Report from 2015 Brainhack Americas (MX)

Optimized implementations of voxel-wise degree centrality and local functional connectivity density mapping in AFNI

Project URL: http://github.com/ccraddock/afni

R. Cameron Craddock[1,2*] and Daniel J. Clark[2]

1 Introduction

Degree centrality (DC) [1] and local functional connectivity density (lFCD) [2] are statistics calculated from brain connectivity graphs that measure how important a brain region is to the graph. DC (a.k.a. global functional connectivity density [2]) is calculated as the number of connections a region has with the rest of the brain (binary DC), or the sum of weights for those connections (weighted DC) [1]. lFCD was developed to be a surrogate measure of DC that is faster to calculate by restricting its computation to regions that are spatially adjacent [2]. Although both of these measures are popular for investigating inter-individual variation in brain connectivity, efficient neuroimaging tools for computing them are scarce. The goal of this Brainhack project was to contribute optimized implementations of these algorithms to the widely used, open source, AFNI software package [3].

2 Approach

Tools for calculating DC (`3dDegreeCentrality`) and lFCD (`3dLFCD`) were implemented by modifying the C source code of AFNI's `3dAutoTcorrelate` tool. `3dAutoTcorrelate` calculates the voxel × voxel correlation matrix for a dataset and includes most of the functionality we require, including support for OpenMP [4] multithreading to improve calculation time, the ability to restrict the calculation using a user-supplied or auto-calculated mask, and support for both Pearson's and Spearman correlation.

`3dDegreeCentrality`: Calculating DC is straight forward and is quick when a correlation threshold or is used. In this scenario, each of the $.5{*}N_{vox}{*}(N_{vox}-1)$ unique correlations are calculated, and if they exceed a user specified threshold (default threshold $= 0.0$) the binary and weighted DC value for each of the voxels involved in the calculation are incremented. The procedure is more tricky if sparsity thresholding is used, where the top $P\%$ of connections are included in the calculation. This requires that a large number of the connections be retained and ranked - consuming substantial memory and computation. We optimize this procedure with a histogram and adaptive thresholding. If a correlation exceeds threshold it is added to a 50-bin histogram (array of linked lists). If it is determined that the lowest bin of the histogram is not needed to meet the sparsity goal, the threshold is increased by the binwidth and the bin is discarded. Once all of the correlations have been calculated, the histogram is traversed from high to low, incorporating connections into binary and weighted DC until a bin is encountered that would push the number of retained connections over the desired sparsity. This bin's values are sorted into a 100-bin histogram that is likewise traversed until the sparsity threshold is met or exceeded. Sparsity is exceeded when the differences between correlation values are less than $1.0/(50 * 100)$.

`3dLFCD`: lFCD was calculating using a region growing algorithm in which face-, side-, and corner-touching voxels are iteratively added to the cluster if their correlation with the target voxel exceeds a threshold (default threshold $= 0.0$). Although lFCD was originally defined as the number of voxels locally connected to the target, we also included a weighted version.

[*] Correspondence: ccraddock@nki.rfmh.org

[1] Computational Neuroimaging Lab, Center for Biomedical Imaging and Neuromodulation, Nathan Kline Institute for Psychiatric Research, Orangeburg, 140 Old Orangeburg Rd, 10962, New York, USA

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

Validation: Outputs from the newly developed tools were compared to Python implementations of these measures from the Configurable Pipeline for the Analysis of Connectomes (C-PAC) [5] using a preprocessed version of the C-PAC benchmark dataset (http://fcp-indi.github.io/docs/user/benchmark.html) in terms of similarity, computation time, and memory used.

3 Results

AFNI tools were developed for calculating lFCD and DC from funcitonal nueroimaging data and have been submitted for inclusion into AFNI. LFCD and DC maps from the test dataset (illustrated in Fig. 1) are identical to those calculated using C-PAC but required substantially less time and memory (see Table 1).

4 Conclusions

Optimized versions of lFCD and DC acheived 4× to 30× decreases in computation time compared to C-PAC's Python implementation and decreased the memory footprint to less than 1 gigabyte. These improvements will dramatically increase the size of Connectomes analyses that can be performed using conventional workstations. Making this implementation available through AFNI ensures that it will be available to a wide range of neuroimaging researchers who do not have the wherewithal to implement these algorithms themselves.

Availability of Supporting Data

More information about this project can be found at: http://github.com/ccraddock/afni. Further data and files supporting this project are hosted in the *GigaScience* repository REFXXX.

Competing interests

None

Author's contributions

RCC and DJC wrote the software, DJC performed tests, and DJC and RCC wrote the report.

Acknowledgements

The authors would like to thank the organizers and attendees of Brainhack MX and the developers of AFNI. This project was funded in part by a Educational Research Grant from Amazon Web Services.

Author details

[1]Computational Neuroimaging Lab, Center for Biomedical Imaging and Neuromodulation, Nathan Kline Institute for Psychiatric Research, Orangeburg, 140 Old Orangeburg Rd, 10962, New York, USA. [2]Center for the Developing Brain, Child Mind Institute, New York, 445 Park Ave, 10022, New York, USA.

References

1. Rubinov, M., Sporns, O.: Complex network measures of brain connectivity: uses and interpretations. Neuroimage **52**(3), 1059–1069 (2010). doi:10.1016/j.neuroimage.2009.10.003
2. Tomasi, D., Volkow, N.D.: Functional connectivity density mapping. Proc. Natl. Acad. Sci. U.S.A. **107**(21), 9885–9890 (2010). doi:10.1073/pnas.1001414107
3. Cox, R.W.: AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. Comput. Biomed. Res. **29**(3), 162–173 (1996)
4. Dagum, L., Menon, R.: Openmp: an industry standard api for shared-memory programming. Computational Science & Engineering, IEEE **5**(1), 46–55 (1998)
5. Craddock, C., Sikka, S., Cheung, B., Khanuja, R., Ghosh, S.S., Yan, C., Li, Q., Lurie, D., Vogelstein, J., Burns, R., Colcombe, S., Mennes, M., Kelly, C., Di Martino, A., Castellanos, F.X., Milham, M.: Towards automated analysis of connectomes: The configurable pipeline for the analysis of connectomes (c-pac). Frontiers in Neuroinformatics (42) (2013). doi:10.3389/conf.fninf.2013.09.00042