NICEMAN: NeuroImaging Computational Environments Manager

Submission No:

3613

Submission Type:

Abstract Submission

Authors:

Matthew Travers[1], Robert Buccigrossi[1], Christian Haselgrove[2], Kyle Meyer[3], Yaroslav Halchenko[3]

Institutions:

[1]TCG, Inc., Washington, DC, [2]University of Massachusetts, Boston, MA, [3]Dartmouth College, Hanover, NH

First Author:

Matthew Travers, BS - Lecture Information | Contact Me
TCG, Inc.
Washington, DC

Introduction:

Neuroimaging researchers typically use heterogeneous software environments for the collection, processing, and analysis of their data. Computational resources may be spread across available computing infrastructure including local desktops and laptops, institutional high performance computing clusters, and cloud instances. Computing components (native or containerized) can and do change, making it difficult if not impossible to track exact specifications and even to remember the status or location of any specific resource, data, or analysis. But ongoing changes must be captured to assure that computing environments are consistent across pipeline executions and to make it possible to accurately recreate processing environments at a later date or different location. The NICEMAN project (niceman.repronim.org) is being developed 1) to provide tools that automate and standardize how researchers interface with computing infrastructure to run analyses or work in interactive sessions, 2) to capture provenance information of compute resources, and 3) to give researchers the ability to verify, adjust, or simply create environments according to the previously used or desired specifications.

Methods:

Unified software and data distributions such as (Neuro)Debian (Halchenko and Hanke 2012), Anaconda/Conda-forge, and DataLad (datalad.org), and the use of version control systems such as SVN, Git, and git-annex make it possible to concisely and unambiguously specify or discover detailed information for every component of the computational environment. Services such as snapshots.debian.org (and soon snapshot-neuro.debian.net) allow us to reconstruct original computational environments from exact specifications of Debian environments. There is also a growing number of standards and tool implementations for the collection and storage of basic provenance information (e.g., NIDM used in Nipype (Gorgolewski et al. 2011)). Often such information alone is not sufficient for automated reconstruction of the environments, but they provide a good starting point to identify components that need further specification. Tools such as ReproZip (Rampin et al. 2016) provide a solution to trace the execution of an analysis and compose a provenance profile with a zip bundle of all desired components for the computational environment.

NICEMAN, a free and open-source command line tool currently under development, provides the facility to use existing tools to trace execution and capture detailed information about environment components, and their possible sources. The provenance files can be used as input back into NICEMAN to verify and compare how any given computational environment differs from the target one, or to recreate the computational environment from scratch. Collecting and maintaining the specification with unambiguous details of the needed components instead of a pre-crafted container makes it possible to construct environments across a wide range of base environments. This approach also allows the user to change components (e.g., upgrade analysis software to verify replicability of the results) in a unified fashion.

NICEMAN also provides a registry of computational resources (e.g., remote hosts accessible via ssh, containers, and cloud resources), facilitating unified access and interaction with them. The goal is for NICEMAN to provide a dashboard and entry point to the registered computing resources, existing environments, and running tasks available to the user, while providing the option to have provenance information collected and stored automatically for specific interactive sessions or computational runs.

Results:

NICEMAN is being actively developed and had public releases already. We will demonstrate examples of NICEMAN operation on real analysis use cases (e.g. the "Simple workflow" (Ghosh et al. 2017)).

Conclusions:

NICEMAN can help to establish and manage reproducible computational environments across a wide range of possible deployments.

Informatics:

Workflows [2]
Informatics Other [1]

Keywords:

Workflows
Other - Provenance; Reproducibility; Automation

[1|2]Indicates the priority used for review

My abstract is being submitted as a Software Demonstration.

No

If you answered yes to the question above, would you like a traditional poster in addition to the Software Demonstration?

Yes

Would you accept an oral presentation if your abstract is selected for an oral session?

Yes

I would be willing to discuss my abstract with members of the press should my abstract be marked newsworthy:

No

Please indicate below if your study was a "resting state" or "task-activation” study.

Other

By submitting your proposal, you grant permission for the Organization for Human Brain Mapping (OHBM) to distribute the presentation in any format, including video, audio print and electronic text through OHBM OnDemand, social media channels or other electronic media and on the OHBM website.

I accept

Healthy subjects only or patients (note that patient studies may also involve healthy subjects):

Healthy subjects

Are you Internal Review Board (IRB) certified? Please note: Failure to have IRB, if applicable will lead to automatic rejection of abstract.

Yes

Are you Animal Use and Care Committee (AUCC) certified? Please note: Failure to have AUCC, if applicable will lead to automatic rejection of abstract.

Yes

Please indicate which methods were used in your research:

Functional MRI
Structural MRI

For human MRI, what field strength scanner do you use?

3.0T

Provide references using author date format

Ghosh, Satrajit S., Jean-Baptiste Poline, David B. Keator, Yaroslav O. Halchenko, Adam G. Thomas, Daniel A. Kessler, and David N. Kennedy. 2017. “A Very Simple, Re-Executable Neuroimaging Publication.” F1000Research 6 (February):124.
Gorgolewski, Krzysztof, Christopher D. Burns, Cindee Madison, Dav Clark, Yaroslav O. Halchenko, Michael L. Waskom, and Satrajit S. Ghosh. 2011. “Nipype: A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python.” Frontiers in Neuroinformatics 5 (August):13.
Halchenko, Yaroslav O., and Michael Hanke. 2012. “Open Is Not Enough. Let’s Take the Next Step: An Integrated, Community-Driven Computing Platform for Neuroscience.” Frontiers in Neuroinformatics 6 (June):22.