

Manipulations 1 à 14

Simul_neurophysio_V4 et WinXNBC9_11_S

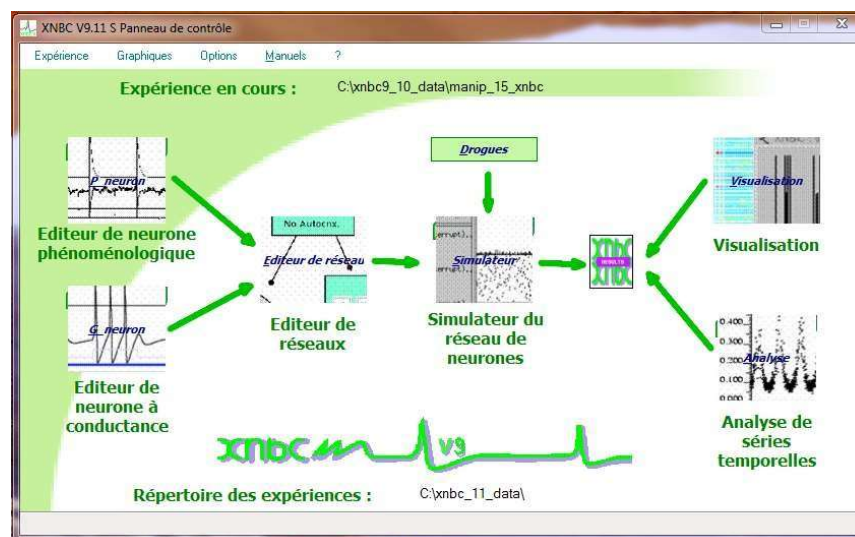
Jean-François Vibert

Faculté de médecine Pierre et Marie Curie, site Saint-Antoine,
Université Sorbonne Université, INSERM IPLESP UMR S 1136

Simul_neurophysio est une application Windows en surcouche à XNBC9 qui permet de lancer directement l'un des 2 simulateurs de neurone (P_neuron ou G_neuron) nécessaires pour réaliser les manipulations 1 à 14. Il permet également de lancer XNBC9 décrit ci-dessous, qui permet de réaliser les manipulations 15 à 30. P_neuron et G_neuron étant des outils de XNBC9n ils sont décrits avec lui.



WinXNBC9_11 est un outil de neurobiologie numérique issu de XNBC9 V10, la version originale sous Linux. Cette version a été légèrement simplifiée (S) pour des raisons pédagogique, mais propose les mêmes outils que XNBC9 original, exceptés un second éditeur de réseau assez complexe et un outil d'analyses fréquentielles.



Téléchargement du simulateur de réseaux de neurones.

Vous pouvez télécharger le programme sur :

<http://livre-neurophysiol.sorbonne-universite.fr/simulateurs/>

Le simulateur **simul_neurophysio_V4**, que vous allez télécharger inclus **WinXNBC9_11_S** qui est une version simplifiée du simulateur de neurones et de réseaux de neurones XNBC V9 utilisé en neurobiologie numérique.

Aussi bien **simul_neurophysio** que **WinXNBC9_11_S** sont une interface disponible en 4 langues (français –par défaut–), anglais, espagnol et italien), mais les outils eux-mêmes, qui sont appelés par ces 2 programmes sont exclusivement en anglais (mais aisément compréhensibles). De même, les manuels complets de **P_neuron** et **G_neuron** et le manuel utilisateur complet de XNBC est celui, très complet de la version précédente (également en anglais) mais il n'y a que des changements cosmétiques entre la version 8.34 (celle du manuel) et la version 9.11 utilisée ici.

Pour ceux qui seraient intéressés, on trouvera les sources et les exécutables Windows et Linux de ce simulateur (XNBC), open source distribué sous la licence GPL, sur le site du simulateur :

<http://xNBC.sorbonne-universite.fr/>

ainsi que sur le site de sourceforge :

<https://sourceforge.net/projects/xNBC/>

Installation du simulateur de réseaux de neurones.

Sur PC sous Windows

Téléchargez [simul_neurophysio_V4.exe](#) selon votre PC. C'est un installeur qui installera le programme complet **simul_neurophysio.exe**. Il demandera dans quel répertoire l'installer. Il est impératif de l'installer dans un répertoire pour lequel vous disposez des droits d'écriture, car il produit de nombreux petits fichiers pour chaque simulation. Votre répertoire **Documents** est un bon choix, dans lequel vous pourrez créer un répertoire **manips**. C'est alors dans ce dernier que vous installerez **simul_neurophysio.exe**. Vous trouverez dans ce répertoire XNBC9_11, et, dans **xNBC9_11_data**, toutes les manipulations 15 à 30 préparamétrées dans les répertoires **xNBC9_11_data/manip_xx_xNBC**, où xx est le numéro de la manipulation. Les manipulations 1 à 14 ont des paramètres communs, et un seul répertoire **manip_1_xNBC** existe. Le répertoire **xNBC9_11_data/neurones** contient des fichiers de paramétrage de neurones de divers types, utilisables par **P_neuron** et **G_neuron**.

Attention, les chemins des fichiers sur PC sont en général présentés comme `/cygwin/C/...` (le C peut changer en fonction du nombre de vos disques) c'est normal car WinXNBC a été porté de Linux à Windows en utilisant le logiciel Cygwin. C'est aussi ce qui explique que les fenêtres de choix de fichier ne se conforment pas au standard Windows : un chemin est présenté en haut qui sert de filtre, et l'on choisit ensuite parmi les chemins filtrés, celui qui convient (éventuellement en remontant au répertoire père (../) et de filtre en filtre on trouve ou donne le nom du fichier voulu.

Sur Mac sous Mac OS X

WinXNBC étant un logiciel développé pour Windows, on peut le faire tourner sur Mac, soit en installant Windows nativement en dual-boot via Bootcamp, mais cela nécessite d'acheter une licence Windows, soit en restant en environnement Mac en utilisant un émulateur Windows, tel que Wine, qui est libre de droits.

Pour utiliser Wine, il faut :

Avoir un Mac à processeur Intel (Mac vendus à partir de 2006)

Avoir installé Xcode (à l'aide du Mac App Store) et ses Command Line Tools.

Une fois Xcode installé, allez dans les Préférences du logiciel puis dans l'onglet Téléchargements et ensuite, installez les Command Line Tools.

Vous devrez ensuite installer HomeBrew, un Package Manager qui vous simplifiera grandement l'installation de Wine.

Ouvrez un Terminal (disponible dans Applications/Utilitaires) et entrez la commande suivante :
`ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`

Ceci va télécharger le nécessaire à l'installation d'Homebrew. Appuyez sur Entrée dès que l'on vous demande quelque chose. Quand on vous demande de taper votre mot de passe admin, faites-le.

Ensuite, le terminal vous dira que tout s'est bien déroulé et vous demandera de lancer *brew doctor*, ce que vous ferez à l'aide de la commande suivante : *brew doctor*

Cette commande va permettre à Homebrew d'inspecter le système afin de voir si tout est bien configuré pour son installation.

Si tout s'est bien déroulé, le message suivant apparaîtra : *Your system is ready to brew*

On peut maintenant installer Wine avec la commande suivante : *brew install wine*

Cette étape est la plus longue (environ 1h, voir plus sur des machines plus anciennes), le Terminal va faire défiler tout un tas d'informations, c'est tout à fait normal. Une fois que l'installation est terminée, on pourra utiliser WinXNBC sur Mac !

On devra donc lancer WinXNBC à partir du terminal (autant le garder dans le Dock !) avec la commande suivante :

`wine <chemin complet de l'endroit où vous avez téléchargé WinXNBC>`

Ce chemin, que vous obtenez en sélectionnant le programme téléchargé, en le copiant (command-C) puis en le collant (command-V) dans le terminal, est du genre :

`/Users/<lenom de votre compte>/Downloads/simul_neurophysio.EXE`

Cela lancera l'installateur le premier coup (voir ci-dessus dans installation sous Windows). Comme sous Windows, il demandera dans quel répertoire l'installer. Il est impératif de l'installer dans un répertoire

pour lequel vous disposez des droits d'écriture, car il produit de nombreux petits fichiers pour chaque simulation. Le plus simple est de l'installer dans

```
/Users/<lenom de votre compte>/Applications/
```

Ensuite, pour lancer simul_neurophysio_V3.EXE, vous ouvrez un terminal et vous lancez la commande suivante :

```
/Users/<lenom de votre compte>/Applications/ simul_neurophysio.EXE
```

Attention, les chemins des fichiers sur Mac sont en général présenté comme /cygwin/Z/... (le Z peut changer en fonction du nombre de vos disques) c'est normal car WinXNBC a été porté de Linux à Windows en utilisant le logiciel Cygwin. C'est aussi ce qui explique que les fenêtres de choix de fichier ne se conforment pas au standard Mac OS X : un chemin est présenté en haut qui sert de filtre, et l'on choisit ensuite parmi les chemins filtrés, celui qui convient (éventuellement en remontant au répertoire père (/..) et de filtre en filtre on trouve ou donne le nom du fichier voulu.

Utilisation du simulateur de réseaux de neurones.

Le simulateur se compose d'un panneau de contrôle permettant de définir globalement la simulation en cours (les simulations préparées sont appelées ici manip_15 à manip_30), et de lancer les différents outils de neurobiologie numérique (ces outils sont exactement les mêmes que ceux de XNBC V9, et c'est pourquoi ils sont en anglais. Seul le panneau de contrôle a été simplifié et mis en français pour ne donner accès qu'aux outils nécessaires dans ce cadre pédagogique). Pour réaliser une simulation, il faut d'abord définir un ou plusieurs types de neurones grâce à un des deux éditeurs de neurone utilisés pour les manipulations 1 à 14 (P_neuron ou G_neuron, mais pour les manipulations 1 à 30, concernant des réseaux vous utiliserez uniquement P_neuron), puis il faut les assembler en réseaux avec l'éditeur de réseau (link_edit) et enfin lancer la simulation avec le simulateur (nbc_x). Au cours de la simulation, on effectuera des actions comme stimuler un ou plusieurs neurones, ajouter du bruit, injecter une drogue, etc. Au cours de cette simulation, on visualise succinctement ce qui se passe globalement dans le réseau. Après la simulation, on utilise le visualiseur (visu) pour voir en détail ce qu'il s'est passé au cours de cette simulation. On pourra également utiliser l'outil d'analyse temporelle (xtms) pour voir graphiquement l'évolution temporelle des fréquences de décharge de neurones.

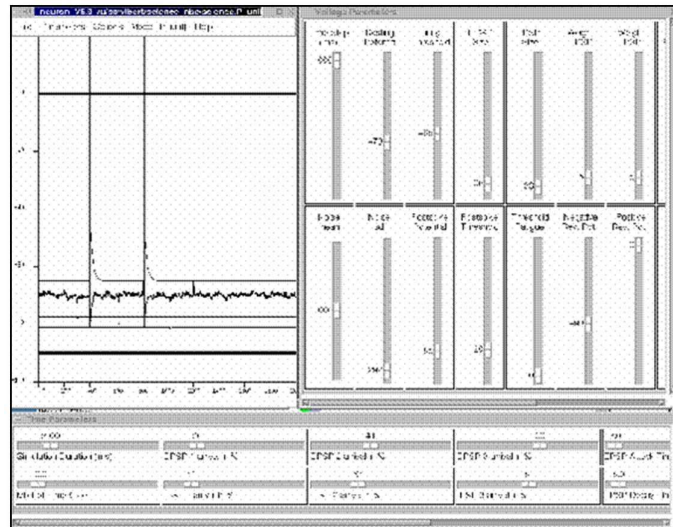
Afin de faciliter la réalisation des différentes manipulations nous avons préparé des fichiers de paramètres qui pré-définissent les caractéristiques des neurones, des réseaux et des paramètres de simulation. Ces fichiers sont enregistrés dans des répertoires portant le nom « manip_XX_xnbc » ou XX est le numéro de la manipulation. Les fichiers de description des neurones, qui sont communs à plusieurs manipulations sont tous enregistrés dans un répertoire « neurones » au même niveau que les répertoires « manip_XX_xnbc ».

Généralités sur l'utilisation des différents outils.

Comme expliqué précédemment, pour les manipulations 1 à 14, concernant le fonctionnement du neurone, seuls seront utilisés les éditeurs de neurone, soit l'éditeur de neurone phénoménologique (P_neuron), soit l'éditeur de neurone à conductance (G_neuron). Pour les manipulations 15 à 30, concernant les réseaux de neurones, on utilisera toujours un neurone phénoménologique et on les assemblera en réseaux. D'une manière générale, pour réaliser ces manipulations 15 à 30, il faut charger

la manipulation à partir du panneau de contrôle à partir du menu « Expérience / Ouvrir », et choisir la manipulation désirée. Ceci chargera le réseau et permettra de lancer directement le simulateur. Néanmoins, il pourra être intéressant de pouvoir modifier le réseau, les poids synaptique ou les connexions entre neurones ou réseaux.

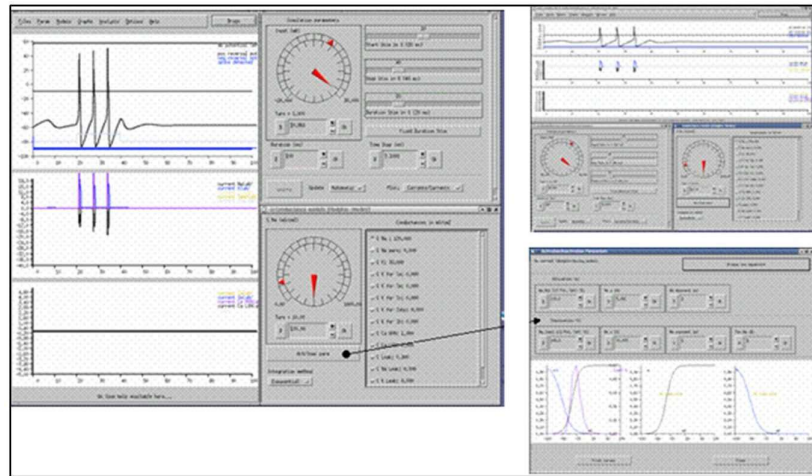
L'éditeur de neurone phénoménologique (P_neuron)



Cet outil permet de choisir les paramètres du neurone à l'aide de curseurs (sur la partie droite et sur le bas de l'écran) avec un contrôle visuel en temps réel dans une fenêtre graphique affichant le potentiel de membrane (sur la partie gauche de l'écran) pendant une durée de simulation réglable par un des curseurs du bas. De nombreux curseurs sont disponibles, mais ne peuvent parfois pas être vus simultanément sur l'écran. Les barres de défilement permettent de déplacer les échelles à droite et à gauche (et de haut en bas) pour atteindre des échelles non visibles.

Trois EPSP et trois IPSP peuvent être positionnés en utilisant les curseurs pour les déplacés dynamiquement. Les poids synaptiques des EPSP et des IPSP sont réglables séparément (mais est le même pour les trois de chaque sorte). Ils permettent de faire décharger le neurone et d'étudier ce qui arrive lorsque les variations de poids (seuil de décharge, EPSP, IPSP, potentiel de réversion, etc.) Les équations utilisées sont fournies dans le manuel disponible à partir du menu « Manuels », ainsi que les valeurs par défaut.

L'éditeur de neurone à conductance (G_neuron)



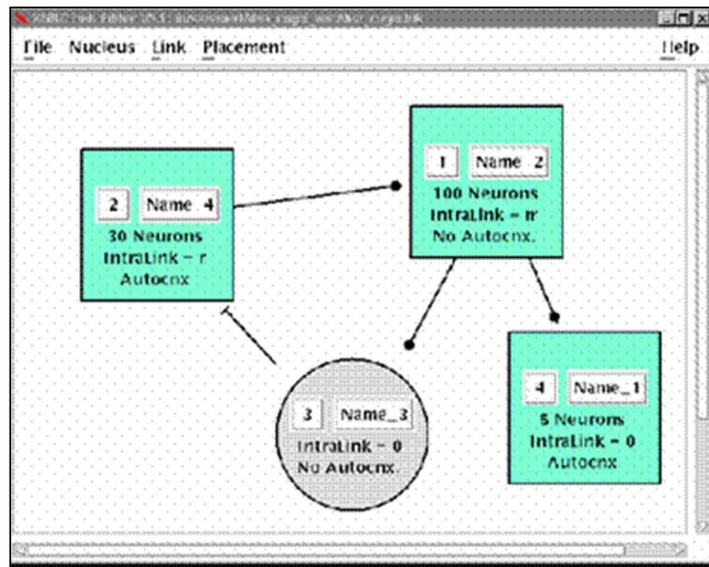
Cet outil permet de choisir les paramètres du neurone de simuler un seul neurone qui peut recevoir des connexions d'autres neurones à des moments choisis et avec un poids synaptique choisi. G est ici pour rappeler le symbole de la conductance électrique, puisque le modèle met en œuvre les courants transmembranaires en utilisant le formalisme de Hodgkin-Huxley. L'utilisateur agit sur le comportement du modèle en réglant interactivement de la conductance en cours (G).

G_neuron permet de simuler un modèle de conductance (GAC) de neurones sur la base de Hodgkin-Huxley (HH). Le modèle dit CBM comprend 14 différents courants transmembranaires. Ce modèle prend explicitement en compte la Na⁺, K⁺, Ca⁺⁺ et Mg⁺⁺ concentrations d'ions. Les courants suivants sont mis en œuvre dans le MBC: I_{na}, I_{na} persistante, I_{ca}, I_{il}, I_{ik}, I_{im}, I_{ia}, I_{iahp}, I_{ic}, I_{ih}, I_{inmda} et I_{leak}.. Les courants sont modélisés en utilisant un type HH simplifiée de la cinétique et le type Michaelis-Menten de la cinétique d'inactivation chimique variables activation / inactivation. L'éditeur de neurones est utilisé pour ajuster les paramètres, y compris tous les paramètres cinétiques d'ions. Le bruit peut être ajouté au potentiel de membrane. Une fois les paramètres sont réglés, ils peuvent être sauvegardés pour un usage ultérieur.

Trois méthodes d'intégration sont proposées: exponentielle, Euler et d'ordre 4 Runge-Kutta. Tous les paramètres peuvent être réglés individuellement. L'interface utilisateur de G_neuron est conviviale. Les paramètres sont ajustés en déplaçant les potentiomètres à double cadran (ou en tapant la valeur), tandis que les évolutions temporelles du potentiel de membrane et de courant ionique (partie gauche de l'écran) montrent le changement en temps réel selon les valeurs des paramètres mis à jour. Un sous-menu graphique permet de dessiner n'importe quelle variable par rapport à toute autre variable. Les expériences de voltage et de courant imposé (voltage et current clamp) peuvent être simulées afin d'ajuster les valeurs de conductance.

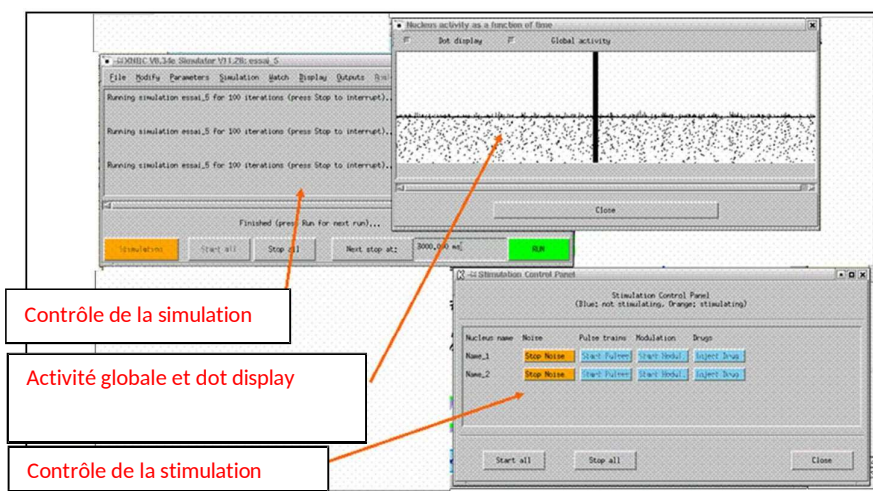
L'effet des drogues telles que TTX et TEA peuvent être simulés pour comparer le comportement du modèle avec et sans les canaux correspondants bloqués. Le potentiel post-synaptique est modélisé en utilisant les fonctions alpha avec temps de montée et constante de désintégration et de l'amplitude comme l'un des deux courants synaptiques (I_{syn_epsp} et I_{syn_ipsp}). Les synapses avec la libération de glutamate peuvent induire une neuromodulation NMDA avec une modification de la membrane de longue durée.

L'éditeur de réseaux (link_edit)



Cet outil permet de créer un réseau de neurones. Les réseaux ou « Nucleus » sont créés à partir du menu du même nom (« Nucleus/New »). Les neurones constituant le réseau doivent être définis à partir de leur fichier de description selon qu'ils sont phénoménologiques (créés avec P_neuron et d'extension .P_unit) ou à conductance (créés avec G_neuron et d'extension .G_unit). Leur nombre est défini dans le même panneau. Un neurone unique est représenté par un cercle, et un réseau par un carré. Dans les deux il y a un nom un numéro et le nombre de neurones pour les carrés, ainsi que la connectivité dans le réseau. Les connexions entre neurones sont réalisées avec le menu « Link / New » puis en cliquant sur le numéro du réseau de départ puis sur le numéro du réseau cible. La fenêtre qui s'ouvre permet de définir la connectivité (excitatrice ou inhibitrice). La connectivité intra réseau se définit en cliquant 2 fois sur le même réseau.

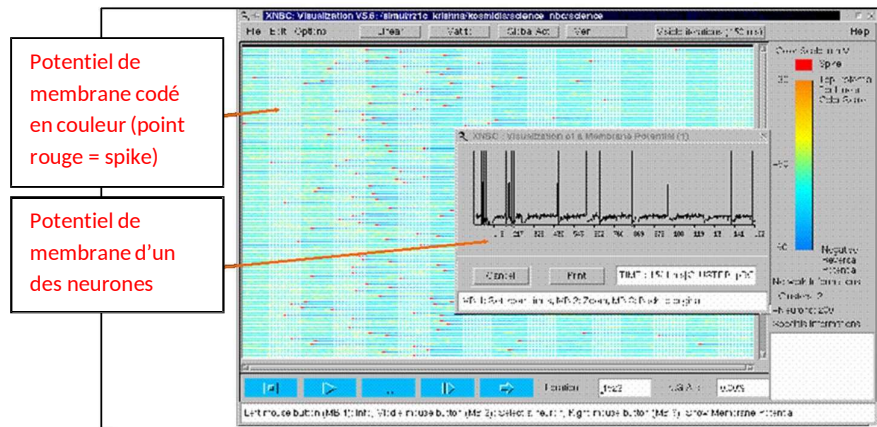
Le simulateur (nbc_x)



C'est le simulateur qu'il faudra lancer après avoir chargé une simulation. Normalement, le réseau est chargé, ainsi que les paramètres de stimulation. Avant de lancer la simulation avec le bouton vert

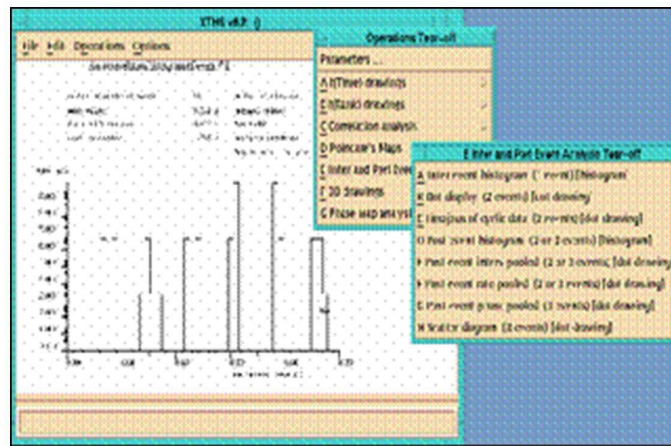
« Run », il convient d'ouvrir la fenêtre permettant de contrôler les stimulations avec le bouton bleu « Stimulation » et de placer cette fenêtre et la fenêtre permettant de suivre le comportement du réseau de manière à ne rien masquer. On peut ensuite lancer la simulation en pressant el bout vert « Run ». Par défaut cela lance 100 ms de simulation. On continue ensuite la simulation en pressant le bouton vert autant de fois que nécessaire, éventuellement en lançant des stimuli à parti de la fenêtre « Stimulation » en agissant sur les boutons bleus « start » ou oranges « stop », comme indiqué dans le descriptif des manipulations. Avant de terminer, il faut sauver le résultat de cette simulation à partir du menu « File/ Save files and exit simulation ». En cas d'oubli un rappel est fourni.

L'outil de visualisation (visu_nbc)



C'est l'outil qui permet de visualiser de plusieurs manières le résultat de la simulation. Il se manipule comme un magnétoscope. Il lit les fichiers sauvegardés à la fin de la simulation et les présente sous diverses formes. La visualisation dite « linéaire » (« linear ») est celle qui est lancée par défaut. Elle présente de haut en bas le potentiel de membrane des neurones sous forme colorée dont l'échelle est présente à droite. La ligne noire sépare les différents réseaux. En abscisse on a le temps écoulé. Il est possible d'augmenter la plage temporelle visualisée en actionnant le bouton en haut à droite. En cliquant avec le bouton droit de la souris sur une des trace colorées on obtient la visualisation du potentiel de membrane du neurone correspondant jusqu'au temps où on a cliqué. Les traits rouges représentent le moment de survenue des potentiels d'actions. Il est possible de zoomer en indiquant les limites basse et haute de la zone à agrandir avec le bouton gauche de la souris, et de zoomer en cliquant avec le bouton du milieu entre les 2 traits verticaux qui viennent d'être posés. Le bouton droit ramène à l'original. La visualisation dite « matricielle » (« matrix ») montre sur la diagonale les neurones avec leurs connexions (axones) vers les autres neurones représentées dans le sens des aiguilles d'une montre. La couleur des neurones correspond à la valeur de leur potentiel de membrane. Les potentiels d'action sont représentés par des petits traits rouges qui se déplacent sur les axones dans le sens des aiguilles d'une montre. La représentation de l'activité globale (« Global Act. ») montre le nombre de neurones actifs (émettant un potentiel d'action) à un moment donné, c'est en pratique la même représentation que celle qui est visualisée lors de la simulation.

L'outil d'analyse des séries temporelles (xtms)



C'est l'outil qui permet une représentation graphique des séries temporelles. Un train de potentiels d'action est en fait une suite d'événements survenant à des dates qui se suivent et qui portent une information dans la manière dont elles se suivent. Si plusieurs trains de potentiels d'actions coexistent il est possible d'étudier leurs interrelations. Ces successions de dates s'appellent des séries temporelles. Lorsqu'il est lancé, xtms propose l'analyse de la simulation qui vient d'être réalisée et il suffit de presser le bouton « OK » pour obtenir la fenêtre où seront affichés les graphiques selon l'analyse choisie avec le menu « Operations ». Dans les manipulations proposées dans le cadre du livre, ne seront utilisées que l'analyse de la fréquence (« f(time) drawings ») et des histogrammes post ou péri stimulus (« Inter and Peri Event Analysis »).

Description des manipulations

Cette section explique les « Manipulations » indiquées dans les chapitres 2 et 4 du livre de neurophysiologie. Il s'agit d'expériences simulées, qui sont proposées pour permettre une compréhension dynamique des phénomènes expliqués. Ces manipulations seront réalisées en utilisant des simulateurs informatiques téléchargeables sur le site de l'éditeur comme expliqué dans le livre. Il s'agit d'outils complets de neurobiologie numérique, libres, développés pour la recherche et utilisés ici pour des exercices pédagogiques, car qui peut le plus peut le moins. Des versions Windows et Linux sont disponibles. Pour Mac OSX la version Windows doit être utilisée via un émulateur Windows standard sur cet OS.

Manipulations 1 à 14

Manipulation 1 : Influence des concentrations ioniques interne et externes

Lancer le simulateur de neurones à conductances G_{neuron}

En utilisant le simulateur G_{neuron} , on peut voir l'influence des concentrations ioniques interne et externes sur la valeur de E_K (et de E_{Na}) dans le panneau accessible du menu [Param/Equil. pot & sp. det...]. Pour chaque zone d'entrée de valeur, un bouton D permet de revenir à la valeur « normale », par défaut, les boutons + et - permettent d'augmenter ou diminuer la valeur d'abord lentement puis de plus en plus vite.

Il est conseillé de se mettre en mode manuel (Manual) de mise à jour (Update) des graphiques avec le bouton en bas de la fenêtre droite du haut. La mise à jour des graphiques se fait alors en pressant le bouton [Update].

Avec les valeurs par défaut (réglées pour un neurone typique), le changement des concentrations de K et Na intra et extracellulaires modifie le potentiel de repos. Pour voir des modifications faites varier ces concentrations avec les d'entrée de valeur des seconde et troisième lignes. Vous constatez alors, d'une part les potentiels d'équilibre (en Bleu pour K et rouge pour Na) qui varient, et d'autre part que le potentiel de repos lui-même ne varie pratiquement pas si on change Na, mais varie beaucoup si on change K. On peut donc constater que ce sont surtout les modifications de K extra ou intra cellulaires ($[K]_{ext}$ et $[K]_{int}$) qui modifient le potentiel de membrane. Les modifications de Na ($[Na]_{ext}$ et $[Na]_{int}$) le changent peu.

Manipulation 2 : Influence des variations de conductance ionique sur la valeur du potentiel de repos.

En utilisant le simulateur G_neuron, on peut voir l'influence des variations de conductance ionique sur la valeur du potentiel de repos. On partira en mettant toutes les conductances ioniques à zéro. Pour ce faire, dans le panneau Conductance, on cliquera successivement sur toutes les coches dont la conductance par défaut n'est pas nulle (GNa, GK, GCaHVA et Gleak) et pour chaque on met 0 comme valeur, soit avec le cadran soit en tapant 0 dans la zone de texte.

On active ensuite la conductance au sodium (GNa) et on l'augmente : on voit le potentiel augmenter et se rapprocher du potentiel d'équilibre ENa (dépolariation).

On remet GNa à zéro.

On active ensuite la conductance au potassium (GK) et on l'augmente : on voit le potentiel diminuer et se rapprocher du potentiel d'équilibre EK (hyperpolarisation).

Manipulation 3 : Influence de multiples synapses qui arrivent sur le neurone

En utilisant le simulateur G_neuron, on peut voir l'influence de multiples synapses qui arrivent sur le neurone

Remettons les valeurs de conductance à 0.

Dans le panneau accessible du menu Param/Noise & clamping Noise sd (déviatoin standard du bruit), autrement intensité du bruit), on mettra une valeur 0.5, 1, 2, ce qui ajoute sur la membrane du bruit synaptique. En pratique, la membrane n'est jamais au repos, et se présente plutôt tel que vous le voyez là.

Si vous remettez les valeurs de conductance à leurs valeurs par défaut (celles observées sur le neurone typique), et que l'on mette du bruit, on voit que des accidents pointus (des potentiels d'action ou spikes) apparaissent. Vous comprendrez pourquoi un peu plus loin.

Manipulation 4 : Influence d'une seule synapse excitatrice ou inhibitrice qui arrive sur le neurone.

En utilisant le simulateur G_neuron, on peut voir l'influence d'une seule synapse excitatrice ou inhibitrice qui arrive sur le neurone.

Restez avec les valeurs de conductance à leurs valeurs par défaut (celles observées sur le neurone typique), et remettre le bruit SD à 0,

Dans le panneau accessible à partir du menu PSP & experiment, mettre 1 bouton synaptique, et valider avec [OK]. Dans le panneau qui apparaît, déplacer le temps d'arrivée avec le curseur vers 50 ms (50% du graphe), puis sélectionnez ([neurtransmitor]) l'Acétylcholine comme neurotransmetteur excitateur. Ensuite augmentez progressivement le poids synaptique (PSP weight) avec le [+] et validez avec [OK] ? Constatez l'apparition d' epsp, puis à partir d'une certaine valeur, d'un potentiel d'action. On simule la l'arrivée d'un potentiel d'action sur la membrane du neurone via un transmetteur chimique excitateur. Quand vous avez vu le spike, changez de neurotransmetteur et mettez du GABA. Constatez l'IPSP.

Manipulation 5 : Influence d'une impulsion électrique qui arrive sur le membrane du neurone.

En utilisant le simulateur G_neuron, on peut voir l'influence d'une impulsion électrique qui arrive sur la membrane du neurone.

Restez avec les valeurs de conductance à leurs valeurs par défaut (celles observées sur la vraie cellule), et remettre le bruit SD à 0,

Dans le panneau de droite en haut, (Simulation parameters) on va régler la durée de la stimulation électrique à 1ms, puis la fixer (bouton au-dessous de la durée), puis on va démarrer la stimulation (start stim) vers 50 ms (on pourra la déplacer).

Avec le cadran on augmente alors la stimulation électrique et l'on voit apparaître un epsp d'abord, puis un spike lorsque la stimulation atteint un certain niveau (seuil de décharge).

Manipulation 6 : Notion de seuil

En utilisant le simulateur P_neuron, qui est plus simple que G_neuron on peut mieux voir l'influence d'une seule synapse excitatrice ou inhibitrice qui arrive sur le neurone.

On manipule P_neuron uniquement via des curseurs pour changer les valeurs du modèle.

On laisse les valeurs par défaut. On supprime le bruit synaptique s'il y en a (Noise sd à 0).

On voit alors 3 epsp, qui sont mis par défaut au démarrage. Si on augmente la taille des epsp (curseur vertical epsp size à droite, rangée du haut), on arrive à dépasser le seuil de décharge et on obtient un potentiel d'action. On constate que le seuil augment puis redescend exponentiellement à sa valeur initiale. Cela traduit la période réfractaire relative (voir plus bas).

Ajoutez du bruit (curseur vertical Noise sd) et constatez ce qui se passe. Notez que le bruit est plus faible au début de la période réfractaire. Cela traduit le shunt de la membrane, autrement dit, la

membrane étant en période réfractaire, elle est moins facilement excitable, car les différents canaux sont ouverts).

Manipulation 7 : Sommations spatiale et temporelle

En utilisant le simulateur P_neuron, on laisse les valeurs par défaut. On supprime le bruit synaptique s'il y en a (Noise sd à 0).

On peut avoir un spike avec des potentiels petits (infra liminaires) si ils arrivent en même temps (sommation spatiale). Pour comprendre, mettre 2 ou 3 epsp infraliminaires au même endroit avec les curseurs horizontaux et un potentiel d'action apparaît. Si l'on déplace les un des curseurs, on constate que le potentiel d'action va rester même si le temps d'arrivée des epsp n'est pas totalement synchrone. C'est la sommation temporelle.

Si l'on ajoute un ou plusieurs ipsp (en augmentant la valeur du curseur vertical IPSP size. Et en les déplaçant, on observe la sommation temporelle qui est une sommation algébrique des epsp et ipsp qui va faire apparaître ou disparaître un potentiel d'action en fonction des moments d'arrivée des epsp et de leur taille. On constate qu'après le potentiel d'action il existe une hyperpolarisation, qui contribue également à la période réfractaire relative.

Fermer le simulateur P_neuron (Menu Files/Quit).

Manipulation 8 : Effet d'une augmentation du potentiel de membrane

En utilisant le simulateur G_neuron (on laisse les valeurs par défaut), on va augmenter le potentiel de membrane, en injectant un courant continu en agissant sur le potentiomètre.

On constate que pour une certaine valeur de voltage injecté, on obtient un potentiel d'action immédiatement maximal, et que si l'on augmente plus le voltage, on ne change pas la taille des potentiels d'action, mais seulement leur fréquence.

Fermer le simulateur G_neuron (Menu Files/Quit).

Manipulation 9 : Période réfractaire relative et période réfractaire absolue

En utilisant le simulateur P_neuron, on laisse les valeurs par défaut.

On fait déclencher un potentiel d'action en ajoutant du bruit continu (curseur Noise mean). On met à 100% l'EPSP 3 arrival time (curseur horizontal). On positionne l'EPSP 2 peu après le potentiel d'action. On s'aperçoit que pour obtenir un nouveau potentiel d'action, il faut augmenter beaucoup la taille de l'EPSP (curseur EPSP Size) et que plus on s'éloigne du potentiel d'action, plus il est facile d'en déclencher un nouveau : c'est la période réfractaire relative, qui est due à l'augmentation du seuil.

Si on positionne l'EPSP2 au même moment que le potentiel d'action, on n'obtient pas de nouveau potentiel d'action, même pour une valeur très élevée de l'EPSP. C'est la période réfractaire absolue.

Fermer le simulateur P_neuron (Menu Files/Quit).

Manipulation 10 : Expérience de voltage imposé (voltage clamp)

En utilisant le simulateur G_neuron (on laisse les valeurs par défaut), on va regarder ce qu'il faut envoyer comme courant au travers de la membrane pour imposer un potentiel de membrane constant.

Dans le panneau accessible à partir du menu Param/PSP & experiment, on coche la case Voltage Clamp, et on ferme le panneau. Dans le panneau accessible du menu Param/Noise & clamping, , on met la valeur de Holding potential à -50 mV (cela fixe la valeur de base du potentiel de membrane) et la valeur de Command pote. A -20 mV (cela donne la valeur à maintenir constante au-dessus du potentiel de base).

On constate que les valeurs de I_{Na} et I_K se modifient. Si on dessinait la somme, on constaterait que la courbe a la forme de la courbe verte de la figure 2.10.

Là on n'a pas de potentiel d'action car on modifie le courant injecté pour maintenir le potentiel de membrane constant. Faites varier les différentes valeurs de potentiel de commande et de conductance Na et K . Regardez ce qu'il se passe.

Décocher ensuite la case Voltage Clmp et cocher la case Current Clamp (fonctionnement normal).

Manipulation 11 : Effet du blocage de canaux ioniques

En utilisant le simulateur G_neuron, on peut voir l'influence séparée des canaux Na et K en bloquant sélectivement le canal sodique par du TTX et potassique par du TEA.

On simule l'injection de ces drogues en les activant à partir du menu Drugs.

Regarder l'effet de chaque drogue, et des deux simultanément en les mettant à Yes ou No

Manipulation 12 : Le potentiel d'inversion

En utilisant le simulateur G_neuron, on peut voir l'influence du potentiel d'équilibre d'un ion sur l'effet d'une synapse en fonction de la valeur du potentiel de membrane.

Restez avec les valeurs de conductance à leurs valeurs par défaut (celles observées sur le neurone typique), et remettre le bruit SD à 0,

Dans le panneau accessible à partir du menu PSP & experiment, mettre 1 bouton synaptique, et valider avec [OK]. Dans le panneau qui apparaît, déplacer le temps d'arrivée avec le curseur vers 50 ms (50% du graphe), puis sélectionnez ([neurtransmiter]) l'Acétylcholine comme neurotransmetteur excitateur. Ensuite augmentez progressivement le poids synaptique (PSP weight) avec le [+] et validez avec [OK] ? Constatez l'apparition d'epsp. On simule l'arrivée d'un potentiel d'action sur la membrane du neurone via un transmetteur chimique excitateur. Changez de neurotransmetteur et mettez du GABA. Constatez l'IPSP.

Changez maintenant le potentiel d'équilibre de l'ion (potentiel dit de réversion) et observez que selon la valeur du potentiel de membrane, la synapse devient excitatrice ou inhibitrice. Ce qui compte est la valeur du potentiel de membrane, relativement au potentiel d'équilibre de l'ion.

Manipulation 13 : La forme du potentiel d'action

En utilisant le simulateur G_neuron, on peut voir l'influence des différents paramètres de l'équation de Hodgkin-Huxley sur la forme du potentiel d'action.

Dans le panneau de droite du bas (Conductance models) en activant le bouton Act/Inact params, on ouvre un panneau permettant de régler les paramètres d'activation (m) et d'inactivation (h) des canaux de chaque conductance (préalablement sélectionne dans la liste des conductances de droite).

Manipulation 14 : La fréquence du potentiel d'action

En utilisant le simulateur G_neuron (on laisse les valeurs par défaut), on va augmenter le potentiel de membrane, en injectant un courant continu en agissant sur le potentiomètre.

On constate que pour une certaine valeur de voltage injecté, on obtient un potentiel d'action immédiatement maximal, et que si l'on augmente plus le voltage, on ne change pas la taille des potentiels d'action, mais seulement leur fréquence.

Dans le panneau accessible du menu Param/Noise & clamping Noise sd (déviation standard du bruit), autrement dit intensité du bruit), on mettra une valeur 0.5, 1, 2, ce qui ajoute sur la membrane du bruit synaptique. En pratique, la membrane n'est jamais au repos, et se présente plutôt tel que vous le voyez là. Regardez ce qu'il se passe.

Fermer le simulateur G_neuron (Menu Files/Quit).