Review

AFNI: What a long strange trip it's been

Robert W. Cox *

Scientific and Statistical Computing Core, National Institute of Mental Health, National Institutes of Health, Department of Health and Human Services, 10 Center Dr. Room 1D80, Bethesda, MD 20892-1148, USA

ARTICLE INFO

ABSTRACT

AFNI is an open source software package for the analysis and display of functional MRI data. It originated in 1994 to meet the specific needs of researchers at the Medical College of Wisconsin, in particular the mapping of activation maps to Talairach–Tournoux space, but has been expanded steadily since then into a wide-ranging set of tool for FMRI data analyses. AFNI was the first platform for real-time 3D functional activation and registration calculations. One of AFNI's main strengths is its flexibility and transparency. In recent years, significant efforts have been made to increase the user-friendliness of AFNI's FMRI processing stream, with the introduction of "super-scripts" to setup the entire analysis, and graphical front-ends for these managers.

Published by Elsevier Inc.

Contents

The past

Contingency. AFNI is one result of unexpected events. I met a woman and fell in love. She had a tenured faculty position in Milwaukee, so I moved there in 1993, got married, and looked for a job. Meanwhile, FMRI had started up at the Medical College of Wisconsin (MCW) in late 1991, thanks largely to the acumen and efforts of 2 graduate students: Peter Bandettini and Eric Wong (Bandettini et al., 1992). By early 1993, Jim Hyde (chairman of MCW Biophysics) had decided they were "drowning in data" and was casting about for a lifeline. At that moment, I showed up and looked like I knew something about data analysis. I got the job that autumn, and spent much of the next 6 months frantically trying to learn about the fields of MRI, FMRI, neuroscience, and their guardian phalanxes of jargon and tropes. (Plus contribute to grant writing.)

I started developing AFNI (Cox, 1996) in mid-1994 to meet the needs of the MCW researchers using this new FMRI tool to explore brain function. Their most insistently expressed desire was the ability to transform their data to Talairach–Tournoux coordinates—an inherently 3D process. Up until then, all FMRI processing and display at MCW had been in 2D slices. At that time, when the Web was just starting, most research groups processed their data with home-grown software, and good information about the tools used at other sites was hard to glean. After pondering the Talairach–Tournoux atlas (Talairach and Tournoux, 1988) and some FMRI data for a while, I sketched out a plan for a volumetric-based software tool to enable researchers to surf through their datasets in 3D, jumping between multiple datasets, and overlaying functional maps on structural volumes as desired.

My design and implementation from the start was based on a few generic principles:

a) Let the user stay close to her data and view it in several different ways, providing controls to allow for interactive adjustment of statistical thresholds and colorization. This ability helps the user

* Fax: +1 301 402 1370.
E-mail address: robertcox@mail.nih.gov.

become familiar with the structure of her data and to understand her results.

b) Let the user compose and control the data analysis and image processing as needed for her particular research problems and predispositions.

c) Let the user see the intermediate results of the processing sequence, so that it is easy to backtrack to understand how particular results were obtained. This ability has proved especially important when trying to decipher unexpected results (e.g., activation "blobs" that don't make sense).

d) Everything should be open-source and the software should be modular enough that other people could usefully contribute to it.

By September 1994, I had the first version of the AFNI graphical interface ready for MCW faculty to try. Using it, a user could view low-resolution 3D functional activation maps overlaid on sagittal, axial, and/or coronal slices from a high-resolution volume, with the ability to scroll through slices at will in any or all viewer windows, alter the statistical threshold on the activation map, and change the colorization scheme. The first researchers who tried it were ecstatic—they could finally see their brain mapping results the way they wanted, and control how the maps appeared. *And* they could transform their results to Talairach–Tournoux space, which was why it all had started.

I developed AFNI to run on Unix + X11 workstations with 32+ MB of RAM (which was a lot back in those ancient times). Datasets only had 3 dimensions; it wasn't practicable to read in and process entire 3D + time collections of images—the basic units of FMRI data. Processing of FMRI data to produce functional activation datasets proceeded slice-by-slice, and only at the end were the 3D functional overlay datasets assembled for visualization in AFNI. By mid-1996, with more memory available, an extensive re-write of AFNI allowed datasets to have a fourth dimension, and all data analyses moved from a 2D basis to 3D. This change dovetailed with the development at MCW of real-time FMRI activation analysis (Cox et al., 1995) and volume registration (Cox and Jesmanowicz, 1999), both of which were incorporated into AFNI soon after they were developed. About the same time, the central FMRI time series analysis methodology in AFNI evolved from the correlation method (Bandettini et al., 1993) to multiple linear regression and hemodynamic impulse response function deconvolution.

In 2001, the NIH recruited me to establish a group to support the rapidly expanding FMRI research efforts on the Bethesda campus. As this group expanded, AFNI branched out in several directions, most fundamentally to include processing of dataset defined over triangulated cortical surface models, not just 3D regular grids, in the SUMA software (Saad and Reynolds, this issue).

Structure

Originally, AFNI dealt only with 3D (and later, 3D + time) datasets stored in a format of my own devising, where all auxiliary non-image data are stored in a text file of the general form "attribute-name = values", and the much bulkier image values are in a separate binary file. One advantage of this approach is that arbitrary expansions of the auxiliary data are simple, and at the same time image data I/O is also straightforward. One major disadvantage is that this custom format is not readily useful with other software. In the late 1990s, the NIH became concerned with the interoperability of neuroimaging research software. A working group was formed, chaired by Stephen Strother and including representatives from most major FMRI software developers, which met under the aegis of the NIMH and NINDS extramural programs. The upshot was the new and simple NIfTI-1 format for storing data defined over regular grids (Cox et al., 2004). This format includes the ability to include arbitrary application-specific extensions to the header

information, which AFNI uses to store (via XML) various auxiliary information not otherwise defined in the NIfTI-1 specification—for example, the history of the commands that produced a given dataset are saved in its header. AFNI now supports NIfTI-1 for input and output at the same level as the AFNI custom format. All that the user has to do is specify that the output filename ends in ".nii" or ".nii.gz" to get a NIfTI-1 output file. We support other formats, such as DICOM, mostly by the use of file converters.

AFNI comprises the interactive visualization program of the same name, a collection of plugins to the graphical interface, plus a large set of batch (Unix command line) data processing programs. Most of these batch program names start with the characters "3d" (e.g., *3dDeconvolve* is an FMRI time series deconvolution program)—this naming scheme was chosen to emphasize that they process data stored in 3D image volumes. In the early days of FMRI, most neuroscience researchers still thought in 2D slice terms, not in 3D volumetric terms; I wanted the program names to help change this conceptualization. However, AFNI "3d" programs that operate on a voxel-wise basis are not actually restricted to data defined over 3D grids; for example, it is simple to map a 3D FMRI time series dataset to a cortical surface domain (e.g., via *3dVol2Surf*), and then carry out the rest of the analyses on the surface model using the same regression and statistical software as in the volumetric analysis. Only the inter-voxel processing steps, such as spatial smoothing, require special purpose surface-aware software. The recent development of the GIfTI file format for surfaces and surface-based datasets now allows interoperability of surface-analysis software from different packages, much as the NIfTI-1 format does.

A competent C programmer can write an AFNI "3d" batch program without climbing a steep learning curve. For example, the core AFNI library provides utility subroutines to create an output dataset from a programmer-supplied function which takes a collection of numbers from the input dataset at a single voxel and returns another collection of numbers to be stored at that location in the output dataset. For plugin creation, the AFNI graphical user interface has a set of library functions to allow a programmer to create a "fill in the form" style of interface (cf. Fig. 1 and the *InstaCorr* setup control panel). However, my dream of having significant contributions to AFNI from outside developers has only been partially realized. A few such programs have been donated and are distributed with AFNI, but most people (understandably) prefer to work independently. Fortunately, the advent of the NIfTI-1 format makes such third-party efforts readily usable with AFNI batch programs.

With the introduction of SUMA, Ziad Saad and I had to design a "live" interchange protocol for external programs to talk to AFNI, exchanging data and commands with the independently running program. At the NIfTI working group meetings, it was clear that no other FMRI software group was interested in such an effort, so we developed our own XML table-based format for this purpose. An external program can attach to AFNI (or SUMA) via a TCP/IP socket—if the programs are on the same computer, the communication channel can be switched to shared memory, which is faster for exchanging large amounts of data. Once this I/O channel is established, the external program can send commands to AFNI to open windows for image or time series graph viewing, and send new 3D datasets for display. In this way, a user can script the viewing of large numbers of datasets—we have found this ability to be very useful when processing large volumes of data, such as the FCON1000 collection (Biswal et al., 2010). When computing with a lot of datasets, some algorithm (e.g., registration) always seems to do badly on a few cases, and being able to scan systematically through the entire collection to find and fix such problems is important. The external driving program need not be in C; we also provide sample programs in Matlab and Python to demonstrate this functionality. The real-time FMRI acquisition system at the NIH FMRI facility uses a set of Perl scripts to control AFNI. SUMA uses this capability to be closely linked to AFNI, exchanging
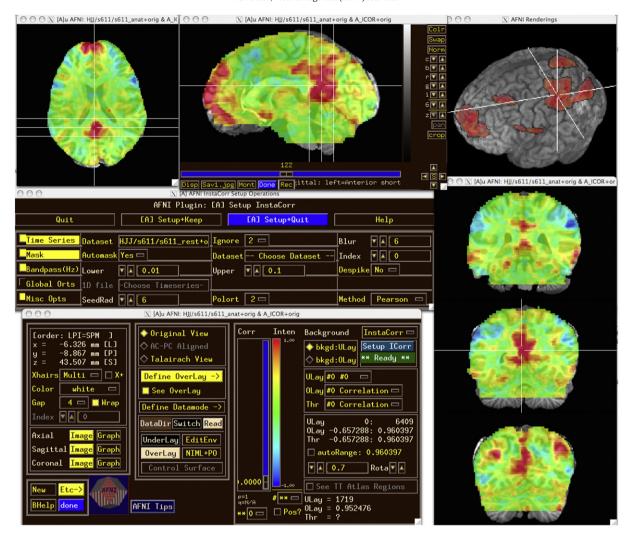
Fig. 1. The AFNI graphical user interface (on OS X), with the individual subject *InstaCorr* module setup window shown, and the *InstaCorr* results from clicking in the posterior cingulate region (at the crosshair location) overlaid on the subject's SPGR volume—un-thresholded in slice views and thresholded at *r*>0.6 in a 3D see-thru volume rendered view. The sagittal slice viewer shows the AFNI controls which have been turned off (for de-cluttering) in the other image windows.

crosshair locations, datasets, and colorized overlays between the folded 2D surface domains and the 3D grid domains.

The present

AFNI has had the reputation for being hard to use. In the past, users were required to create their own processing scripts to chain together the various "3d" programs to achieve the goals they wanted, using our sample scripts as starting points. Physicists and statisticians like this style of operation, since they generally want complete transparency of and control over every detail of the analysis process. It is common in neuroscience, however, that a researcher wants first to see some activation results, and possibly later go back to refine the analysis—such a user wants an easy-to-use standard and reliable analysis path to follow at the start.

We have worked hard to make it simple to run the basic and common analysis pathways through the AFNI suite of programs. Our first big step was the creation of the *afni_proc.py* super-script, which is a (Python) command line program that takes as input the list of MRI 3D and FMRI 3D + time dataset files for a single subject, the stimulus timing files, and a few more pieces of information, and produces a script that can process all the data from that subject to give the statistical maps ready for group analysis. Different classes of stimuli are given user-chosen labels, and contrasts are specified symbolically (e.g., "+Faces −Houses"). All the intermediate datasets are saved,

making it easy to check the analysis if the final output seems unusual. The generated processing script itself serves as a record of what was done, and can be manually edited if non-standard steps need to be injected into the calculations. We strongly recommend that all but the most experienced AFNI users start with or switch to using *afni_proc.py* to generate their FMRI analysis pipeline.

Our next step was the creation of a graphical interface, *uber_subject.py*, to make it easy to run *afni_proc.py* by button-clicking and menu-selection. This GUI is now available with AFNI, but is still a work in progress, growing in flexibility at a brisk pace. We also have a command line script, *align_epi_anat.py*, which is the recommended tool for 3D registration processing in AFNI—despite its name, suggesting it is only for aligning anatomical and EPI volumes, it is capable of other tasks, including mapping 3D and 3D + time datasets to template spaces. We are currently developing a graphical interface for these alignment tasks, which are often needed in non-FMRI applications. A usable super-script for setting up group (inter-subject) analyses has also been written and is being refined.

It is the flexibility of AFNI programs that is one of our software's greatest strengths. Most of our analysis codes have dozen of options —one source of the obsolescent "AFNI is hard to use" meme, from the time when it was necessary for each user to create her own processing script ex nihilo. Each option is there to meet some particular need. For example, the *3dDeconvolve* linear regression program has

many different built-in hemodynamic response models, including amplitude- and duration-modulated stimuli, fixed-shape and decon-volution hemodynamic response models, because FMRI-based inves-tigations are so mutable and the questions asked about the response amplitude and shape can vary so much. Response amplitudes can be calculated for each stimulus separately, to be merged at a later step, or pooled across stimuli that are classed together—the most common form of analysis in FMRI. For more specialized purposes, other time series regression programs are available, including nonlinear voxel-wise regression (e.g., for fitting pharmaco-kinetic and -dynamic models) and a code for linear regression with voxel-dependent models, including parameter sign constraints and L^1, L^2, and LASSO fitting options. In addition, over the years we have implemented a large number of utility programs for various statistical and 3D image processing purposes, available to users who have particular processing needs. Of these, perhaps the most useful is *3dcalc*, a gener-al purpose voxel-wise calculator program—the Swiss Army knife of AFNI, not well-adapted for any single purpose, but very helpful for getting some computation done quickly.

At the group analysis level, AFNI has options ranging from simple *t*-tests to a sophisticated mixed-effects meta-analysis tool (*3dMEMA*) that incorporates voxel-wise estimates of the per-subject variance of the FMRI response amplitude and can model outliers. The linear mixed effects code (*3dLME*) can model and fit correlations between data points when multiple parameter estimates are carried to the group level from each subject simultaneously (e.g., when comparing and contrasting the shape of the hemodynamic response among subjects, not just its ampli-tude). Subject-wise covariates are easily incorporated into these analyses by including a file containing a table of these values, and the *t*-test pro-gram also allows for voxel-wise covariates for each subject (e.g., gray matter volume fraction). AFNI also includes several flavors of task-based and resting-state connectivity and network analyses.

The AFNI graphical interface contains several tools for quick interac-tive data analyses. The *InstaCorr* module allows the user to load a rest-ing-state FMRI dataset and pre-process it – masking, despiking, blurring, bandpassing – then, when the user clicks on a voxel location, the correlation map of that voxel's time series with all other voxels' data is computed and displayed within milliseconds. The group version of *InstaCorr* also performs a 1- or 2-sample *t*-test (optionally allowing for subject-wise covariates) across the individual subject $\tanh^{-1}$ trans-formed correlation maps; on my desktop, 100 rs-FMRI datasets from the FCON1000 collection ($\approx$69 K voxels and 170 TRs per dataset: 1.2 Gbytes total) can be correlated, *t*-tested, and transmitted to AFNI for dis-play in about 0.4 s per seed voxel click. (CPU-intensive AFNI programs, such as group *InstaCorr*, are generally parallelized using OpenMP.) My purpose in developing these interactive computational tools is to allow researchers to explore the structure of their data, to aid in forming hy-potheses, and to help identify problems such as scanner artifacts (Jo et al., 2010). Fig. 1 shows a screen snapshot of the individual subject *InstaCorr* setup and results windows.

We provide support for non-NIH AFNI users via an active forum at our Web site http://afni.nimh.nih.gov, including the ability for users to upload datasets if needed to help us understand their issues. (Such datasets go to a directory that is not accessible except to the AFNI group, and are deleted when no longer needed.) We have no particular release schedule or version numbering for AFNI. When we add new features to the software, an updated package is put up on our Web site. No significant software is bug-free, but we pride our-selves on fixing reported problems with AFNI programs rapidly—within a day or two if at all possible. Bugs that might have widespread impact are further highlighted with a popup message-of-the-day (dy-namically fetched from the AFNI server), explaining the issues and our recommendations.

Our close interaction with users, particularly at the NIH, has been the source of many developments. For example, our EPI-structural registra-tion algorithm (Saad et al., 2009) was developed when a few discerning users complained that mutual information did not do a good job for this type of image alignment. At first we were skeptical, and suggested using the correlation ratio cost function instead; however, when these users uploaded some sample datasets, we played with them and discovered that datasets with high visual quality can still be very hard to register ac-curately—the ventricles did not match no matter what cost function or software package we used. Not liking to admit defeat so easily, I exam-ined the joint histogram of the datasets carefully and then developed a cost function specialized for this particular application. This example il-lustrates the importance of one of my generic rules: *Always look at your data and intermediate results*. Complicated algorithms and software can fail without giving any obvious indication of error. Only by staying close to the data and the processing stream can a researcher expect to get quality results.

The future

One of the most exciting things we are working on now is a set of tools for "atlasing"—to make it easy to incorporate brain atlas datasets into AFNI, to serve as guides for the user, to use in ROI generation, and in more advanced forms of group analysis now under development. AFNI already has a number of atlases built-in, including the San Anto-nio Talairach Daemon (Lancaster et al., 2000) and the Eickhoff–Zilles probabilistic atlas (Eickhoff et al., 2005), but each of these was incor-porated by custom coding and manipulations. With our new soft-ware, we aim to make it easy to build atlas datasets and then transform individual subject coordinates to these spaces automatical-ly and on-the-fly. In our current software, a user can click a button and pop up a report on the overlap of each activation blob with the regions defined in a set of atlases. We are also developing software and protocols to allow AFNI to query remote atlases (e.g., from com-mercial entities such as Elsevier). Our efforts are not limited to human brain mapping, since there are many non-human FMRI studies car-ried out at the NIH and other sites. We are enhancing our nonlinear 3D registration capabilities to go along with this atlasing suite, and are also developing improved methods for tissue segmentation to be used for this and other purposes (Vovk et al., 2011).

Does it make sense for there to continue to be a number of highly overlapping open-source software packages for analysis of FMRI data-sets? Each software package for FMRI is not just a set of tools, but is also a social ecosystem, with its own style of support and types of users. When I started at the NIH in 2001, the NIMH Scientific Director at the time (Bob Desimone) was enthusiastic about the idea of a cen-tral software platform to which all developers would "plug in". A de-cade later, this vision has not been realized, or even seriously attempted. More recently, there have been efforts to develop and popularize FMRI pipeline toolkits which can incorporate standalone programs from a variety of sources. We have found it difficult to in-corporate AFNI programs, which have many user-adjustable parame-ters, into these pipelines; as a result, we have chosen to develop our own customized pipelines, such as *afni_proc.py* and *align_epi_anat.py*.

At present, there is little evidence of convergence on any higher level of interoperability than the NIfTI-1 and GIfTI file formats. From the viewpoint of developers, this situation makes sense, since agree-ing and adapting to some centralized system would be a lot of effort with no self-evident payoff. From the viewpoint of advanced users, cleaner interoperability is desirable; however, the science that we subserve should not come to depend on a single software source—failures (gross or subtle) in such a system could easily contaminate the entire FMRI-based literature for a long time. Multiple packages can avoid this problem, and also provide a competitive prod for the developers to do better. For these reasons, I believe that instead of trying to unify the major FMRI software packages, a useful intermedi-ate step would be a common input format for specifying a task-based FMRI analysis; this format would make it simpler to run the same data through several toolkits and determine the sensitivity of the

results to the myriad assumptions underlying the packages' algorithms. A similar idea is the definition of a similar platform-independent way to specify a network analysis on pre-specified regions, either of task-based FMRI or rs-FMRI datasets.

Going forward, what is the place of AFNI in the functional neuroimaging world? AFNI is best for researchers who want software that preserves the intermediate results and is transparent about its processing steps, who want the ability to easily craft their own analysis streams, who want flexibility in the display and thresholding of their results, and who want the rapid responses that our team endeavors to give to feature requests, to bug reports, and to calls for help on our forum. In turn, we are ready to cooperate and collaborate with such users to extend and expand the capabilities of AFNI. The AFNI core team at the NIH comprises people who have worked with FMRI data, methods, statistics, and software development for years, and who are ready to continue innovating and helping.

References

Bandettini, P.A., Wong, E.C., Hinks, R.S., Tikofsky, R.S., Hyde, J.S., 1992. Time course EPI of human brain function during task activation. Magn. Reson. Med. 25, 390–397.

Bandettini, P.A., Jesmanowicz, A., Wong, E.C., Hyde, J.S., 1993. Processing strategies for time-course data sets in functional MRI of the human brain. Magn. Reson. Med. 30, 161–173.

Biswal, B.B., Mennes, M., Zuo, X.N., Gohel, S., Kelly, C., Smith, S.M., Beckmann, C.F., Adelstein, J.S., Buckner, R.L., Colcombe, S., Dogonowski, A.M., Ernst, M., Fair, D., Hampson, M., Hoptman, M.J., Hyde, J.S., Kiviniemi, V.J., Kotter, R., Li, S.J., Lin, C.P., Lowe, M.J., Mackay, C., Madden, D.J., Madsen, K.H., Margulies, D.S., Mayberg, H.S., McMahon, K., Monk, C.S., Mostofsky, S.H., Nagel, B.J., Pekar, J.J., Peltier, S.J., Petersen, S.E., Riedl, V., Rombouts, S.A., Rypma, B., Schlaggar, B.L., Schmidt, S., Seidler, R.D., Siegle, G.J., Sorg, C., Teng, G.J., Veijola, J., Villringer, A., Walter, M., Wang, L., Weng, X.C., Whitfield-Gabrieli, S., Williamson, P., Windischberger, C., Zang, Y.F., Zhang, H.Y., Castellanos, F.X., Milham, M.P., 2010. Toward discovery science of human brain function. Proc. Natl. Acad. Sci. U. S. A. 107, 4734–4739.

Cox, R.W., 1996. AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. Comput. Biomed. Res. 29, 162–173.

Cox, R.W., Jesmanowicz, A., 1999. Real-time 3D image registration for functional MRI. Magn. Reson. Med. 42, 1014–1018.

Cox, R.W., Jesmanowicz, A., Hyde, J.S., 1995. Real-time functional magnetic resonance imaging. Magn. Reson. Med. 33, 230–236.

Cox, R.W., Ashburner, J., Breman, H., Fissell, K., Haselgrove, C., Holmes, C.J., Lancaster, J.L., Rex, D.E., Smith, S.M., Woodward, J.B., Strother, S.C., 2004. A (sort of) new image data format standard: NiFTI-1. 10th Annual Meeting of the Organization for Human Brain Mapping (OHBM 2004), Budapest. http://nifti.nimh.nih.gov/nifti-1/documentation/hbm_nifti_2004.pdf.

Eickhoff, S.B., Stephan, K.E., Mohlberg, H., Grefkes, C., Fink, G.R., Amunts, K., Zilles, K., 2005. A new SPM toolbox for combining probabilistic cytoarchitectonic maps and functional imaging data. NeuroImage 25, 1325–1335.

Jo, H.J., Saad, Z.S., Simmons, W.K., Milbury, L.A., Cox, R.W., 2010. Mapping sources of correlation in resting state FMRI, with artifact detection and removal. NeuroImage 52, 571–582.

LaConte, S., Strother, S., Cherkassky, V., Anderson, J., Hu, X., 2005. Support vector machines for temporal classification of block design FMRI data. NeuroImage 26, 317–329.

Lancaster, J.L., Woldorff, M.G., Parsons, L.M., Liotti, M., Freitas, C.S., Rainey, L., Kochunov, P.V., Nickerson, D., Mikiten, S.A., Fox, P.T., 2000. Automated Talairach atlas labels for functional brain mapping. Hum. Brain Mapp. 10, 120–131.

Z.S. Saad, R.C. Reynolds, 2012. SUMA. NeuroImage, this issue.

Saad, Z.S., Glen, D.R., Chen, G., Beauchamp, M.S., Desai, R., Cox, R.W., 2009. A new method for improving functional-to-structural MRI alignment using local Pearson correlation. NeuroImage 44, 839–848.

Talairach, J., Tournoux, P., 1988. Co-planar stereotaxic atlas of the human brain: 3-dimensional proportional system: an approach to cerebral imaging (M. Rayport, Trans.). Thieme, New York.

Vovk, A., Cox, R.W., Stare, J., Suput, D., Saad, Z.S., 2011. Segmentation priors from local image properties: without using bias field correction, location-based templates, or registration. NeuroImage 55, 142–152.