Data organization and provenance for transparent, efficient and reproducible analytics with neuroimaging and related modalities

Avants

1/2/2020

Introduction

Rigorous organization is key to unlocking the potential of increasingly diverse and high-dimensional scientific data [https://www.sciencedirect.com/science/article/pii/S0958166918301903] and to addressing the perpetual need for reproducible analytics [https://www.pnas.org/content/115/11/2628.short]. In this document, we review concepts and resources used within the neuroimaging community to aid data organization which, in turn, promotes reproducible science. We also propose a relatively lightweight data organization strategy that serves indexing, computational and inferential needs for clinical trial data.

The source code for this document is here.

The path toward enlightened data

Enlighted data arises from transforming raw data into a well-curated, easily visualized and intuitively organized computation-friendly format.

Raw data refers to the data that we receive from an "original" resource. This may be data off of a scanner, a dataset downloaded from a public server or, in some cases, physical samples. One of the most critical steps in data organization in support of science/reproducibility is to record the original source of data and everything that happens to the data thereon. This is known as "provenance" which may be defined as "descriptions of the entities and activities involved in producing and delivering or otherwise influencing a given object." Provenance tracks the path an image may take on its journey from the scanner to inclusion as a variable(s) in a study/statistical analysis.

Well-curated data comes with detailed provenance and is indexed by a tabular file that contains entries with established values (no surprises). A "suprise" in a tabular file (e.g. a csv file) may be:

- a column that has no name (making it impossible to deduce the meaning of the column's values)
- a column of values that contains a mixture of numeric data and character data or includes disorganized strings (such as an entry that reads "I'm not sure what happened to this but it's missing" or "bad value" or "hmmm... ")
- various unpredictable forms of denoting missingness e.g. instead of consistently using the NA signature, using the blank character, the word "missing" or a value such as FALSE or -999.

Well-curated data should have none of these issues and (at least in the world of `R`) should only use NA to denote missingness.

An **easily visualized and intuitively organized computation-friendly format** means that data is organized in a predictable tree structure. A tree has a root, branches and leaves. The leaves are the end of the tree and are, ultimately, data points. The root is the highest level of the data group. The branches map with increasing level of specificity from the root to the leaves. Tree structures are very amenable to computation because they can be easily searched and serve to organize and manage access to data at different points in its hierarchy.

The NRG format is a specification for how neuroimaging data should be organized, on disk, in a tree. Let us begin with an example for how a T1-weighted image may appear within a NRG-compliant directory structure:

```
basedir=/home/avants/ADNIX/data
rootdir=${basedir}ADNI/
t1=${rootdir}/S0001/20071131/T1-MPRAGE/000/ADNI_S0001_20071131_T1-MPRAGE_000.nii.gz
```

Let us unpack the meaning of each of these branches in english:

```
t1=StudyName/SubjectID/Date/Modality/Repeat/StudyName_SubjectID_Date_Modality_Repeat.extension
```

An associated json file would be:

```
${rootdir}/S0001/20071131/T1-MPRAGE/000/ADNI_S0001_20071131_T1-MPRAGE_000.json
```

A second modality and a second repeat would follow the same style:

```
dwi=${rootdir}/S0001/20090629/DWI/001/ADNI_S0001_20090629_DWI_001.nii.gz
```

Static genomics data could be organized in this way as well:

```
snps=${rootdir}/S0001/20090629/SNPSCHIP1/000/ADNI_S0001_20090629_SNPSCHIP1_000.csv
```

While genotype does not change over time, there remains value/information in knowing when the data was collected or in associating the genotype with baseline data alone. Note that the date format has a given resolution that may be finer than shown here. It is also organized such that higher values indicate a later date (this helps sort data effectively).

The **separator**, above, is the underscore (`_`). Ideally, the separator should never be used in the naming of the modalities, the subjects or the study. The separator is used to aid in automated parsing of filenames and in mapping directly from a table to data on disk. Dash (`-`) is another reasonable separator. The character `*` is not a valid choice as it is used for other purposes. At times, a longer separator (`----`) may have value if a suitable short option is not available.

This – in brief – is the NRG format. There are some cases that can stress the implementation of the format and may need some creativity to handle.

- Naming the modality: T1-MPRAGE or T1-MPRAGE-Echo20-GRAPPA-SIEMENS ? This is a judgement call about level of detail; generally, however, sequence details should be available via provenance and/or in the data tables.
- Handling multiple subject IDs: ADNI raises this case as both images have IDs and subjects have IDs. We recommend always sticking with subject IDs, if possible.
- Denoting “baseline”, “screening”, “visit number” and similar schemes: we recommend sticking with alphanumerically sortable date values and leaving these annotations to the tabular format. However, in rare cases, we have effectively concatenated the “visit name” on to the end of a date as a workaround e.g. `20071131VISIT02`.
- Is it “wrong” to store a json file in parallel to a mhd or nifti image? Json-based provenance is the only allowable parallel data within the same directory. However, it could be argued that it should be stored in its own directory:

```
${rootdir}/S0001/20071131/T1-MPRAGE/000/Provenance/ADNI_S0001_20071131_T1-MPRAGE_000_Provenance.json
```

Despite some issues that deserve discussion, NRG remains an effective way to map tabular data to disk and database and avoids the trouble that can occur when many files have the same name (e.g. all T1 images being named `anat.nii.gz`). Furthermore, NRG is likely BIDS compliant, although it is more specific and redundant than default BIDS.

NRG and software provenance

Above, we detailed the general format for analysis ready data. But how do we convert raw data to this format? How should the raw data be organized? And how do we track provenance for each of these items?

Typically, we store data and code in parallel high-level directories. Previously, we defined:

```
basedir=/home/avants/ADNIX/data
rootdir=${basedir}ADNI/
```

where `ADNIX/data/ADNI` contains nifti (and potentially genomics and/or psychometric measurements) data for individual subjects in our study. Raw data should exist elsewhere (maybe `ADNIX/rawdata`) and should remain in its original format and file structure, however disorganized that may be. Any reorganization of that data should be performed by code, on the fly, and the resulting intermediate files discarded when they are no longer needed. In this system, common practice would involve defining:

```
codedir=/home/avants/ADNIX/src/
```

and then storing scripts within `src` and versioning them with tools like `git` and `github`.

Furthermore, we also define:

```
codedir=/home/avants/ADNIX/demographics/
```

which will contain demographics and related data (variables that should live in `csv` file) that will be matched to the imaging data.

To further clarify, we may perform the following steps implemented with a mixture of scripts in the `src` directory:

1. Download raw data to the location from which it will be reorganized into NRG format:

```
src/00_download_raw_data.sh
```

2. Run an organizational script on this raw data.

```
src/01_nrgize_raw_data.R
```

3. Perform a proof-of-concept analysis and/or visualization/tabulation. This might involve reporting the number of time points for each subject and visualizing which modalities are present/absent and the density of the longitudinal sampling over time.

```
src/02_basic_nrg_data_description.py
```

4. Run a full analysis of structural imaging data

```
src/03_basic_freesurfer.py
```

which will result in a new output folder:

```
fsdir=/home/avants/ADNIX/freesurferResults/
```

Ideally, results would also be organized in NRG format:

```
${fsdir}/S0001/20071131/FS-Thickness/000/ADNI_S0001_20071131_FS-Thickness_000.nii.gz
```

The code used in the steps above will not only be version controlled but may also invoke a specific virtual machine or container. At minimum, the `src` directory should detail the software and versions on which it relies by keeping track of executables, libraries, operating system and hardware.

Without going into further detail, we also note that the data directories, themselves, can be version controlled with tools such as `git annex` and `datalad` and the computational environment can be stored via `reprozip` (with some limitations).

Examples

We first show how to convert DICOM data to nifti data in NRG format. We follow this with an example of tabular data that indexes several different data sources consistently.

DICOM 2 Nifti

First, we generate DICOM data based on `ANTsR`. This document (which is compilable and computes the results below) is part of the provenance of this example data.

```
library( ANTsR )
library( dcm2niir )
dcmfns = paste0( "./data/subject", c(1:2), "/"  )
myres = dcm2niir::dcm2nii( basedir=dirname( dcmfns[1]) , verbose=TRUE )
studyID='example'
subjectID='001'
dateID='20071131'
modalityID='SLICE'
repeatID="000"
uniqueIdentifiers = c(studyID, subjectID, dateID, modalityID, repeatID)
nrgDir = paste0( uniqueIdentifiers, collapse='/' )
nrgFN = paste0( uniqueIdentifiers, collapse='-' )
cat("On Disk => ")
cat( paste0( nrgDir, "/", nrgFN ) )
tempimg = antsImageRead(  myres$nii_after  )
# OPTIONALLY: compute some statistics / basic data checking here
dir.create( nrgDir, recursive = T )
antsImageWrite( tempimg, paste0( nrgDir, "/", nrgFN ) )
```

The `myRecon` variable summarizes the execution of `dcm2niix` from within `R`. The usual workflow would be to use these results to do the reconstruction then *move* and *rename* the `nifti` and `json` data such that it fits NRG format.

One might also use the output of the above commands to identify the acquisition date. More generally, one may derive the various variables required for the NRG format via the DICOM.

Future work may develop this system into a well-documented `R` package. Utilities, examples and high-level helper functions will assist in the modality/manufacturer specific issues in data conversion, permit higher frequencies of re-use and speed data organization in future efforts.

Computational Environment: Important provenance information:

```
pander::pander( sessionInfo() )
```

R version 4.0.0 (2020-04-24)

Platform: x86_64-apple-darwin17.0 (64-bit)

locale: en_US.UTF-8||en_US.UTF-8||en_US.UTF-8||C||en_US.UTF-8||en_US.UTF-8

attached base packages: *stats*, *graphics*, *grDevices*, *utils*, *datasets*, *methods* and *base*

other attached packages: pander(v.0.6.3)

loaded via a namespace (and not attached): *compiler(v.4.0.0)*, *magrittr(v.1.5)*, *tools(v.4.0.0)*, *htmltools(v.0.4.0)*, *yaml(v.2.2.1)*, *Rcpp(v.1.0.4.6)*, *stringi(v.1.4.6)*, *rmarkdown(v.2.2.3)*, *knitr(v.1.28)*, *stringr(v.1.4.0)*, *xfun(v.0.14)*, *digest(v.0.6.25)*, *rlang(v.0.4.6)* and *evaluate(v.0.14)*

Example of machine-learning ready indexed and merged datasets

Result of merging several different data sources. NOTE: this is "long" format i.e. each time point for each subject occupies one row. Generally, this is much easier to handle the "wide" format which tries to keep all subjects' data on a single row. The example combines and indexes both NIFD and 4RTNI in a consistent manner. PPMI, ADNI and other datasets would be added in the same manner.

```
myData = read.csv( data_fn )
set.caption("Example combination of 4RTNI and NIFD data.")
pander( myData )
```

Table 1: Example combination of 4RTNI and NIFD data. (continued below)

study	INDDID	phenotype	AgeatMRI	Sex	Education	AgeatBaseline
NIFD	0001	Control	64.49	Male	15	62.39
NIFD	0002	bvFTD	56.59	Female	13	54.62
4RTNI	1_S_5093	PSP	82	Female	12	82
4RTNI	1_S_5093	PSP	82.52	Female	12	82
4RTNI	1_S_5093	PSP	82.52	Female	12	82
4RTNI	1_S_5094	CBS	60	Female	16	60

pathology	deltaTime	IsBaseline	MRIDate	imageId
Control	2.096	0	2010-05-21	NIFD-0001-20100521-t1mprageT1-1
Unknown	1.978	0	2011-07-19	NIFD-0002-20110719-t1mprageT1-0
Unknown	0	1	2014-10-21	4RTNI-1_S_5093-20141021-T1w-1
Unknown	0.5205	0	2015-04-29	4RTNI-1_S_5093-20150429-T1w-1
Unknown	0.5205	0	2015-04-29	4RTNI-1_S_5093-20150429-T1w-2
Unknown	0	1	2014-10-13	4RTNI-1_S_5094-20141013-T1w-1

It is worth looking closely at the printed table above. This illustrates repeat data, longitudinal data, cross-sectional data and the style of tabulating such data such that we can perform inference across several different studies. This is a relatively *minimal* example. Other columns that might be useful would be a *baseline* indicator column, a *repeat* column and a *MRIPROTOCOL* column. A provenance filename or, in ADNI, an *ImageID* column could also prove valuable.

Common issues of consistency:

- naming of disease phenotypes: shorthand versus full label vs pseudonymous categories are all problematic;
- gender versus sex and labels for these ("M" vs "Male" vs "MALE" are all different but have the same meaning);
- pathology (or genetic) group definitions can be quite specific / unshared so different aggregate summary categories may be necessary in order to gain statistical power;
- meaning of time points versus repeat data;
- poor choice of separator (as might happen in the example above) which could lead to some manageable but irritating problems with parsing.

Relevant resources

- R / python for basic scripting and visualization needs

- dcm2niix, dcm2niir for converting dicom to nifti
- containers e.g. docker (for computational environment)
- git for version controlling code
- git annex for version controlling data
- datalad: "With DataLad you can not only gain access to the data resources and maintain your computational scripts under version control system, you can maintain the full record of the computation you performed in your study."
- heudiconv: converting dicom to nifti in BIDS format
- reprozip: storing a computational environment
- ReproNim training http://www.repronim.org/ohbm2018-training/
- Relevant papers
  - "The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments" https://www.ncbi.nlm.nih.gov/pubmed/27326542
  - "Best practices in data analysis and sharing in neuroimaging using MRI" https://www.ncbi.nlm.nih.gov/pubmed/28230846
  - "Everything Matters: The ReproNim Perspective on Reproducible Neuroimaging" https://www.frontiersin.org/articles/10.3389/fninf.2019.00001/full
  - Commentary on BIDS vs NIDM: https://www.incf.org/blog/bids-and-neuroimaging-data-model-nidm
  - "Computational and Informatic Advances for Reproducible Data Analysis in Neuroimaging" (2019) https://www.annualreviews.org/doi/abs/10.1146/annurev-biodatasci-072018-021237