Large-scale neural modeling (LSNM) Upgrade: Final Report

Antonio Ulloa
Neural Bytes LLC
July 21, 2014

Introduction

The large-scale neural modeling (LSNM) software is a set of programs written in C++, Unix shell script, and Matlab that simulate brain activity during auditory and visual working memory experiments. The software was originally developed by Malle Tagamets and Barry Horwitz in the mid 1990's to simulate neuroelectrical activity and PET signal during a visual working memory experiment (Tagamets and Horwitz, 1998), and later modified by Fatima Husain to simulate neuroelectrical activity and fMRI signal during an auditory working memory experiment (Husain et al, 2004). The system was significantly expanded by Theresa Long and Barry Horwitz by adding simulations of cross-hemispheric connections, different attention levels, weakened neural connectivity strength, sick neuronal populations (as in Alzheimer's disease) and multi-trial, multi-subject simulations; and by Brent Warner and Barry Horwitz, by adding functional connectivity simulations (Horwitz et al, 2005). There are several other variations of the programs described above, such as simulations of: Transcranial magnetic stimulation (TMS) (Husain et al, 2002), working memory of auditory sequences (Ulloa et al, 2008), auditory perceptual grouping (Husain et al, 2005), decision-making (Wen et al, 2008), structural equation modeling (SEM) analysis of simulated fMRI timeseries (Kim and Horwitz, 2009), and Magnetoencephalography (MEG, Banerjee et al, 2012). Different versions of the code and documentation for the simulations described above ended up being physically located in different storage media, different servers and directories, under different names, and with a loosely defined directory structure.

As summarized above, the LSNM software had significantly grown in size over the years, and therefore needed to be streamlined to make the code easier to understand and to modify. Specifically, the issues that were addressed within the current project were: *(1) LSNM lacked a systematic and centralized documentation system to guide users in the implementation of new brain models; (2) LSNM lacked a version control system to keep track of software and documentation updates, and (3) The C++ code and shell scripts contained hardcoded references to specific usernames and local directory locations, thus making the software's portability a difficult and time-consuming process*. Below I describe in detail how I solved those three issues.

(1) Implementation of a systematic and centralized documentation system to guide users in the implementation of new brain models

A “wiki” style of documentation was implemented using the open-source software Tiddlywiki. This wiki page is contained in a single file called “LSNM_Documentation.html”, located within the “Documents” directory, and it current and historical documentation and links that are relevant to LSNM. “LSNM_Documentation.html” is a HTML/Javascript file that is easy to browse and modify using any modern browser (e.g., firefox, safari, explorer), thus making the documentation a "living document" in the sense that it is in a constant state of evolution. The documentation file contains links to both published research manuscripts available online and local file links available in the local “Documents” directory.

Figure (1) shows a snapshot of what the LSNM documentation page looks like. The title of the documentation is shown at the top as “LSNM 2.0 Documentation.” Within the body of the document, on the left-hand side, there is a menu that contains a logically arranged list of entries that new users can consult in order to get acquainted with the software. As an example, figure (1) shows the “Introduction” entry, which contains a summary of the historical background of LSNM, as well as a timeline of its development. We could very easily update this “Introduction” by clicking on the “edit” link and then modifying it as needed. On the right-hand side of the page we have a “search” box that allows us to perform an automated keyword search, a “new tiddler” option to add new entries to the documentation system, and a “username” field that is used to sign the edits made. On the bottom section of the right-hand side area, there is a list of all the entries made to the documentation, which can be visualized either in alphabetical order or in chronological order. It is also possible to assign “tags” to the entries in the documentation system, as needed, to keep entries organized. For a more detailed reference guide to Tiddlywiki, please refer to its online guide at http://classic.tiddlywiki.com.

(2) Implementation of a version control system to keep track of software and documentation updates

A version control system was implemented using "Git", a widely-available version control system tool, to keep track of documents and programs contained in the LSNM system. Git is a standard open-source version control system that is included in the basic linux distribution. Git can be used by issuing commands directly in a shell terminal or by making use of two Graphic User Interfaces (GUIs), called "gitk" (see Figure 2 for a snapshot of gitk) and "git gui". All of the files, directories, and subdirectories contained in "lsnm" are being tracked by Git, and there is a master copy of the "lsnm" directory in Helix (helix.nih.gov, the primary login machine for the NIH Helix Systems), located in /home/horwitzb/bhguest/sgihome and called LSNM.git. Note that in order to have access to the files contained within LSNM.git, one must first clone a copy of the LSNM in one's local directory. In order to incorporate the changes made locally into the LSNM repository, one needs to “commit” those changes and then “push” those changes to the server. Please refer to Appendix A for a list of the most frequently used “Git” commands for the purposes of managing the LSNM system. For a more detailed description of Git, as well as more information on how to perform sophisticated tasks with it, please refer to Scott Chacon’s Git user guide (Chacon, 2009), also available online at http://git-scm.com/book.

(3) Removal of hardcoded references to usernames and local directory locations to make the software portable

All the references to specific user names and directory names, other than LSNM’s standard directory structure, were removed and the code was recompiled. By doing this, the software’s portability was increased, as the dependency on physical location was eliminated. In order for LSNM to know what its physical location is, I introduced a unix environment variable called "LSNM", which needs to be added to the user's ".bash_profile", located in the user's home directory. The following two lines need to be added to the file ".bash_profile":

```
LSNM=$HOME/location/of/LSNM/files

export LSNM
```

The first time that the LSNM environment variable is set up (right after copying the LSNM source code to a new user’s home directory, for example), it is necessary to force linux to upload the environment variable LSNM by entering the following command at a shell terminal:

```
source .bash_profile
```

which only needs to be entered once because subsequent times linux will read the .bash_profile everytime a user logs in to their unix account.

Figure (3) shows LSNM’s current directory structure (please note that the Matlab directory is not depicted in this chart). The directory structure of the LSNM software is shown below. The box labeled "**$LSNM**" corresponds to the root directory where the software is located. The "batch files", which are in charge of executing simulations across different subjects, scans, trials, and attention levels, are located at the "**$LSNM**" directory level. Also contained at the "**$LSNM**" directory level are the "*.s" files that are used as the input timeline for the simulation. Additionnaly, the "**$LSNM**" directory level contains the output subdirectory (specified in "code/mkbatch*.in") of the batchfiles. The "**bin**" directory contains all the executable files, which have been compiled using the C++ code contained in the directory "**code**". The "**code**" directory contains C++ programs that generate the batch files contained at the "**$LSNM**" level, as well as "*.in" files that in turn contain input variables to make the batch files. The "**code**" directory also contains the "**netgen**" and "**sim**" directories, which in turn contain the connection weight generation (within a hemisphere) code and simulation execution code, respectively. The "**netgen**" and "**sim**" directories also contain the makefiles needed to compile the source code contained in them. The subdirectory **"inputs**" cointans "*.inp" and "*.rsb" files. "*.inp" files contain input stimuli and "*.rsb" files contain a sequence of events in one trial. The subdirectory "**sfiles**" contains files that vary attention levels and that reset the system. Finally, the subdirectory "**weights**" contains the connection weights among the brain regions modeled.

Conclusions and future work

In conclusion, the following upgrades have been made to LSNM: *(1) Implementation of a systematic and centralized documentation system to guide users in the implementation of new brain models; (2) Implementation of a version control system to keep track of software and documentation updates, and (3) Removal of hardcoded references to specific usernames and local directory locations to make LSNM portable*. The introduction of the above upgrades facilitates the usability and further improvement of the LSNM software.

Future work to be done on the system includes the simplification of “netgen”, the network connection weights generator, so that a user can easily add brain regions to the model without having to directly modify C++ code. Another proposed direction is to unify the auditory and visual simulation directories, as most of the C++ code and shell scripts contained in those two directories is duplicated.

References

Banerjee A, Pillai AS, Horwitz B (2012). Using large-scale neural models to interpret connectivity measures of cortico-cortical dynamics at millisecond temporal resolution. Front Syst Neurosci. Jan 6;5:102.

Chacon S (2009). Pro Git. Everything you need to know about the Git distributed source control tool. Apress.

Horwitz B, Warner B, Fitzer J, Tagamets MA, Husain FT, Long TW (2005). Investigating the neural basis for functional and effective connectivity. Application to fMRI. Philos Trans R Soc Lond B Biol Sci. May 29;360(1457):1093-108.

Husain FT, Lozito TP, Ulloa A, Horwitz B (2005). Investigating the neural basis of the auditory continuity illusion. J Cogn Neurosci. Aug;17(8):1275-92.

Husain FT, Nandipati G, Braun AR, Cohen LG, Tagamets MA, Horwitz B. (2002). Simulating transcranial magnetic stimulation during PET with a large-scale neural network model of the prefrontal cortex and the visual system. Neuroimage. Jan;15(1):58-73.

Husain FT, Tagamets MA, Fromm SJ, Braun AR, Horwitz B. (2004). Relating neuronal dynamics for auditory object processing to neuroimaging activity: a computational modeling and an fMRI study. Neuroimage. Apr;21(4):1701-20.

Kim J, Horwitz B. (2009). How well does structural equation modeling reveal abnormal brain anatomical connections? An fMRI simulation study. Neuroimage. May;45(4):1190-9.

Tagamets MA, Horwitz B. (1998). Integrating electrophysiological and anatomical experimental data to create a large-scale model that simulates a delayed match-to-sample human brain imaging study. Creb Cortex. Jun;8(4):310-20.

Ulloa A, Husain FT, Kemeny S, Xu J, Braun AR, Horwitz B (2008). Neural mechanisms of auditory discrimination of long-duration tonal patterns: a neural modeling and fMRI study. J Integr Neurosci. Dec;7(4):501-27.

Wen S, Ulloa A, Husain FT, Horwitz B, Contreras-Vidal JL (2008). Simulated neural dynamics of decision-making in an auditory delayed match-to-sample task. Biol Cybern. Jul;99(1):15-27.

Appendix A. Most frequently used Git commands for the purposes of maintaining the LSNM system.

The following basic commands are useful to keep track of changes to the LSNM software:

How to show the status of changes to the code:

```
$ git status
```

How to commit a file that has been modified or recently created (it skips the 'staging' area):

```
$ git commit -a -m 'include a relevant comment here'
```

How to rename a file that is currently being tracked:

```
$ git mv old_file_name new_file_name
```

How to erase a file (physically and from the tracking system):

```
$ git rm file_not_needed
```

How to add a new (existing) file or directory to the tracking system:

```
$ git add file_name
```

How to add all new (untracked) files to the tracking system:

```
$ git add -A
```

How to invoke a GUI that shows you history of changes to the code

```
$ gitk
```

How to use git on a GUI interface rather than on the command line:

```
$ git gui
```

How to initialize git in current directory (you only need to do this once at the very beginning and when you need to start from scratch):

```
$ git init
```

How to change editor used in git:

```
$ git config --global core.editor emacs
```

How to declare your name in the git version control system (you only need to this once):

```
$ git config --global user.name "John Doe"
```

How to declare your email address (for labeling purposes only):

```
$ git config --global user.email "john.doe@nih.gov"
```

How to clone a git repository in preparation to put it on a git server:

```
$ git clone --bare LSNM LSNM.git
```

How to copy a git repository to a server (shared folder in Helix, for example):

```
$ scp -r LSNM.git user@helix.nih.gov:/home/horwitzb/bhguest/sgihome
```

How other users can clone down the git repository found in the server so they can work on the code locally:

```
$ git clone user@helix.nih.gov:/home/horwitzb/ghguest/sgihome/LSNM.git
```

How to push changes to the repository on the server:

```
$ git push origin master
```

where 'origin' has been defined as:

```
$ git remote add origin user@helix.nih.gov:/home/horwitzb/bhguest/sgihome/LSNM.git
```

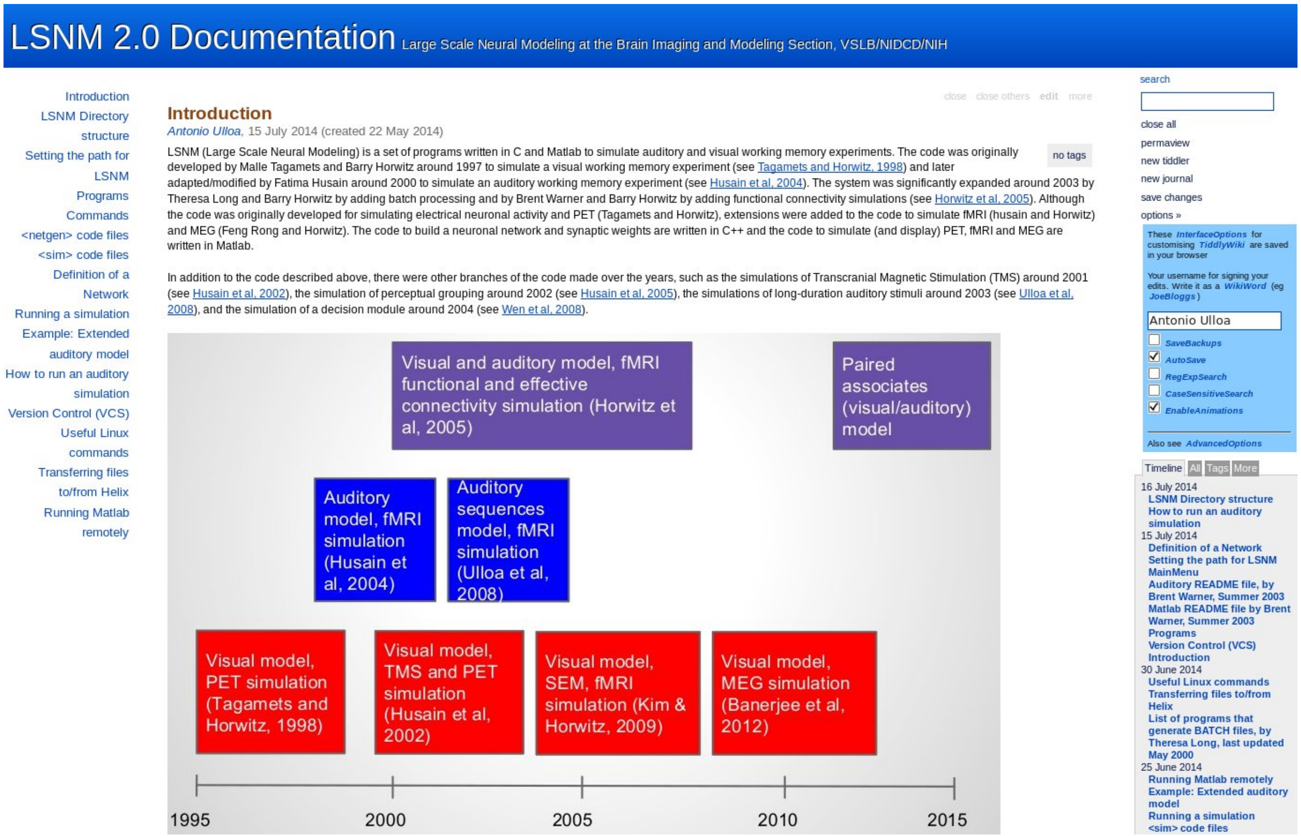

Figure 1. A Snapshot of the LSNM's wiki-style documentation page

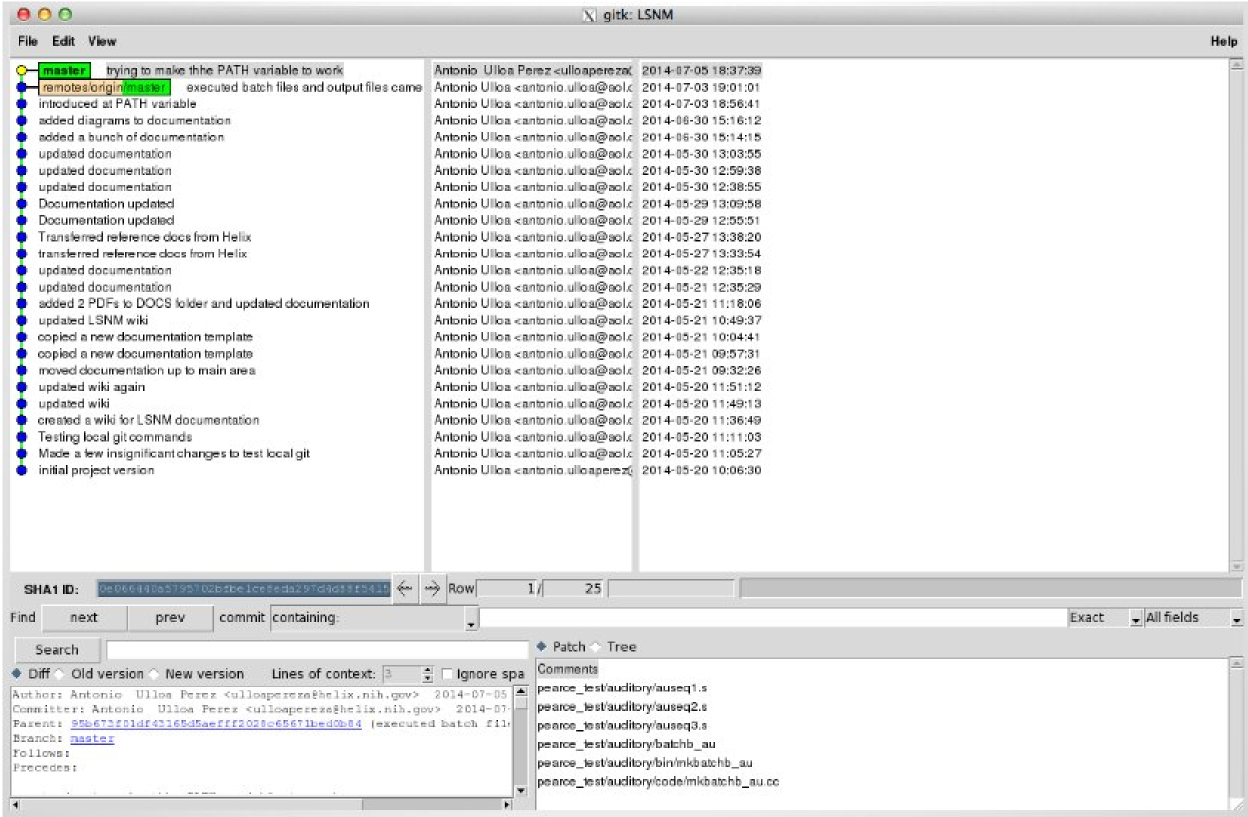

Figure 2. Snapshot of the version control system’s GUI browser, showing the history of updates to LSNM

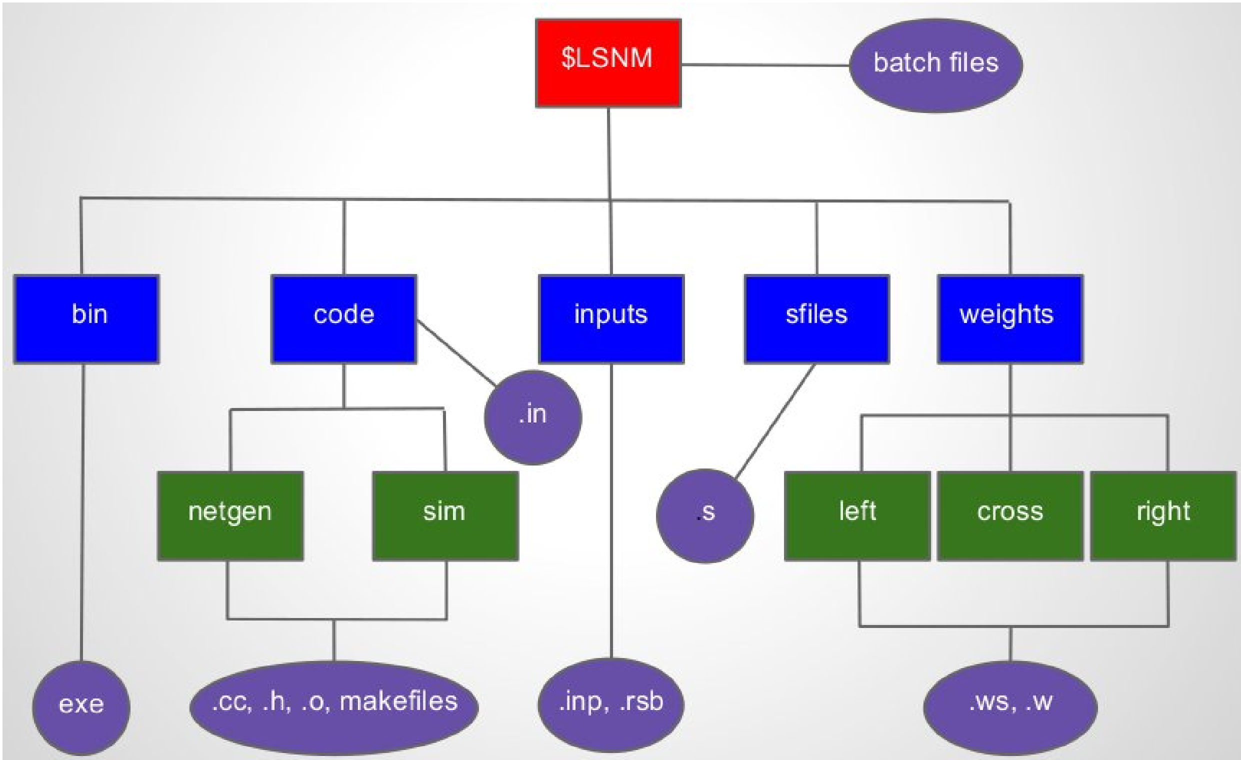

Figure 3. Graphic depiction of LSNM's current directory structure (Matlab directory not shown)