/home/bhguest/brent/auditory
nimbus%
nimbus% more README
Written by Brent Warner
Summer 2003

PROTOCOL FOR RUNNING AN AUDITORY SIMULATION

1. In home directory (where you want to run the sim), ensure that the following directories exist and contain these file types.

dir: /bin
file types: Contains all of the executable programs. When a program is compiled in the /code directory it is placed here. If you are setting up the simulation for the first time, this directory should be empty. For simulations after the first one, unless a change is made to the C code, these files will not be altered.

dir: /code
file types: *.cc and corresponding *.in files, makefile
Contains all of the C code for the simulation, including batch jobs. Most of the *.cc files have a *.in file that goes along with them and specifies several parameters that can be altered from sim to sim.

subdir: /netgen
file types: *.cc, *.h, *.o, makefile
Contains the code for the network generator that creates all of the region-region connections within a hemisphere. Has its own makefile.

subdir: /sim
file types: *.cc, *.h, *.o, makefile
Contains the code for the actual simulation program. Has its own makefile as well.

dir: /inputs
file types: *.inp, *.rs*
*.inp files are the actual inputs to the model. They specify which nodes in the inputs regions (MGNs and GMGN) are turned on. *.rs files are the sequence files...i.e. the order in which inputs occur, switching from high to low attention, resets, etc.

dir: /sfiles
file types: *.s
*.s files are sort of like inputs. They are responsible for switching between high and low attention and resetting the system.

dir: /weights
file types: *.ws

subdir: /cross
*.ws files are "weight" files that specify the connections between regions. This directory contains the *.ws file for EVERY possible left-right and right-left connection. Note

that not every one is used in every run....the ones used
are chosen at random by the simulation unless it is set that
all are used. These *.ws files can be regenerated using
the crossws_au program.
subdir: /left, /right
Contains the *.ws files for that hemisphere's within-
hemisphere connection made, both excitatory and inhibitory.

Once all of this setup has been done once, it does not need to be done again.

2. SETTING CORRECT DIRECTORY STRUCTURE

There is a fairly elaborate system of directories and subdirectories that needs to be maintained during a run. The simulation therefore needs to be set up to run in the correct directory and also place output in the correct location.

Most or all of the files in the /code directory, it's subdirectories /netgen and /sim, and the /inputs directory contain specific directory information. In *.cc files, the home directory is saved in a variable called BASE. In *.in files, the dir is listed in the first line of the code. In *.rs* (sequence) files there are several #include statements giving locations. Each of these must be changed to reflect the user's preferred dir hierarchy.

It is easiest to the the grep command in unix to find these instances, for example:

grep 'BASE' *

will find every instance of "BASE" in the current directory. You would then go and change the value of BASE in the code.

The directory where output gets placed is defined in first line of the .in file corresponding to the batch file that you are running. This directory must exist before you run the batch!

3. MAKING THE NECESSARY PROGRAMS (for first time users)

Each batch job runs a number of the same programs which must be made so that they exist in your /bin. Each of the following must be made SEPARATELY, not all in one line as I have listed below.

in /code:

make altgenw_au, altgenw_L_au, crossws_au, crosswt_au_i, genw_au, mkattn_au

in /code/netgen:

make netgenC_au, netgenC_L_au

in /code/sim:

make au_sim1

Note that there are other programs (*.cc files) that exist in these directories. These are others will be made when they are necessary.

4. RUNNING A SIMULATION

The simulations are all run in a batch format, meaning that several simulations are run using varying parameters to represent multiple subjects, inputs, attention levels, region connections, etc. The batches that I have used are:

mkbatchr_au.cc -- Right hemisphere only. Can alter attention level and one region-region connection (usually STG->FS, but can be changed in genw.in)
Uses auseqr.rs for sequence file

mkbatchw_au.cc -- Both R and L hemispheres. R receives form input, L receives noise. Can alter attention levels and one region-region connection as in above.
Uses au_test.rs for sequence file (currently)

mkbatchb_au.cc -- Same as mkbatchw_au, except now allows user to change connection strengths for multiple region-region connections in RIGHT HEMISPHERE ONLY. Additionally, cross-hemisphere connections are regenerated for each trial.
Uses auseq1.rsb, auseq2.rsb,... for sequence files (number corresponds to total number of trials)

mkbatchrlc_au.cc -- Essentially like mkbatchb_au, except the left and right hemispheres have been combined into one large hemisphere with half specific neurons (old right) that are task-dedicated and half non-specific neurons (old left) that are "distracted". The only real difference from mkbatchb_au is that now any alterations of connection strength in the specific (right) are mirrored in the non-specific (left). Before, the left hemisphere connections remained constant.
Uses same sequence files as mkbatchb_au.

Each of these has a *.in file to go along with it. Before running one, check *.in file and set the desired values. This file will specify many parameters, which are explained in the actual file. Also, you should change the sequence file(s) that particular batch uses to reflect the organization of inputs you want.

TO ACTUALLY RUN THE SIMULATION (using mkbatchb_au as an example)

go the the /code directory and use the following command sequence.

```
make mkbatchb_au        -- makes the executable in the /bin
../bin/mkbatchb_au      -- runs the executable to produce the batchfile,
cp batchb_au ..         -- copies the batch up one directory, where it will be
                           run
cd ..
chmod u+x batchb_au     -- makes the batchfile executable
batchb_au               -- runs the batch
```

All of the output files from the batch will be placed in the directory specified in line 1 of the corresponding *.in file.

5. SOME OTHER USEFUL PROGRAMS

C PROGRAMS:
The mkbatchb_au and mkbatchrlc_au versions have a sequence file for each trial that is done. Thus, the output files are split up trial-by-trial. Since it is useful to see all of the trials at once when analyzing the data, there is a program that combines them called mkcatout_au.cc

mkcatout_au is run similarly to the batch jobs. Go to its *.in file, change the necessary parameters outlined there (make sure the output directory is set) and use the following sequence of commands while in /code.

```
make mkcatout_au
../bin/mkcatout_au
catout_au
```

Depending on the number of subjects/attention level/trials, this program can take quite awhile. It traverses all of the output directories, unzips the files, concatenates them for a subject/attn level, places the new files in the attention level directory (/b30 for example), and then rezips everything.

MATLAB PROGRAMS:
There are several useful matlab programs for visualizing output. All of these are currently located in /home/bhguest/brent/matlabprogs. Make sure this is added to you matlab path (or copy them over to your directory and add that path). Note that each of these programs only work on the files in the current directory.

loadmovieB_au -- creates a movie of the output. Left hemisphere regions are on the top row, right on the bottom. The order in both from left to right is: MGN, A1, A2, STG, FS, D1, D2, FR

plotauditory -- plots the electrical activity for the right hemisphere. pre-STG units are on figure 1, STG-PFC are on figure 2.

plot_audleft -- same as above but left hemisphere.

plotaud_rl -- plots the homologous left and right hemisphere regions on top of each other on the same plot to represent a combined hemisphere. Used with mkbatchrlc_au.cc

maketopo -- useful when you would like to see the shape of your inputs. Run this and then plot mgns or gmgn.