neuronal populations generating abnormal fast activity while surrounded by an extended patch generating normal background activity. Both background activity and fast activity was generated by the previously described neural mass model.

As depicted in Fig. 46 and Fig. 47, the simulated HFOs and epileptic spikes were observed on multi-contact intracerebral electrodes which resembles a real case scenario, only 8 contacts are represented for clarity. The observability conditions realized by Triple-S simulations are also in consonance with those of real recordings, in fact, HFOs are visible on only 4 contacts of the same electrode while the spikes are widespread all over the contacts of the considered electrode.

To validate the spectral coherence between real and simulated data, we computed the spectrogram for both real and simulated high pass filtered HFOs (cutoff frequency 30Hz). As Fig. 46 and Fig. 47 show, real and simulated signals share the same spectral features for both type of signals (HFOs, spikes). For a quantitative comparison readers may refer to study 1 Chapter 3 Section 3.1.3 page 47.

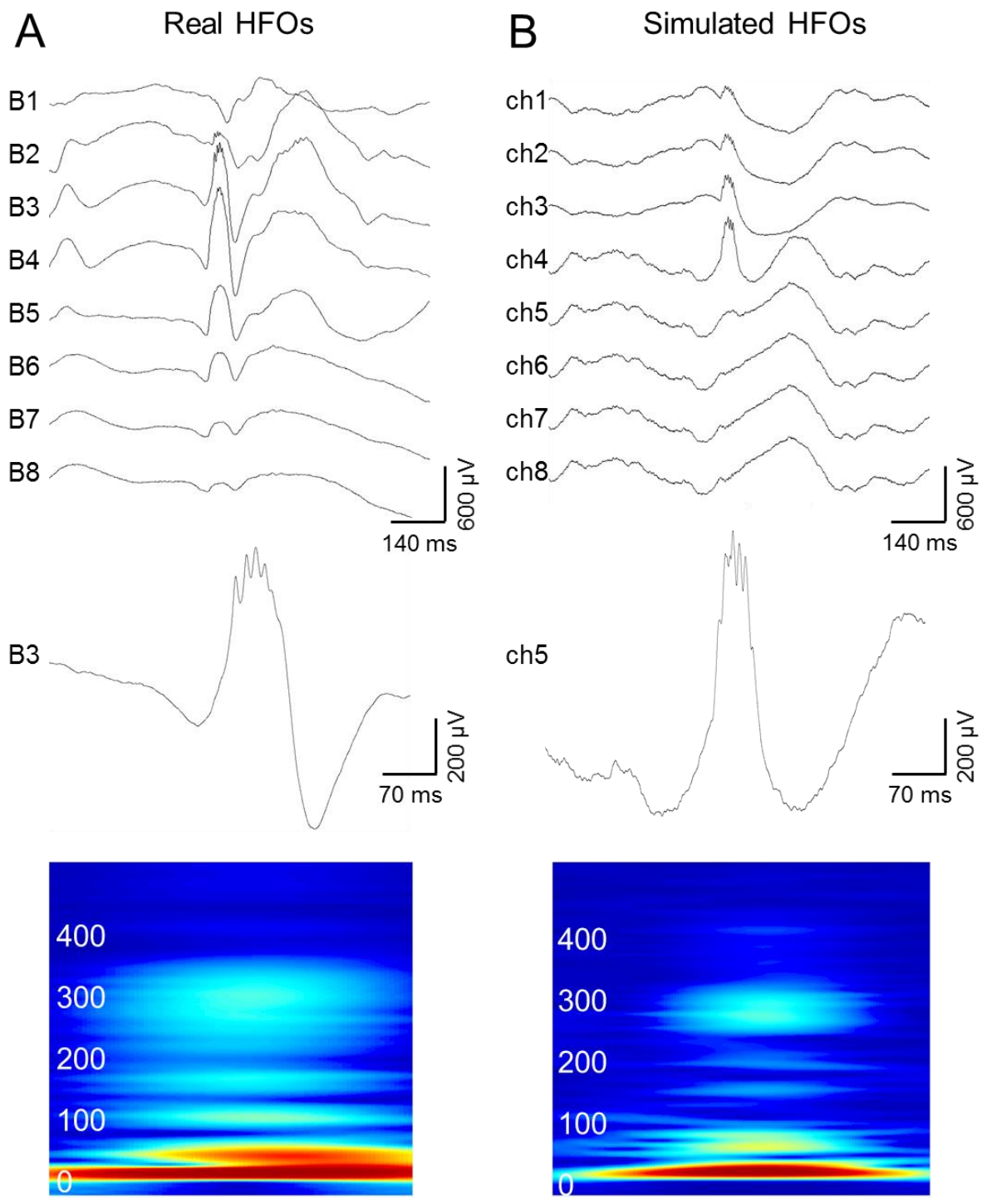

Figure 8. Qualitative comparison of real and simulated HFOs. (A) Real HFOs extracted from intracerebral recordings (electrode B, 8 channels) of a patient suffering from drug resistant epilepsy. One contact is chosen and its corresponding sperctrogram is illustrated underneath. **(B)** Simulated signals on 8 contacts a virtual electrode is depicted. One channel showing clear HFO is chosen (ch5) and its corresponding spectrogram is illustrated underneath.

1.5. Technical Considerations

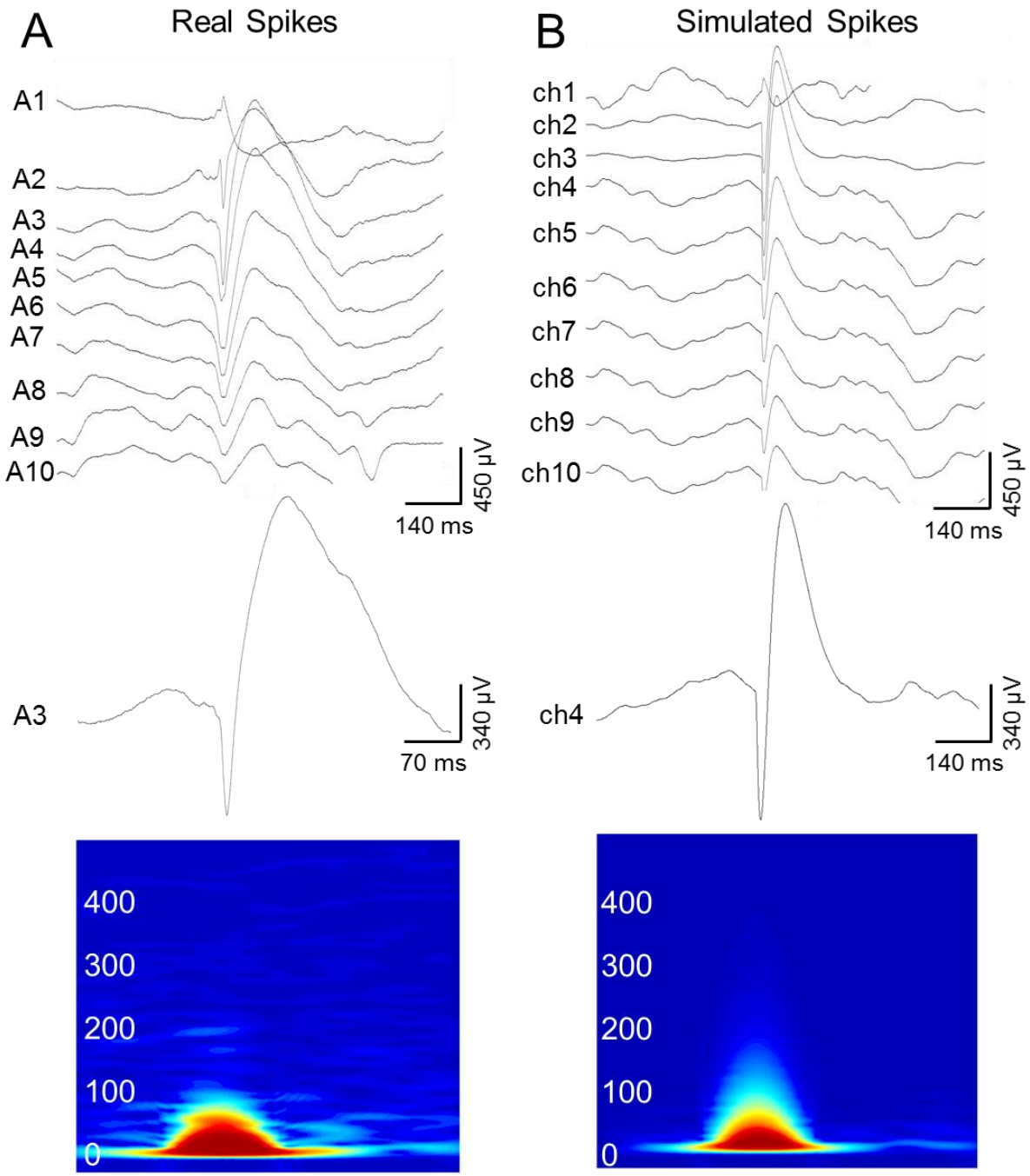

Figure 9. Qualitative comparison of real and simulated spikes. (A) Real spikes extracted from intracerebral recordings (electrode C, 10 channels) of a patient suffering from drug resistant epilepsy. One contact is chosen and its corresponding spectrogram is illustrated underneath. **(B)** Simulated signals on 8 contacts a virtual electrode is depicted. One channel showing clear spike is chosen (ch4) and its corresponding spectrogram is illustrated underneath.

1.5.1. Platform

Triple-S is a 3D slicer module developed using Python programming Language. It has been primarily tested on slicer 4.6.2 version as its core is built upon vtk 6.2 (Visualization Tool Kit (Schroeder et al., 2004)) that provide better graphical rendering, and the new version of QT 4.8.6 which introduce a wide variety of enhanced user interface controls. Using Python v2.7 we exploited the new functionalities within the new version of vtk and introduced new flexible interactive techniques to deal with the 3D scene. In addition, the user interface was designed to comply with the standard requirements for a flexible, reliable and user friendly interface where pyQt application framework was used to facilitate the user's interaction with the software (creating dynamic tables, buttons within tables, combo box within tables, automatic update of data …).

1.5.2. Requirements

Triple-S module is a python extension of the 3D slicer software, so all minimum requirements needed for 3D slicer to function properly are necessary for proper functioning of Triple-S. The module is executable on multiple operating systems (Windows, MAC, and Linux) and demands (1) enough memory (4 GB of RAM is necessary, 8GB is recommended) and (2) dedicated graphics hardware and memory (1GB is recommended) to hold minimum functionalities such as reading original data, visualization, processing algorithms and interface interaction. Furthermore, for the interface devices a three button mouse with a wheel scroll is suggested. It has not escaped our notice that most algorithms implemented in 3D slicer are multithreaded and will benefit from multicore CPU configurations, so we embedded the python *threading* library into the module that speeded up the computation of lead field matrix. In fact, computation time was lower by 6 times on a 24 core CPU than that on a single core. Thus, using a CPU with multiple cores is necessary to reduce the overall processing time.

1.5.3. Availability and Installation

Triple-S is open source module which source-code will be freely available and can be downloaded from a webpage to be designed after obtaining the required license. To add Triple-S to 3D slicer one should follow the following steps:

- Use the "View->Application Settings" dialog option:
- choose "Modules" option
- Press on arrow button (>>) beside the "additional module paths" textbox
- Use option "Add" to add the path of the Triple-S module
- Press "Ok" and restart the Slicer

- Upon restart, you can find Triple-S in the simulation section of the “Modules” combo-box

1.5.4. Input Output Formats

To effectively use Triple-S module, compatible input file formats should be imported. At first, a 3D mesh of the brain cortex should be added through Slicers “Add Data into the scene”, where the user is supposed to choose a VTK file of the cortical mesh which is normally produced by FreeSurfer software (Fischl, 2012) or any other MRI image processing software (Ashburner, 2009; Smith et al., 2004). The MRI data and CT scan are not necessary for Triple-S to function normally, but their presence could greatly facilitate the process of delineating the cortical patch. However, there is no special format of MRI and CT scan files other than what 3D slicer can read.

Regarding the neural population dynamics, the input files should follow a certain format. Triple-S is capable of reading two types of file formats for temporal dynamics, ASCI and binary files (.bin, .dat), in condition that it is accompanied by an appropriate descriptive file. The descriptive file (.des) is an ASCI file common for both binary and text formats, it defines the manner in which the dynamics are saved into the data file. It should contain the following fields each on a separate line:

- [patient] XXXXXXXX //name of patient
- [date] dd/mm/yyyy //date of acquisition
- [time] hh:mm:ss //time of acquisition
- [extractedFom] path //path to raw data if any
- [samplingfreq] f //value of Sampling frequency
- [nbsegments] n //integer, number of segments in current file
- [nbsamples] s //integer, total number of samples
- [segmentsize] z //integer, number of samples of each segment
- [segmentInitialTimes] I //initial time in seconds of each segment
- [nbchannels] N //integer, number of neuronal population
- [channelnames]:

  POP0 -----

  POP1-----

  POP2 -----

The number of the lines on which the name of the channels (populations) are present should be equal to N (nbchannel). Note that in simulations many of the information given by the above listed fields are not necessary (patient name, date of acquisition, time, path to raw data), however their presence even with false information is essential for the uniformity of file format. If binary data file format is used (.bin), the

data should be saved as a 32 bit double precision numbers otherwise a normal text format where numbers are separated by a space character is required (.dat).

Output files exported by Triple-S are of ASCII text file format (.txt) that contain matrices explicitly describing the exported data, for example the position of contacts are exported as a N*3 matrix where each row contain the (x, y, z) coordinates of each contact; samples of output files could be found in supplementary materials SM.