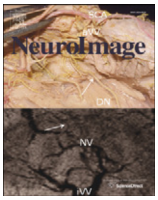
NeuroImage xxx (2011) xxx–xxx

Contents lists available at SciVerse ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg



Review

FSL

Q1 Mark Jenkinson *, Christian F. Beckmann, Timothy E.J. Behrens, Mark W. Woolrich, Stephen M. Smith

Q2 *FMRIB Centre, Nuffield Department of Clinical Neurosciences, University of Oxford,* UK

ARTICLE INFO

Article history:
Received 3 September 2011
Accepted 8 September 2011
Available online xxxx

Keywords:
FSL
Software

ABSTRACT

FSL (the FMRIB Software Library) is a comprehensive library of analysis tools for functional, structural and diffusion MRI brain imaging data, written mainly by members of the Analysis Group, FMRIB, Oxford. For this NeuroImage special issue on "20 years of fMRI" we have been asked to write about the history, developments and current status of FSL. We also include some descriptions of parts of FSL that are not well covered in the existing literature. We hope that some of this content might be of interest to users of FSL, and also maybe to new research groups considering creating, releasing and supporting new software packages for brain image analysis.

Contents

A brief history of FSL

It all began in 1998 when the FMRIB Centre was officially opened, bringing Functional MRI to Oxford. Not only was the lab itself young, but so were the researchers in it, with almost everyone, and certainly all those in the Analysis Group, under 30; sadly, this is no longer true! We were all academic children, or grand-children, of Mike Brady, who was an instrumental player in setting up FMRIB alongside the original founders: Alan Cowey, George Radda and John Newsom-Davis. FMRIB consisted of three groups — Analysis (headed by Steve Smith), Physics (Peter Jezzard) and neuroscience Applications (Irene Tracey). The FMRIB Director was Paul Matthews, with Irene taking over as Director in 2005. Everyone was squashed into the same building, allowing us to interact in seminars, over coffee, at each other's desks, and down the pub. These inter-disciplinary interactions were, and are, crucial for much of the success of the lab.

Q3 * Corresponding author. Fax: +44 1865 222717.

1053-8119/$ – see front matter © 2011 Published by Elsevier Inc.
doi:10.1016/j.neuroimage.2011.09.015

Please cite this article as: Jenkinson, M., et al., FSL, NeuroImage (2011), doi:10.1016/j.neuroimage.2011.09.015

The enduring core of the Analysis Group from the earliest days has been the five "boys" (as we were known). We were all quite junior, this being Steve's second postdoc position, MJ's first, and with Tim, Christian and Wooly all being DPhil students. Steve wants to make it clear that although he might have provided the occasional vague idea about which area each of the boys might work in, he provided virtually no useful supervision to their research, being more ignorant about medical imaging, statistics and general mathematics than any of the others — hence, for example, Christian's description of his own DPhil[1] as being "an exercise in unsupervised learning". So each member of the fledgling Analysis Group pursued an individual research topic: MJ (registration); Wooly (model-based FMRI[2] statistics); Christian (model-free FMRI using ICA); Tim (diffusion tractography); Yongyue Zhang (tissue segmentation); Peter Bannister (head motion correction); Steve (miscellaneous things, including brain extraction and atrophy). There was no overarching strategy besides the goal of developing better methodology for use within the lab, but, of course, the stated purpose of the group was to support the rest of the FMRIB lab in the analysis of their data.

There was a good deal of debate regarding fundamental questions such as "theory vs. results" – would you rather be known for doing novel and interesting theoretical work, or for developing methods that produced the best results in practice – often assuming you couldn't optimise for both. Our views varied widely on this issue, with some more motivated by theory and others more by results. This is an important question as it has significant implications for both the academic output of the group and on its ability to produce and maintain a software package. A purely theoretical bent tends to lead to novel papers and good engineering-oriented grant funding, but is unlikely to lead to general, robust, and well supported software tools. On the other hand, a strongly results-oriented approach tends to lead to good software but is often less good at producing novel papers or getting talks at conferences, and can have less opportunities for funding. Overall we were able to pursue both theory and results because we had a critical mass of people that "spanned this space", rather than each and every one of us having to compromise and live in the middle-ground. One side-effect of distributing and supporting a (fairly) self-complete software package is that we needed to create practical algorithms that (mostly) work robustly on a wide variety of real data; but, at the same time, we have also strived extremely hard to generate leading-edge mathematical theory to feed into our software. It is likely that things would have turned out very differently without this range of people, topics and motivation.

Early software

How did people in FMRIB analyse data before FSL? What is the lineage of FSL? The answers to these questions start with MEDx, whose influence still lingers on, for good and bad. MEDx had already been identified as the primary software that FMRIB would use, even before the lab had been built, and for reasons that none of us ever knew! (In those early days, before most of us joined the lab, there was little or no general discussion about other alternatives, like using SPM instead, although a version of SPM was available in MEDx; and there was certainly no assumption early on that we would produce our own software package). MEDx (see the article by Geoff Aguirre in this issue) was commercial software that integrated display and processing via a single GUI (Graphical User Interface), and could be easily extended with plugins. It was this ability to very easily incorporate our plugins that was initially attractive, and is responsible for us using TCL/TK (for scripting and GUIs), as this was the scripting language required for these plugins — although for us these quickly became just wrappers for our C and C++ ~~programmes~~.

Our research work took the form of implementing algorithms in C/C++, or sometimes directly in TCL, and then linking these into MEDx once we thought they were working well. These modules were then "released" to the rest of the lab and had various people from the different groups use them and give feedback. This turned out to be a very two-way street, as not only would others in the lab get to use our algorithms early on, but we would learn a lot from how they would perform on a wide range of data, as well as how people could misinterpret or misuse what we wrote. It should not be underestimated how much we benefitted, and still do, from this process, as there were often considerable changes made to the algorithms, in order to make them work in a reliable and robust way. The range of different datasets is crucial here. We certainly feel that software written without this immediate feedback about when it does and does not work, is highly likely to fail on many datasets when released to the community at large, as people are much less likely to give good feedback if they don't know you and can't just wander down the corridor, or chat to you in the pub. People outside a given lab will often give up on using tools from that lab without ever contacting the authors. In addition, we also learnt that people using the tools can't actually read the minds of the people writing the tools, and that if we don't explain when and how to use the tools properly, then they don't get used well.

One of the strengths of MEDx was that the integration of *running* the analysis and *visualising* it encouraged the users to *look at their data* throughout the different stages of the analysis pipeline, which is something that we feel is *really important*. As FSL eventually moved away from MEDx and became stand-alone, we have to an extent lost some of that mid-analysis "data-interactiveness" (primarily in order to allow more advanced users to carry out complete analyses very efficiently), but we have at least tried (e.g., through web-page results reporting, that includes lots of snapshots of intermediate results) to maintain (albeit more "static") visualisation of the intermediate analysis stages, such as the registration results forming part of a FEAT FMRI analysis.

Our own software

So, within a year or two, we got to the stage where we had a set of our own tools that performed most of the steps in the FMRI analysis pipeline (brain extraction, smoothing, statistics, registration). At some point (no-one quite remembers), we decided to release the software, as MEDx plugins, to the outside world, not knowing how many other users beyond our lab would have any interest. We thought that if we were starting to develop some tools of novel functionality and scientific value, then it would be very rewarding[3] if other labs started using them widely. It has been! Q4

It was around the time of our first public releases of FSL[4] that we realised that there were relatively few steps in the pipeline where we relied on MEDx, and that we could implement these ourselves and obtain a complete, self-contained analysis pipeline for FMRI. ~~Since~~ it was clearer at this time that there existed a reasonable outside interest in the tools, the decision to fully support a package that was independent of MEDx was a more difficult one than for the original release. Even coding the missing tools wasn't a simple decision, as it required us to write, revise and revise again, implementations of certain existing algorithms that took time to do but would not count

---

[1] PhD in Oxford-speak.

[2] We always preferred to capitalise the "F" in FMRI, partly because we didn't want to be prejudiced against "functional", which is a perfectly good word, and also because the lab name "FMRIB" is already unexciting and unpronounceable enough without demeaning its first letter!

[3] This is really just the same motivation as wanting to publish good work, have it highly cited, and help push science forwards.

[4] Initially these were late-night and poorly coordinated efforts; e.g., at one point Christian was doing some last-minute bug fixes (at 3am) in his code only to discover that it had been released "into the wild" prematurely (by an over-enthusiastic Steve) and that an industrious user had already downloaded, installed and tested it, and was now reporting the bugs that Christian had fixed minutes ago!

Please cite this article as: Jenkinson, M., et al., FSL, NeuroImage (2011), doi:10.1016/j.neuroimage.2011.09.015

towards publications or grants — at least not directly or immediately. This effort was over and above the inevitable efforts that go into writing underlying *library* code (for images and mathematical operations), and often implementing these additional tools was intellectually more demanding and required more careful and intensive testing. A particular example of this was the implementation of cluster-based inference using Gaussian Random Field Theory (the computationally efficient thresholding approach originally developed for SPM). It was nothing more than an implementation of published methods for smoothness estimation and the random field theory mathematics, but required time and care. Looking back on it now, although this was a difficult decision, it was a worthwhile investment, being necessary to reach the wider community, in accordance with the results-oriented drive to develop methods that were not only of intellectual merit but were used in real analyses of real data from real experiments.

Other matters

Once the various missing pieces in the FMRI pipeline had been implemented and tested we finally had a complete, stand-alone piece of software. Then came some really important and difficult decisions: (i) what colour should the GUIs be? (ii) what name do we use? (iii) how do we pronounce the name? and (iv) what logo should we have? OK, so these may not seem quite as important as what statistical thresholding technique should we implement, but these decisions definitely took longer to come to an agreement on, and in some cases there is still no agreement.

We followed the lead of Henry Ford when making the GUIs and decided that the users could have them in any colour they liked, as long as it was grey. However, if you modify the file `$FSLDIR/tcl/fslstart.tcl` then you can have whatever colours you like (but don't tell anyone else or it will spoil the secret). It may not help your analysis, but at least it's a change of scenery.

The next vexed issue was that of a name. A consensus on this was relatively quickly reached and the result was, of course, FSL (standing for “FMRIB Software Library”, and not the GNU-inspired “FSL Software Library”, as a certain recursionist in the group preferred). Following this there was a protracted debate about how it should be pronounced. One camp favoured “Fossil” and one favoured “Eff Ess Ell”. At the same time a logo was developed, based on a trilobite (see Fig. 1), which brought together two things: the love of grey and the pronunciation “Fossil”. As time passed the logo was accepted but the pronunciation chosen was “Eff Ess Ell” — so much for consistency.

Fig. 1. The FSL logo in all its glory, courtesy of “Little Dave” Homfray.

Some philosophy behind the software

We have discussed some aspects of the way that we program but not why we made these decisions. For instance, why did we choose to use C++ and scripts within a Unix-like environment? Although there are many factors, including the fact that we were already familiar with these, the principal reasons were speed, modularity, and portability. We wanted the software to be accessible to all, run quickly, and be powerful, by allowing maximum flexibility and adaptability.

It was (and arguably still is) the case that C++ offers the best combination of speed and portability for programming. There are compilers for all platforms and it produces fast running code with pretty low memory overheads when written carefully. What it is not good at is string and file manipulation, but these things are handled very well by Unix shell scripts, and so this was the combination we went for. It was the case that most scientific labs, including ours, were based around Unix machines (many different variants were common back then such as SGI, Sun, Dec Alpha), and we wanted the code to be usable on all of them, making C++/Unix the best way to have code that worked well in the majority of labs without requiring additional commercial software such as MATLAB. It also pushed us to adopt `/bin/sh` for the shell scripts rather than `bash`, as the latter's syntax varied too much over different platforms (and, unfortunately, still does). The only other major platform was Microsoft Windows and it was initially quite easy for us to get FSL to run natively on Windows by using the freeware “Cygwin” program. Alas, this is no longer a possibility due to changes in Cygwin over the years, and we now skirt the issue completely by recommending that people run FSL in a virtual Linux machine within Windows.

Apart from issues of speed and portability, *modularity* was an important factor for the science and usability. Part of the Unix philosophy that we liked was the fact that it is built from small components (tools/~~programmes~~/executables/scripts — call them what you like) that do individual jobs but can be put together in a very large variety of ways to accomplish a huge range of tasks. This was exactly what we wanted to be able to do with our tools, and so this was the approach we took. It can be seen most strongly with tools such as `fslmaths`[5] and `fslstats` that, just on their own, can be put together to do all sorts of things. It was also this desire to be modular, flexible and hence powerful, that drove the decision to separate the processing from the GUIs and viewing tools. The GUIs provide a default and user-friendly pipeline, that simply organises the execution of the individual command line tools. Similarly, `FSLView` is only a tool for viewing and interacting with images, but not for modifying them, besides manual drawing. In fact, to start with we had no viewing tool within FSL at all and relied totally on other packages for this, such as MEDx and AFNI (which we used, with great gratitude to Bob Cox, in one or two early FSL courses). Q5

Another major tenet of the philosophy of FSL was that it should not require a master's degree in computer science and the patience of a saint to install it. We learnt a lot from another major brain imaging package that we had worked with, where it was a long and painful process to install all the necessary third-party packages (and then the other packages that *they* depended on, etc.) and then get the compilation working. At the time the Unix world did accept this as more or less normal, which has thankfully changed. Hence for FSL, we always tried to provide pre-compiled self-contained downloads for the most common operating systems, but always had source code available for those who needed to compile it for more unusual systems (or just because they secretly liked running the compiler…).

---

[5] American friends might like to: `cp $FSLDIR/bin/fslmaths $FSLDIR/bin/fslmath`.

Please cite this article as: Jenkinson, M., et al., FSL, NeuroImage (2011), doi:10.1016/j.neuroimage.2011.09.015

Names

Putting together a complete software package also brought unexpected challenges, like what names to give to each tool. It might have been easier (on users' hippocampi) to follow the SPM lead and go with generic names (FSL-segment, FSL-register, etc.), but, we didn't, and can't quite remember why, or even whether we ever really discussed it... We spent a disproportionate amount of time discussing the names; and for those who are unfamiliar with FSL, it is a haven for acronyms: BET, FLIRT, FEAT, FILM, PRELUDE & FUGUE, MCFLIRT, FLAME, FNIRT, MELODIC, FAST, SIENA, FDT, BEDPOST, TBSS, and the list goes on. Most of these, we must admit, do not stand for something that is easy to remember, but they are all easy to pronounce... except for TBSS and FNIRT.

Although there are some quite good acronyms in the above list (if we do say so ourselves, although we admit they are not the best ever; Yang et al., 2007) there are some acronyms that have never properly seen the light of day. For example, part of the FLAME process is involved in approximating Bayesian probability distributions or, more simply speaking, cleaning up the posterior, and is accurately described by: *B*ayesian *I*nference with *D*istribution *E*stimation using a *T*-fit. Another example, long forgotten, is a symmetry estimation tool, for finding the inter-hemispheric plane, prior to the days of highly reliable registration, called: *A*utomatic *R*obust *S*ymmetry *E*xtraction. A final example is a near miss, rather than a hidden or forgotten one, and that is the original name for "FNIRT": *F*MRIB's *O*ptimised *N*onlinear *D*eformation tool with *L*evenberg–marquardt *E*stimation (*FONDLE*), which naturally follows after FLIRTing with the data!

Programming

Although certain aspects of programming can be enjoyable (such as folding mode, virtual functions, and using the word "grot" as much as possible) it is really the research that drives us on. As it is not possible to do as much of both as we would like, a good balance is needed between planning, coding, testing, documenting and the non-software-related activities of methodology research, publications, grants, etc. Although we have found no perfect solution to this problem, we have adopted, on the programming side, a mixture of third-party libraries for standard functionality (such as matrix mathematics via the NEWMAT package which we have very happily used now for many years now — thank you Robert Davies!) and some general in-house libraries for image- and timeseries-related storage and processing. Most of the code in FSL is specific application code (e.g., FEAT or FDT or SIENA) that sits on top of these libraries. Non-research-related coding (e.g., FSLView, randomise, etc.) and more extensive testing is something that we have only relatively recently managed to fund a pure (and non-virtual) programming position for, due to the difficulty of getting funding for non-research staff.

Whilst we are in the programming section we have two confessions we ought to make. The first is that we know that some of the error messages from FSL are about as easy to understand as hieroglyphics. Although it is possible to make this better, finding all the instances would require a lot of time that we have been investing instead in new methodology, support and documentation. The second confession is that we have a love affair with Macs.[6] We won't try and justify this in a gushing display of public affection, but instead will explain how we first met. It all began with a very generous gesture from Robert Coghill who wanted a port of FSL to run on his Mac very badly. However, we didn't know how the new Macs worked and so he solved that by just sending us a laptop, free of charge. This was the first time that we had seen the new range of Macs, running the unix-style OSX. In fact the port of FSL to Mac was finished the day after the laptop arrived. It was just so much easier to set-up and maintain than the linux laptops we were used to — it was lovely and a thing of beauty![7] We even got to use PowerPoint, and more recently Keynote, without giving up on using Emacs and the command line in a terminal — something that I don't think we could live without. Within 2 years the entire FMRIB lab was almost totally dominated by Mac desktops and laptops.

Licensing

Universities at present push for all research outputs to be commercially marketed when possible, and the University of Oxford is no exception. We have been very grateful that the University has taken our wishes on these issues onboard, and we have adopted a nearly freeware strategy so that only profit-making use of FSL, such as work by or for pharmaceutical companies, requires a paid licence, with pure academic enterprises using it for free. This has led to many more people making good use of FSL than would otherwise have been the case. The researchers who create FSL receive a fraction of the income generated, which is nice (if modest), but we don't consider this to be a "conflict of interest" in terms of wanting people to use FSL, because even without money involved we would be enthusiastic about them doing so! We are committed to continuing to provide the main FSL software package freely to academia.

Past to present

It was June 2000 when we first released FSL, which at that time only contained a very small subset of the tools it contains today. Over time we have developed new methodology, and improved existing methodology, but also put significant efforts into support, documentation, publications and data formats. It now contains about 200,000 lines of code!

Software structure

FSL divides into three main areas, related to functional, diffusion and structural image ~~analyses~~. There are over 230 individual command line tools (approximately 140 scripts and 90 compiled C++ ~~programmes~~ — including 50 small/flexible tools in the "fslutils" set) plus 23 GUIs, making it very flexible but rather formidable to the first-time user. However, there are only a handful of major tools that most people use directly, as shown in Table 1, which gives a rough idea of the current scope of FSL.

As mentioned above, the GUIs provide a simple interface and pipeline for the various underlying command-line tools. It is always possible to replicate an analysis that was done with a GUI using only command-line tools, and this is made easier by the existence of a command log file that is output by the GUI. This makes custom-scripting relatively easy, including parallelising tasks on a computing cluster, although distributing jobs over a cluster is already automatically handled by some of the "larger" FSL GUIs (e.g., FEAT, FDT).

Some of these tools have existed for a long time (e.g., FEAT, BET, FLIRT) while others are relatively new (e.g., FABBER, TBSS, FSLVBM). At times there have been "delays" before including certain functionality that might arguably have appeared earlier, for example nonlinear registration and VBM-like functionality; we now describe a little of the relevant history.

Initially FSL contained only linear registration and our belief was, based on our experience of using different tools, that (for FMRI), using a robust linear registration method was superior in consistency to relatively low degree-of-freedom (DOF) non-linear registration alternatives.

---

[6] We have no financial links with Apple, although we encourage Apple executives to contact us at the above address!

[7] Sorry, we gushed...

Table 1
The major tools within FSL.

Use	Tool name
FMRI: task-based, using GLM	FEAT
FMRI: resting-state or task-based, using ICA and no temporal model	MELODIC
ASL (perfusion imaging of flow)	FABBER
Diffusion: probabilistic tractography	FDT
Diffusion: multisubject voxelwise analysis	TBSS
Brain extraction	BET
Tissue-type segmentation (GM/WM/CSF)	FAST
Subcortical segmentation	FIRST
Linear and Non-Linear Registration	FLIRT and FNIRT
Voxel-wise analysis of grey matter density	FSL-VBM
Whole brain atrophy (longitudinal and cross-sectional)	SIENA and SIENAX

This was especially true when the data quality from the scanners was relatively poor by today's standards, and the only available standard-space template was the linearly-registered-and-averaged MNI152 (or avg305 before that). Since the template image had relatively little detail in the cortical folds and distortions in the functional EPI scans was substantial, then there seemed little advantage in using non-linear registration. A more consistent linear registration, which was less affected by artefacts, should produce better results, although there would be some systematic mis-locations of activations. However, other developments, such as registering diffusion-based skeletons in TBSS (for which we initially used the IRTK non-linear registration (Rueckert et al., 1999) for approximate alignment), the improvement in data quality, the ability to correct EPI distortions, and the advances in non-linear registration methodology, meant that we (finally) did develop our own non-linear registration method (FNIRT). This method is capable of very high-DOF registrations between images or to the (now non-linear — thanks Andrew Janke!) standard MNI152 template.

At the outset we were a little skeptical of the usefulness of VBM (Ashburner and Friston, 2000; Good et al., 2001), due to the standard criticisms of non-linear registration error and sub-optimal data and templates, as discussed above. We also believed, and still do, in the potential of surface-based analysis (e.g., using FreeSurfer) to quantify more meaningful physiological parameters such as cortical thickness. However, we received many requests for a VBM tool and had several discussions with internal and external users about the relative strengths and weaknesses of the two methods (VBM and surface-based cortical thickness analysis). We also conducted our own comparison study (Voets et al., 2008), showing that there were both similarities and differences in the biologically-plausible results generated, making it impossible to tell, without knowing the ground truth, which method was superior. The upshot was that it became clear that there was role for VBM within FSL as it has high sensitivity for change, even though the quantity being measured is hard to interpret and the user must be very careful to minimise segmentation and registration errors. Consequently we have a VBM implementation, FSL-VBM, which is routinely used for looking at grey matter changes, although white matter *microstructure* change is dealt with entirely differently — using TBSS.

Publications and tools

Most of the major tools in FSL have accompanying papers (including the overviews Smith et al., 2004; Woolrich et al., 2009) that explain the scientific principles behind the software and show some of the validation and test results. However, not all pieces of work are suited to publication, as some tools are just implementations of existing ideas (e.g., FUGUE). Thus the coverage of the FSL toolset in the literature it is a little patchy. In order to partially rectify this we will now discuss some of the tools which are not supported by specific publications.

FUGUE

One tool in this category is FUGUE, which is a tool for performing EPI distortion correction based on an acquired fieldmap. The principles are derived from a combination of work in Jezzard and Balaban (1995) and Cusack et al. (2003). That is, it takes an undistorted fieldmap (acquired using two gradient-echo or spin-echo scans with different TEs) and scales the values to calculate the voxel-wise shift (a non-linear spatial distortion) along the phase encode direction (Jezzard and Balaban, 1995). Furthermore, it calculates an estimate of the signal loss (caused by dephasing of spins within a voxel induced by the B_0-inhomogeneities) which is used as a weighting function for registration of the undistorted EPI. Getting the fieldmap and EPI into correct alignment, to account for any motion that occurred between the acquisitions, is done by creating a distorted version of the fieldmap magnitude image as a registration target for the EPI (Cusack et al., 2003). The remainder of the required functionality is implemented with our existing tools, such as FLIRT for the linear registrations, and the overall pipeline for distortion correction is built into our general FMRI analysis tool, FEAT.

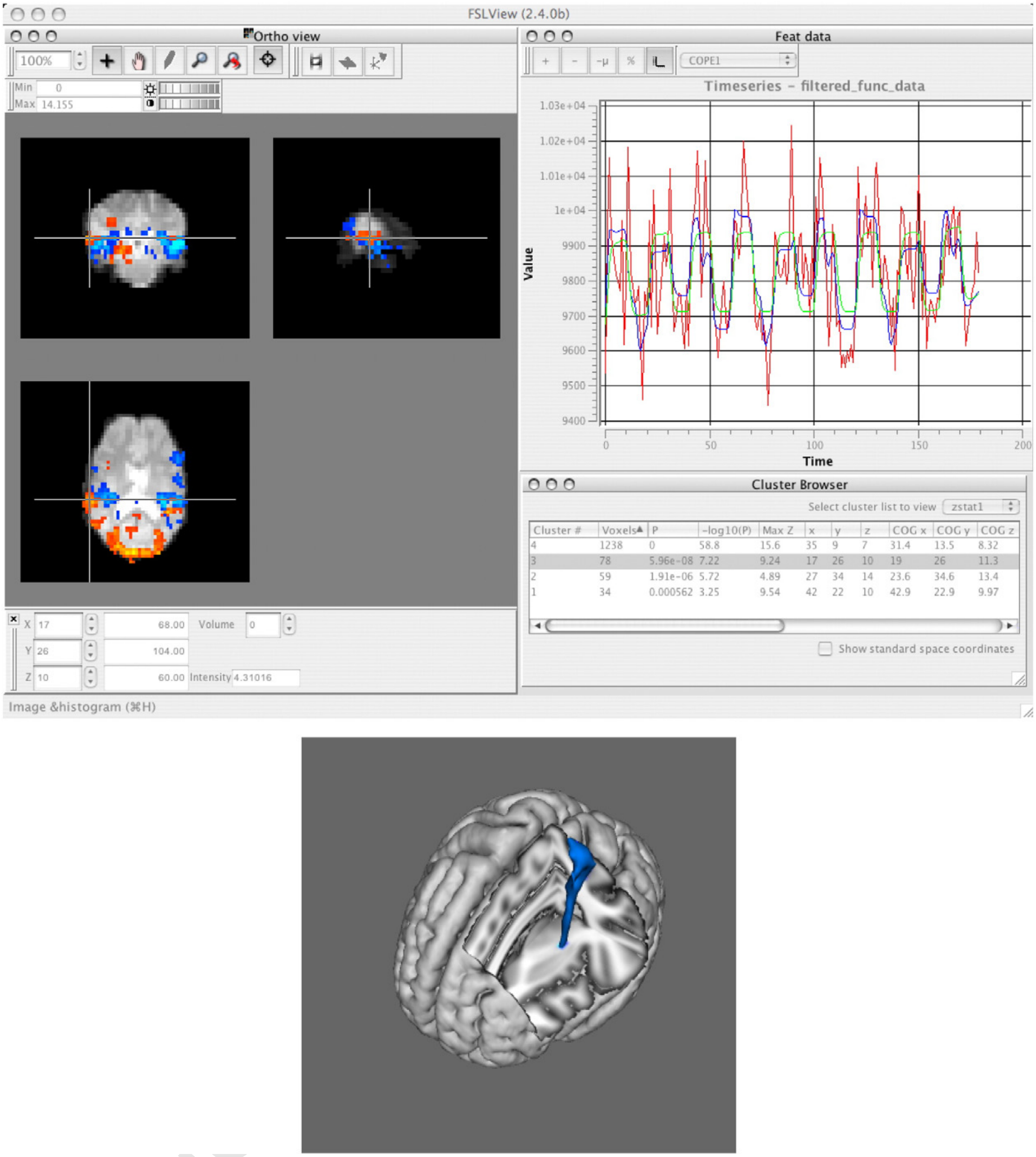
Inference

A more general area for FSL that has patchy coverage in the literature is thresholding and inference. As mentioned earlier, we started with an initial implementation of Gaussian Random Field Theory (Friston et al., 1994; Worsley et al., 1992) for cluster-based thresholding and inference. This is computationally very fast, and the use of cluster extent as the test statistic can have considerable sensitivity advantages over voxelwise testing. It is still the most commonly used thresholding and inference method applied in FMRI analysis within FSL. Since then we have implemented other thresholding and inference methods such as spatial mixture modelling (Woolrich et al., 2005), FDR (Benjamini and Hochberg, 1995) and `randomise` (permutation-based inference; Nichols and Holmes, 2001). In situations where parametric (i.e., Gaussian) assumptions about the data no longer hold, such as in FSLVBM or TBSS, we use `randomise`. The choice does not stop here though, as randomise is capable of producing either uncorrected voxel-wise p-values or a variety of p-values that are corrected for multiple comparisons. For instance, it can reproduce the non-parametric equivalent to the cluster-based method in random field theory, or it can ~~weigh~~ the cluster by the statistic values (cluster-mass thresholding), or use TFCE-based values (Smith and Nichols, 2009). As this form of non-parametric analysis is based on very few assumptions, especially compared to random field theory, we foresee that these techniques will be used more and more in the future. This is already happening within our lab, especially when more complicated analyses are being done, e.g., using ICA-based de-noising, voxel-wise regressors, and corrections for non-stationarity (Salimi-Khorshidi et al., 2011).

Display tools

Another topic that is difficult to publish on is the development of display or viewing tools; although they are important for the software, they rarely represent new science. As we said before, FSLView was not created immediately and FSL existed for several years without it. However, a good viewing tool is crucial as it is the way that you can *look at your data*, which is an absolutely essential activity for everyone. It is only by inspecting raw data and the different stages of results in an analysis pipeline thoroughly that you can have confidence in the final outcomes and interpretations. With FSLView, heroically honed over the years by Dave Flitney, we were able to more easily interact with various images from FEAT (functional analysis) outputs and get closer to the data and results — see Fig. 2. We always have more ideas about desirable functionality in the viewer than we have time and resources to pursue, but we are hoping to continue to improve the ability of FSLView to get closer to the data and results,

Please cite this article as: Jenkinson, M., et al., FSL, NeuroImage (2011), doi:10.1016/j.neuroimage.2011.09.015



Min 0
Max 14.155
COPE1

X 17 68.00
Y 26 104.00
Z 10 60.00
Volume 0
Intensity 4.31016

Select cluster list to view zstat1

Cluster #	Voxels▲	P	-log10(P)	Max Z	x	y	z	COG x	COG y	COG z
4	1238	0	58.8	15.6	35	9	7	31.4	13.5	8.32
3	78	5.96e-08	7.22	9.24	17	26	10	19	26	11.3
2	59	1.91e-06	5.72	4.89	27	34	14	23.6	34.6	13.4
1	34	0.000562	3.25	9.54	42	22	10	42.9	22.9	9.97

☐ Show standard space coordinates

Fig. 2. Top panel: screenshot of FSLView showing two overlayed activation maps (coloured in reds and blues) together with raw time-series, GLM model fits and clustering results. Bottom Panel: screenshot of a 3D rendering of diffusion-based tractography results in FSLView. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

implementing the ability to view MELODIC (ICA) and FDT (tractography and connectivity) outputs in a faster, more intuitive manner. Watch this space.

In addition to viewing data and results, FSLView also has several built-in atlases, some of which are bundled with permission from external efforts (MNI structural atlas, Jülich histological atlas, Talairach Daemon labels, JHU white-matter atlases, Cerebellar atlas) and some of which were developed in-house (Harvard–Oxford structural atlases, using manual labels kindly (and painfully!) provided by David Kennedy, and the Oxford Thalamic Connectivity Probability Atlas). These allow users to get a feeling for population averages/variation at specific points in the brain, as well as where the "standard" regional boundaries would be in their subject or group average. It is important to stress that this does not, and should never, replace the user looking carefully at their data and using their own understanding of the anatomy to interpret their results. However, it can add extra information on top of this, especially when considering how the population varies and how different modalities (e.g., histology from the Jülich atlas) compare with the visible gross anatomy.

Teaching and documentation

Providing teaching material is crucial for any package, FSL included. There was always some documentation provided on the FSL website and bundled with the package itself. However, following the OHBM conference in 2001, we took the next steps and started providing more personal support via an email list and through an annual hands-on taught course. In the near future we will be replacing all of the documentation with a Wiki, combining the current web documentation, FAQ, email list search and course material.

Email list

FSL has been supported by an active email list since mid 2001. The initial decision to start this list was, let's say, not universally popular among the members of our group. It is safe to say that within our group the tradeoff between research time and support time has always been fraught with opposing opinions. However, it has certainly been popular with the community (often reported as one of the best features), with an average of about 10 emails per day (over 40,000 in total) since it started.

In practice the email list has allowed us to get some incredibly valuable feedback on how tools perform in situations both similar and very different to how they were used in our lab. The consequence of this is that we have been able to improve the tools and increase our confidence in the accuracy and robustness of the algorithms. Such interactions, although not the commonest types of email on the list, have certainly led to quicker bug fixes and generally have enhanced the quality and even scope of FSL. Often they require looking at datasets which are uploaded, by invitation only, to our secure site, and then deleted afterwards.

FSL and FreeSurfer course

The email support list is useful for people who have specific questions to ask, but does not really get people up to speed on FSL when they are new to it. In 2002 we decided to hold the inaugural FSL course in Melbourne, Australia where we would not only give some lectures, but get people to use the software with hands-on practicals and a maximum of two people per computer (see Fig. 3). Since then we have had one per year, with up to 160 attendees each time, holding them around the time of the OHBM conference and in a city that is “close”. Some examples will probably give you an idea of what we mean by “close” or, alternatively, how good our geography is. For instance, pairs of cities have included: Florence–Siena; Los Angeles–New York; and Sendai–Melbourne.

We have always held the course jointly with the FreeSurfer developers (starting in Melbourne with our good friend Doug “throw the cigar over the wall” Greve) as this is a case of a positive, long-standing relationship due to both the fact that the two packages are very complementary and the people very complimentary. This linkup has also been excellent for advertising the course; in Marina del Rey, Los Angeles, we gained an extra attendee who had seen the sign for the “Free Surfer” course and decided to come along to check it out, surfboard and all dude!

The course has always been very popular with the attendees and so we have left the format pretty much unaltered: a 90 minute lecture followed by a 90 minute hands-on practical (with plenty of demonstrators around to answer questions), repeated each morning and afternoon. It is not without some pain though, and this was never more extreme than with the first course, where the typical day involved teaching the course from 9 to 5, going for dinner, then back to the lab to finish creating the lectures and practicals for the next day (such as the “FSL” activation — see Fig. 3b), until about 3 am. One morning we were even fixing bugs in the software during the lecture, up to 15 min before the practical started. Thankfully, since that first time, things become less and less fraught every year!

Data formats

Due to our formative years being dependent on MEDx, we initially started by supporting the Analyze data format. Right at the beginning we decided not to support DICOM as a native format since there already existed several perfectly good conversion tools from DICOM to Analyze, and we were not keen on reinventing that particular wheel. Also, DICOM is an ever-changing target, and converters need to be constantly updated, so we are happy with the decision to leave DICOM conversion to others — many thanks are owed to those who do provide these tools, including Doug Greve (mri_convert in FreeSurfer), Chris Rorden (dcm2nii in mricron) and Jolinda Smith (MRIConvert).

Unfortunately, as many readers will know, Analyze was far from a perfect format and there came to be several distinct and incompatible flavours of it. This problem was recognised by the NIH and a committee was formed, chaired by Stephen Strother and filled with representatives from major functional neuroimaging software packages, in order to thrash out a standard data format that would be understandable and accessible for all and would facilitate exchange of data between packages. The result was, after a couple of years, the NIfTI1 format, and we began supporting this in FSL in 2004. This has been very successful in allowing data to be exchanged between packages

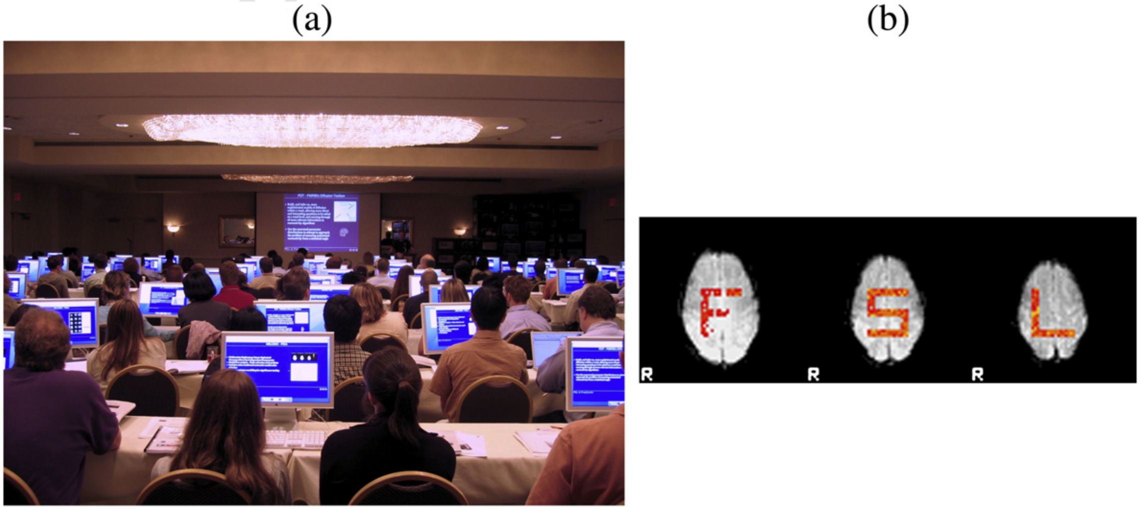


Fig. 3. Panel (a): photo demonstrating the existence of the FSL and FreeSurfer course and the fact that we really do love Macs. Panel (b): the “FSL” brain activation created as a simple (simulated!) example dataset in 2001, so that it could be analysed in less than 10 min on the computers of the day, and that is still going strong and used in our courses to this day.

Please cite this article as: Jenkinson, M., et al., FSL, NeuroImage (2011), doi:10.1016/j.neuroimage.2011.09.015

and therefore increases the scope of what can be done scientifically, since each package has unique capabilities. Recent innovations such as NIPY (`nipy.sourceforge.net`) or CBRAIN (`cbrain.mcgill.ca`) are further increasing the range of possibilities for cross-package analysis ~~(Fig. 4)~~.

A big difference between NIfTI and Analyze was that NIfTI images could store the anatomical orientation of the different axes. Previously, all Analyze images were being treated as the "same way around" as the template images: MNI152, or avg305 before that. These had come, somewhat strangely, with a negative voxel dimension in the x-direction and so we always wrote our images out that way, but realised early on that we could not trust that this would be consistent in the input images — and so we ignored the sign of the voxel dimensions when reading images. We had taken these things on board as strong assumptions and they ran through everything in FSL, so that we then needed to undo them everywhere in order to get things to work for both orders of storage (or handedness, or radiological/neurological storage, all of which roughly boils down to whether the list of intensities in the file starts at the left or the right hand side of the image). At the beginning this caused some problems for users and developers alike, but has now been resolved for several years and FSL can happily cope with either storage order and can tell left from right.

Future directions — XMI

The future for FSL is all about integration. We want to integrate more with tools like FreeSurfer, trying to make interoperation as seamless as possible. We also intend to add support for extra formats/dimensionalities: GIFTI (allowing surfaces to be represented), NIFTI2 (allowing 64-bit dimensions and storage), CIFTI (combining surface and volumetric components of GIFTI and NIFTI together, being developed within the NIH Human Connectome Project).

Scientifically, in the immediate future, much of our research will relate to our part in the Human Connectome Project (HCP); this will combine advanced acquisition techniques for structural, functional and diffusion data with a very large cohort of subjects. We will be working on methods for exploring the connectome based on diffusion data, task FMRI and resting-state FMRI. Much of this will be looking at the relationships between the modalities, which is part of our longer term plans to push forward with much more cross-modal integration (XMI) research. We feel that cross-modality methodology needs to become more integrated at a low-level and be able to simultaneously and coherently analyse data from the different modalities (structure/function/diffusion). One of the biggest strengths of FSL in our view is the ability to analyse images over many different modalities, and a main goal for us is to bring these analyses closer together, both scientifically and in the software. Analysing modalities together, in a truly joint manner, is where we see the future of MRI and analysis research going. An early example of such integration work is the cross-modal ICA work of Groves et al. (2011), where we have developed a Bayesian framework for starting to address some of the major challenges when combining very different types of data in one big analysis.

FSL, as for all packages, has its strengths and weaknesses, fans and critics. It has the capability for great flexibility by virtue of its array of command-line utilities, it can analyse a wide range of MR modalities (task FMRI, resting FMRI, ASL, diffusion, structure), and can be easily scripted and run over computing clusters. It is particularly suitable for multi-modal MRI neuroimaging investigations and will increasingly support a wide-range of connectivity-based analyses.

Acknowledgments

There are many people that have contributed to FSL and our research over the years and we would like to thank them all for what they have done, and apologise to any that we have forgotten. The list of developers (in approximately chronological order) is: YongYue "axe" Zhang, Peter Bannister, Dave Flitney, Heidi Johansen-Berg, Duncan Mortimer, Matthew Webster, Ivana Drobnjak, Rami Niazy, John Vickers, Tom Nichols, Natalie Voets, Jesper Andersson, Brian Patenaude, Gwenaëlle Douaud, Saad Jbabdi, Eugene Duff, Adrian Groves, Michael Chappell, Reza Salimi, Aleksandar Petrović and Stam Sotiropoulos.

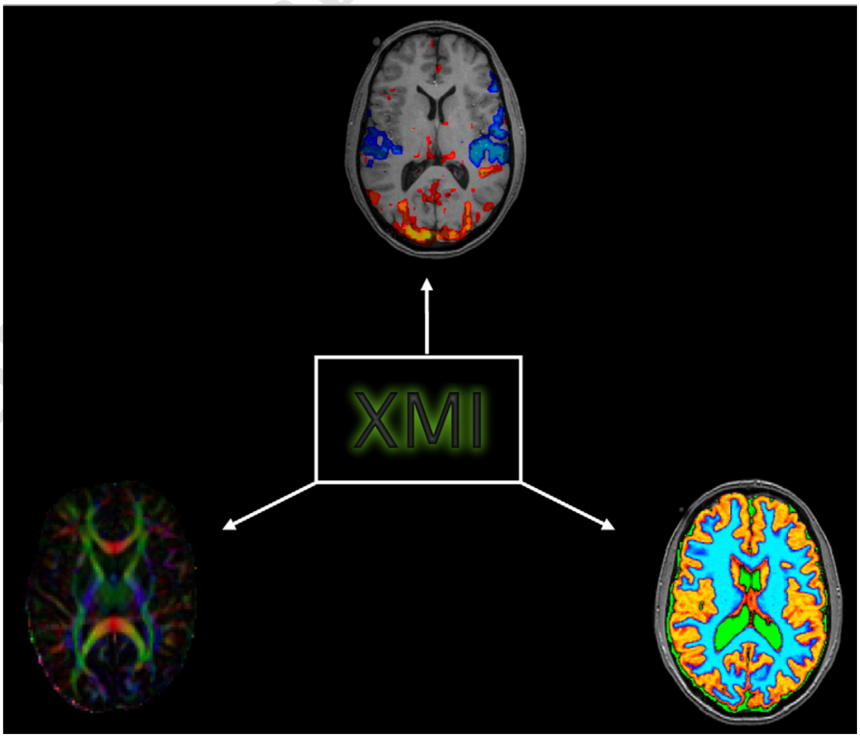


Fig. 4. Our XMI future research plan in detail.

Please cite this article as: Jenkinson, M., et al., FSL, NeuroImage (2011), doi:10.1016/j.neuroimage.2011.09.015

Other collaborators and contributors include: Katrin Amunts, Andreas Bartsch, Marco Battaglini, Mike Brady, Janis Breeze, Jacqueline Chen, Stuart Clare, Mark Cohen, Louis Collins, Ross Cunnington, Joern Diedrichsen, Julien Doyon, Marilena De Luca, Nicola De Stefano, Simon Eickhoff, Alan Evans, Bruce Fischl, Peter Fox, Jean Frazier, David Glahn, Jill Goldstein, Doug Greve, Diana Tordesillas Gutiérrez, Michael Hanke, Peter Hansen, Christian Haselgrove, Morgan Hough, Andrew Janke, Peter Jezzard, David Kennedy, Peter Kochunov, Angela Laird, Jack Lancaster, Didier "it's like S plus" Leibovici, Donna Lloyd, Salima Makni, Jonathan "mmm" Marchini, Paul Matthews, Karla Miller, Susumu Mori, Jeanette Mumford, Bruce Pike, Russ Poldrack, Narender Ramnani, Brian Ripley, Daniel Rueckert, David Salat, James Saunders, Nick Schmansky, Larry Seidman, Andy Stenger, Irene Tracey, Liqun Wang, Richard Wise and Karl Zilles.

And lastly, we would like to give a big thanks to The White Hart for a great place to discuss things over a few beers!

References

Ashburner, J., Friston, K., 2000. Voxel-based morphometry—the methods. Neuroimage 11, 805–821.

Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. Ser. B Methodol. 289–300.

Cusack, R., Brett, M., Osswald, K., 2003. An evaluation of the use of magnetic field maps to undistort echo-planar images. Neuroimage 18, 127–142.

Friston, K., Worsley, K., Frackowiak, R., Mazziotta, J., Evans, A., 1994. Assessing the significance of focal activations using their spatial extent. Hum. Brain Mapp. 1, 214–220.

Good, C., Johnsrude, I., Ashburner, J., Henson, R., Friston, K., Frackowiak, R., 2001. A voxel-based morphometric study of ageing in 465 normal adult human brains. Neuroimage 14, 21–36.

Groves, A., Beckmann, C., Smith, S., Woolrich, M., 2011. Linked independent component analysis for multimodal data fusion. Neuroimage 54, 2198–2217.

Jezzard, P., Balaban, R., 1995. Correction for geometric distortion in echo planar images from B0 field variations. Magn. Reson. Med. 34, 65–73.

Nichols, T.E., Holmes, A.P., 2001. Nonparametric permutation tests for functional neuroimaging: a primer with examples. Hum. Brain Mapp. 15, 1–25.

Rueckert, D., Sonoda, L., Hayes, C., Hill, D., Leach, M., Hawkes, D., 1999. Nonrigid registration using free-form deformations: application to breast MR images. IEEE Trans. Med. Imaging 18, 712–721.

Salimi-Khorshidi, G., Smith, S., Nichols, T., 2011. Adjusting the effect of nonstationarity in cluster-based and TFCE inference. Neuroimage 54, 2006–2019.

Smith, S., Nichols, T., 2009. Threshold-free cluster enhancement: addressing problems of smoothing, threshold dependence and localisation in cluster inference. Neuroimage 44, 83–98.

Smith, S., Jenkinson, M., Woolrich, M., Beckmann, C., Behrens, T., Johansen-Berg, H., Bannister, P., De Luca, M., Drobnjak, I., Flitney, D., Niazy, R., Saunders, J., Vickers, J., Zhang, Y., De Stefano, N., Brady, J., Matthews, P., 2004. Advances in functional and structural MR image analysis and implementation as FSL. Neuroimage 23, 208–219.

Voets, N., Hough, M., Douaud, G., Matthews, P., James, A., Winmill, L., Webster, P., Smith, S., 2008. Evidence for abnormalities of cortical development in adolescent-onset schizophrenia. Neuroimage 43, 665–675.

Woolrich, M., Behrens, T., Beckmann, C., Smith, S., 2005. Mixture models with adaptive spatial regularisation for segmentation with an application to FMRI data. IEEE Trans. Med. Imaging 24, 1–11.

Woolrich, M., Jbabdi, S., Patenaude, B., Chappell, M., Makni, S., Behrens, T., Beckmann, C., Jenkinson, M., Smith, S., 2009. Bayesian analysis of neuroimaging data in FSL. Neuroimage 45, S173–S186.

Worsley, K., Evans, A., Marrett, S., Neelin, P., 1992. A three-dimensional statistical analysis for CBF activation studies in human brain. J. Cereb. Blood Flow Metab. 12, 900–918.

Yang, D., Meng, G., Zhang, S., Hao, Y., An, X., Wei, Q., Ye, M., Zhang, L., 2007. Electrochemical synthesis of metal and semimetal nanotube–nanowire heterojunctions and their electronic transport properties. Chem. Commun. 1733–1735.

Please cite this article as: Jenkinson, M., et al., FSL, NeuroImage (2011), doi:10.1016/j.neuroimage.2011.09.015