



Report from 2015 OHBM Hackathon (HI)

# Wrapping FreeSurfer 6 for use in High-performance Computing Environments

Project URL: <https://github.com/nipy/nipype>

David G. Ellis<sup>2</sup>, Regina E.Y. Kim<sup>3</sup>, Ipek Oguz<sup>4</sup> and Hans J. Johnson<sup>1,3,4\*</sup>

## 1 Introduction

FreeSurfer[1] is a popular software suite for automatic analysis of MRI data, including subcortical and cortical segmentation, as well as cortical surface reconstruction and correspondence. FreeSurfer's most prominent tool, the **recon-all** workflow, consists of approximately 170 sequentially run commands in a **tcsh** shell script that uses approximately 50 unique FreeSurfer tools. The purpose of this project is to reconstruct the **recon-all** workflow from FreeSurfer's **tcsh** shell script into an equivalent workflow using **Nipype**, which *"supports a uniform interface to existing neuroimaging software and facilitates interaction between these packages within a single workflow"*[2].

The goal of this work is to enhance the efficiency and usability of the workflow by allowing it to take advantage of the increasing availability of high performance computing resources. **Nipype** also enhances the modularity of the workflow which allows for algorithms from other packages (e.g., ANTS[3], FSL[4], BRAINSTools[5][6][7]) to be explored, added into the workflow, or take the place of existing processing steps. Therefore, the **Nipype** environment permits increased collaboration on the **recon-all** workflow and allows for the limitations of the workflow to be easily addressed.

## 2 Approach

**Nipype** interfaces were created for the tools used in the **recon-all** workflow. These interfaces allow developers to recreate in a **Nipype** workflow the exact

same commands used in the FreeSurfer's **tcsh** script. The **Nipype** version of the **recon-all** workflow was then created by using the **Nipype** interfaces to connect the FreeSurfer commands in the order necessary. To verify that the new **Nipype** workflow is equivalent to FreeSurfer's **recon-all** workflow, both workflows were run on the same set of MRI images on multiple platforms (CentOS 6.4 and Mac OS X) and in a high-performance computing environment. Output surface files were converted to VTK file format, and the output image files were converted to NIFTI file format. The images and surfaces output from FreeSurfer's **recon-all** workflow were compared to the outputs from **Nipype recon-all** workflow.

## 3 Results

All output images and surfaces from FreeSurfer's **recon-all** were identical to those of the **Nipype** workflow that was run on the same operating system. During testing on a 16 Core CentOS 6.4 machine with 64GB of memory, FreeSurfer's **recon-all** workflow completed processing in over 8.9 hours. With multi-processing, the **Nipype** workflow achieved identical results and completed processing in under 6 hours when tested on the same machine.

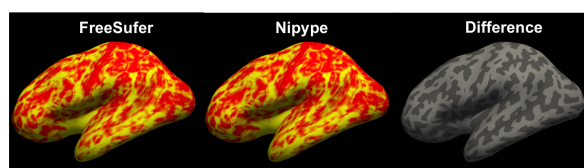
## 4 Conclusions

The **Nipype** workflow created in this project was shown to be equivalent to the pre-existing FreeSurfer **recon-all** workflow. Furthermore, by utilizing **Nipype**'s ability to run commands in parallel, the new workflow reduces the running time of **recon-all** by over 30% when compared to FreeSurfer's **recon-all** script which runs commands in a sequential order. The **Nipype** environment also allows for increased collaboration in further developing the workflow. Future work

\*Correspondence: [hans-johnson@uiowa.edu](mailto:hans-johnson@uiowa.edu)

<sup>1</sup>Electrical and Computer Engineering, The University of Iowa, Iowa City, 4016 Seamans Center for the Engineering Arts and Sciences, 52242, Iowa, USA

Full list of author information is available at the end of the article



**Figure 1** Visualization of the cortical thickness projected onto the left hemisphere's inflated surface for both the FreeSurfer(left) and Nipype(middle) workflows as well as the difference the two thicknesses(right). The thickness measurements showed no difference between the workflows.

will involve incorporating more options to the Nipype workflow so that it can function as a complete replacement for the `tcsh` shell script. The collaborations at the 2015 OHBM BrainHack meeting were instrumental in accomplishing this task. Collaborations with the FreeSurfer, Nipype, and Human Connectome teams allowed members of this project to quickly identify problems and avoid unnecessary failures.

#### Availability of Supporting Data

The resulting workflow from this project can be found at: <https://github.com/nipy/nipype>. An example of how to run the workflow on the Nipype tutorial data can be found at: [https://github.com/nipy/nipype/blob/master/examples/smri\\_fsreconall.py](https://github.com/nipy/nipype/blob/master/examples/smri_fsreconall.py).

#### Competing interests

None

#### Author's contributions

HJJ performed the project and wrote the report. DGE created Nipype interfaces for the FreeSurfer commands, scripted the Nipype workflows, ran the workflow output comparisons, and contributed to the report. IO and RK advised on the project and contributed to the report.

#### Acknowledgements

The authors would like to thank the organizers and attendees of the 2015 OHBM Hackathon. This paper is funded by multiple grants: Huntington's Disease Society of America (Human Biology Project Fellowship), 3D Shape Analysis for Computational Anatomy (R01 EB008171), Neurobiological Predictors of HD (R01 NS040068), Cognitive and Functional Brain Changes in Preclinical HD (R01 NS054893), Algorithms for Functional and Anatomical Brain Analysis (P41 RR015241), Enterprise Storage in a Collaborative Neuroimaging Environment (S10 RR023392), Core 2b HD (U54 EB005149), and Nipype (R03 EB008673).

#### Author details

<sup>1</sup>Electrical and Computer Engineering, The University of Iowa, Iowa City, 4016 Seamans Center for the Engineering Arts and Sciences, 52242, Iowa, USA. <sup>2</sup>Department of Biomedical Engineering, The University of Iowa, Iowa City, 1402 Seamans Center for the Engineering Arts and Sciences, 52242, Iowa, USA. <sup>3</sup>Department of Psychiatry, Carver College of Medicine, The University of Iowa, Iowa City, 200 Hawkins Drive, 52242, Iowa, USA. <sup>4</sup>Iowa Institute for Biomedical Imaging, The University of Iowa, Iowa City, 200 Hawkins Drive, 52242, Iowa, USA.

#### References

1. Fischl, B.: FreeSurfer. *Neuroimage* **62**(2), 774–781 (2012)
2. Gorgolewski, K., Burns, C.D., Madison, C., Clark, D., Halchenko, Y.O., Waskom, M.L., Ghosh, S.S.: Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python. *Front Neuroinform* **5**, 13 (2011)
3. Avants, B.B., Tustison, N., Song, G.: Advanced normalization tools (ANTS). *Insight J* (2009)

4. Woolrich, M.W., Jbabdi, S., Patenaude, B., Chappell, M., Makni, S., Behrens, T., Beckmann, C., Jenkinson, M., Smith, S.M.: Bayesian analysis of neuroimaging data in FSL. *NeuroImage*, 45 (2009)
5. Kim, E.Y., Johnson, H.J.: Robust multi-site MR data processing: iterative optimization of bias correction, tissue classification, and registration. *Frontiers in Neuroinformatics* **7**(November), 1–11 (2013). doi:[10.3389/fninf.2013.00029](https://doi.org/10.3389/fninf.2013.00029)
6. Kim, E.Y., Magnotta, V.a., Liu, D., Johnson, H.J.: Stable Atlas-based Mapped Prior (STAMP) machine-learning segmentation for multicenter large-scale MRI data. *Magnetic resonance imaging* **32**(7), 832–844 (2014). doi:[10.1016/j.mri.2014.04.016](https://doi.org/10.1016/j.mri.2014.04.016)
7. Kim, R.E., Paulsen, J.S., Nopoulos, P., Uc, E., Johnson, H.J.: Efficient and Extensible Workflow: Reliable whole brain segmentation for Large-scale, Multi-center Longitudinal Human MRI Analysis using High Performance/Throughput Computing Resources. In: *Clinical Image-Based Procedures. Translational Research in Medical Imaging*. Springer, Chicago, USA (2015)