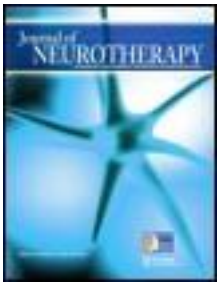


Journal of Neurotherapy: Investigations in Neuromodulation, Neurofeedback and Applied Neuroscience

Open Source Method of Graphical QEEG Analysis Using PERL and Visual Basic

Jon A. Frederick MS [a]

[a] Department of Psychology , University of Tennessee , USA

Published online: 20 Oct 2008.

To cite this article: Jon A. Frederick MS (2001) Open Source Method of Graphical QEEG Analysis Using PERL and Visual Basic, Journal of Neurotherapy: Investigations in Neuromodulation, Neurofeedback and Applied Neuroscience, 4:4, 39-47, DOI: 10.1300/J184v04n04_07

To link to this article: http://dx.doi.org/10.1300/J184v04n04_07

Open Source Method of Graphical QEEG Analysis Using PERL and Visual Basic

Jon A. Frederick, MS

Existing products for displaying coherence, phase, and asymmetry data make statistical assumptions that are not optimal for every application. In particular, the most popular database software packages usually require the assumption that means and standard deviations derived from the manufacturer's normative subject group are valid standards of comparison for your particular subject or group. If you wish to make comparisons between experimental and control groups of your own, or to make within-subject comparisons, this article will show you how to graphically represent your coherence, phase, or asymmetry data.

System Requirements: (1) Windows 95, 98 or 2000 operating system; (2) Pentium class processor 166 MHz or greater with 32 MB RAM or greater; (3) Microsoft Excel for Windows (97 or 2000); and (4) monitor or printer supporting 256 colors.

The macro recorder and Visual Basic editor in Microsoft Excel are powerful tools for automating redundant tasks, such as importing, parsing, and formatting data. Any series of mouse or keyboard actions can be captured by selecting TOOLS: MACRO: RECORD (under the TOOLS menu, select MACRO, and then select the RECORD option).

Jon A. Frederick is a graduate student, Department of Psychology, University of Tennessee.

Address correspondence to: Jon A. Frederick, 307 Austin Peay, Knoxville, TN 37919 (E-mail: smiile@utk.edu).

The author is grateful to Dr. Joel Lubar and the University of Tennessee Department of Psychology for the freedom and encouragement to explore computer science, and to Harley Helterhoff for his excellent consultation as a programmer. Jon Frederick's work is supported in part by a graduate assistantship from the Department of Information Infrastructure at the University of Tennessee.

The software prompts you for a name for your macro, and then translates your actions into Visual Basic code, which can then be viewed by looking at TOOLS: MACRO: VISUAL BASIC EDITOR. Once inside the Visual Basic editor, if the code for your particular macro isn't already displayed, it can be found by clicking and browsing through the MODULES in the PROJECT window. Thus, Excel can teach you how to write Visual Basic code, which you can then modify to write more powerful applications.

This application was written by drawing a head with nineteen electrode locations on it according to the international 10-20 standard. The DRAWING toolbar was then used to create 171 lines between all locations, while capturing these actions with the macro recorder. Re-drawing the line for a new head is as simple as executing the macro with the same name as the electrode pair you wish to represent (i.e., O1T3, F3F4, etc.). Red lines are used to indicate positive changes and blue lines represent negative ones. The degree of saturation of red or blue indicates the magnitude of the change. To try this software with some sample data, visit the following web site:

http://www.snr-jnt.org/JournalNT/Vo14/vbmapper.htm

and download the following three files: vbmapper.pl, 1020.xls, and sampldat.txt

Vbmapper.pl is a Windows program, written in PERL that translates lists of electrode pair variables and statistics into the Visual Basic code that draws the heads. Any statistic, such as standard errors (t-values), correlation coefficients (r or r-squared), or raw difference scores, is supported. Understanding how this program is written will help you to adapt it to the needs of your research. The appendix shows how to install PERL to give you this flexibility. However, for a faster demonstration that does not require PERL, just download the executable, which is named vbmapper.exe

Test Driving Vbmapper. Vbmapper.pl and vbmapper.exe are, functionally, the same program. Vbmapper.exe is the compiled version and thus has all of the PERL subroutines built-in, so it does not require PERL to be installed. Its file size is, however, 300 times larger, and you do not have the option of reading or editing the program. Further, it will only run under the operating system (Windows) for which it was compiled. Vbmapper.pl is the "source code" for the .exe version, and will run on any computer that has PERL installed (including Macintosh and X-Windows). Compiling executables is useful for sharing programs with people who don't want to become programmers, and also for protecting proprietary source code. My program is "open source" because I encourage users to read and modify, and share the source code.

To run vbmapper, first, make sure that your data file (e.g., sampldat.txt) is in the same directory, and click on vbmapper.exe or vbmapper.pl from the Windows Explorer, or enter "vbmapper.exe" or "vbmapper.pl" (without the quotes) from the DOS command line. Enter the minimum and maximum absolute value of the statistic to be displayed ("alpha" or critical value), and the points of zero and maximum saturation. Default values are already displayed in the text entry fields, and provide an interesting view of the sample data set. Enter your frequency variables as a comma-delimited list, without spaces (the default list for the sample data file is "TH,LA,HA,LB,HB,GA"). This version of the software supports the variable naming conventions of the Lexicor Exporter program, but could be readily modified to support other software packages. Any order of electrode pair names (FZPZ or PZFZ) is already supported. Finally, select the name of the sample data file from the list and click RUN. Visual Basic code for drawing the heads is automatically copied to the clipboard, and is ready to paste into the Excel Visual Basic editor.

Open 1020.xls and, when prompted, ENABLE MACROS. Make sure no other Excel documents are open. Invoke the Visual Basic Editor (TOOLS: MACRO: VISUAL BASIC EDITOR). If an empty module does not appear, find one in the modules in the PROJECT window on the left, or create a new one with INSERT: MODULE. Paste the contents of the clipboard into the empty module. Go back to the Excel window, and make sure that you are on Sheet1. The mapping macro can now be executed by selecting TOOLS: MACRO: MACRO, selecting the macro named aaaamap (given this name so it will always appear at the top of the list), and clicking RUN. Six coherence maps should now appear on Sheet4. The default view on Sheet4 is 40%, but can be changed by selecting ZOOM from the VIEW menu. The graphics are now ready to be copied, rearranged, annotated with text, and printed.

They can be copied to the clipboard as a bitmap image (for editing in Photoshop, for example) by holding down the ALT key and pressing the PRINT SCREEN key (you will want to go to TOOLS:OPTIONS: VIEW and turn off GRIDLINES before you do this). Note that these graphics take up a lot of memory, so your computer will slow down substantially if you try to draw numerous heads at once, fail to save your changes, or try to do this with numerous other applications open. It is recommended that changes in Excel be saved to a different filename using the SAVE AS command under the FILE menu, to avoid corrupting 1020.xls, which is a template file. It is also recommended that selecting graphics for cutting, copying, or deleting be done exclusively with the ARROW tool in the DRAWING toolbar. If the drawing toolbar does not appear on your worksheet by default, you can invoke it by selecting TOOLS: CUSTOMIZE: TOOLBARS: DRAWING; remember to turn the ARROW tool off when you are finished or you will be very frustrated trying to click on cells. Use of selection methods other than the ARROW tool reveals a bug in Excel: that is, invisible graphic objects accumulate in your worksheet, taking up memory and grinding your computer to a halt.

The plots from the sample data represent the differences in t-scores in coherence between two reading tasks in 17 young adults (data from Angelakis et al., 1999). The results are presented in Figure 1 and Figure 2. Figure 1 shows the effects that are 2.11 or more t-scores from zero, or univariately $p < .05$. The relatively modest number of findings in the lower frequency bands suggests that they might result from random error. Figure 2 shows all coherence differences for a much lower criterion, $t > 1.0$. However, scores are sorted and plotted in order of magnitude, so that the more "significant" lines appear in the foreground. To allow for maximum contrast, the zero saturation point is set slightly below the minimum critical value in both graphs so that the least significant lines are closest to pink and sky-blue. The maximum saturation is set for $t > 3.0$ in Figure 1 and Figure 2.

This method shows that while the right posterior decreases in 4-8 Hz and the frontal increases in 10-12 Hz occur at a level less than chance in Figure 1 ($p < .05$ for only 3 pairings, whereas 8.55/171 possible pairings = .05), Figure 2 reveals each of these changes to be consistent with an underlying factor which is changing all of the surrounding pairs in the same direction. Our research has suggested that the balance between type I and type II error is best optimized by testing the significance of *patterns* among univariate effects, rather than for the univariate changes themselves (Frederick, Angelakis, Lubar, & Stathopoulou, 2000). Some-

FIGURE 1. Significant t-score differences in coherence between two reading tasks (n = 17, t > 2.11, p < .05).

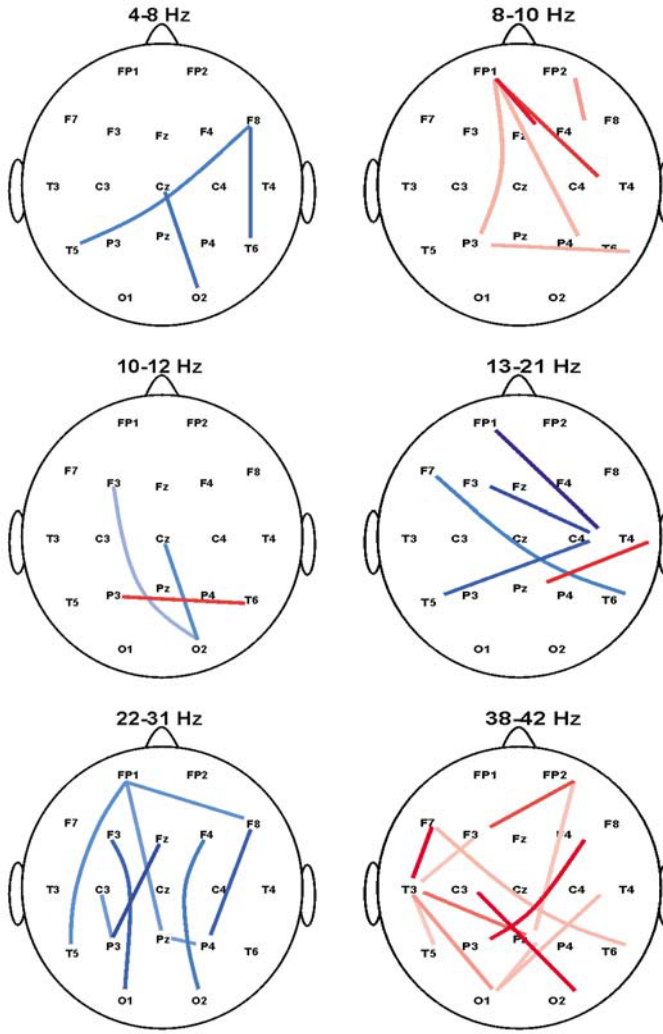
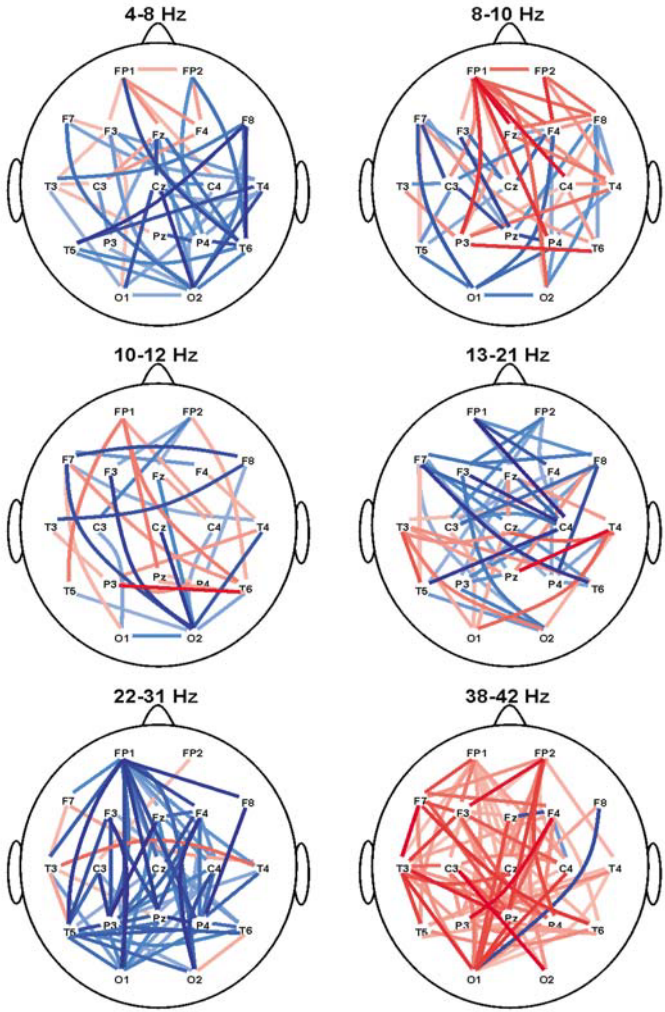


FIGURE 2. T-score differences in coherence between two reading tasks (n = 17, t > 1.0). Significance is proportional to saturation of red/blue; results of highest significance are plotted in the foreground.



times, significant patterns can be detected among changes that are not significant univariately. The large number of measurements in QEEG data should be seen as a source of power rather than a source of error–in psychometric terms, the massive diversity and redundancy of a QEEG data set is the equivalent of a census form or a full-scale IQ test, compared to a brief telephone interview. When properly analyzed, a large number of test items can reduce within-subject error variance by comprehensively assessing different aspects of the same underlying factor, thus requiring a smaller subject number to converge around the true population mean. This graphical method is no substitute for rigorous statistical analysis, but it is of heuristic value for identifying patterns in your data.

Future Directions. A number of modifications could make this software more powerful and more elegant. I hope to provide the following upgrades in the near future: (1) an option for mapping power and amplitude data; (2) other color options (e.g., to support black-and-white only publications); (3) statistics modules–the current version assumes that you have already done your statistics. Hopefully, future versions will accept raw data files as input and provide correlations, t-tests, etc., as output. For more details, visit the vbmapper homepage (http://www.snr-jnt.org/JournalNT/Vo14/vbmapper.htm). One of the advantages of open source software is that its evolution does not depend upon the time or interests of one or a few individuals. I hope that users will accept the challenge of customizing this product to their own applications and sharing their innovations with the community.

REFERENCES

Angelakis, E., Lubar, J.F., Vanlandingham, P., Stathopoulou, S., Blackburn, J., & Towler, K. (1999). QEEG of normal college students during different reading tasks that characterize subtypes of dyslexia (SNR conference abstract). *Journal of Neurotherapy*, *3*(4), 55.

Frederick, J.A., Angelakis, E., Lubar, J.F., & Stathopoulou, S. (2001). EEG coherence effects of reading task conditions in young adults (2000 SNR Conference Abstract). *Journal of Neurotherapy*, *4*(4), 82-84.

Schwartz, R., Christiansen, T., & Wall, L. (1997). Programming Perl. Sebastopol, CA: O'Reilly & Associates.

Srinivasan, S. (1997). Advanced Perl programming. Sebastopol, CA: O'Reilly & Associates.

Wall, L., Christiansen, T., & Schwartz, R. (1996). Learning Perl. Sebastopol, CA: O'Reilly & Associates.

APPENDIX

Installing PERL

PERL stands for "Practical Extraction and Report Language," and was first written by Larry Wall in 1986. Originally designed as a scripting language, to automate redundant tasks for UNIX system administrators, PERL has evolved into a full-featured programming language freely available for nearly every operating system (Wall, Christiansen, & Schwartz, 1996). PERL is known for being easy to learn and use, with a community of thousands of Usenet group subscribers providing free technical support (Schwartz, Christiansen, & Wall, 1997). "Laziness" is said to be a virtue in a PERL programmer, because an international archive (www.cpan.org) offers thousands of free scripts and modules to accomplish many tasks.

PERL can be downloaded from http://www.activestate.com. To get to the download site from this home page, click on the link to "Active Perl" (the current version is 5.6), and then "Download Now." Windows 95/98 or NT users will also need to download the Windows Installer available at this site (click on the link for "Windows NT" or "Windows 95/98" after the note for "Windows Users"). The installation file for PERL itself is the link to "Windows Intel" located under "ActivePerl 618." To clarify any confusion, ActivePerl 618 is Activestate.com's independent distribution of Larry Wall's PERL 5.6. Some of these links and directory names may change with upgrades, but readers in the more distant future should still be able to find a valid version of PERL for Windows by browsing activestate.com or by searching the web for "ActivePerl." By the way, ActivePerl is 8.7 megabytes, so expect to be waiting for a while if you are using a 56K modem.

The installation is the same as for any modern Windows software. Save the "Windows Intel" link (the actual current file name behind this link is ActivePerl-5.6.0.618-MSWin32-x86-multi-thread.msi) to your hard drive. Make sure you take note of where you are saving it before you click the SAVE button–I recommend the Desktop. If you are running Windows 95/98 or NT, download and run the Windows Installer first (the current name of this file is InstMsi.exe). Then, simply click on the ActivePerl install file, and follow the instructions provided by the setup program. All of the default settings should work well for this application.

Vbmapper.pl takes advantage of a package or "module" known as Tk, which creates a graphical user interface ("GUI" or Windows program). Without Tk, all of the input/output of a PERL program is controlled at the command line, much like a DOS command. Tk was written as part of the Tcl programming language by John Ousterhout, and ported to PERL by Nick Ing-Simmons (Srinivasan, 1997). Like other PERL modules, ActivePerl provides an easy method of installing Tk into your version of PERL. After installing PERL, make sure you are connected to the Internet, and execute "PPM" (without the

quotes) at the DOS command line. Then, at the PPM > prompt, enter "install Tk" (without the quotes). Make sure the T is upper case and the k is lower case. PPM will automatically download the Tk module from activestate.com, and complete the installation for you. Be patient during the download, because your DOS window will be frozen for a minute or two.