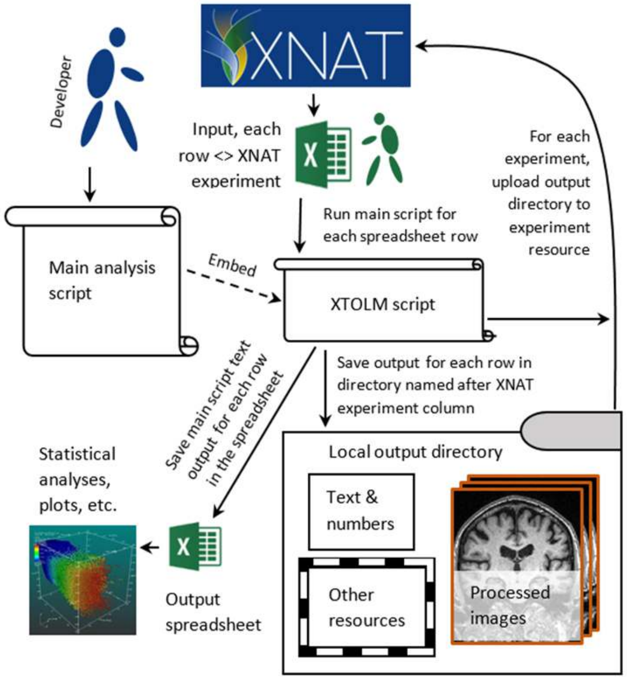
XTolm: an XNAT-aware Linux shell batch analysis environment

Mikhail Milchenko, Dan Marcus.
https://github.com/NrgXnat/xnat_bash_utils

Washington University in St. Louis
School of Medicine
MIR Mallinckrodt Institute of Radiology

Many imaging data analysis scripts are developed in Linux shell. These scripts are often developed ad-hoc and are hard to reuse. Over the years, we isolated a set of best practices and standard routine tasks relevant to developing such analyses against XNAT-hosted data. We implemented those in an experimental bash wrapper environment called XTolm[1]. XTolm offers developers several syntactical structures for iteration over spreadsheets, nested metadata management, and generation of results spreadsheets. XTolm also wraps XNAT REST API serialization, allows working on spreadsheet row subranges and supports grid execution.

XTolm workflow


Developer
XNAT
Input, each row <> XNAT experiment
Run main script for each spreadsheet row
For each experiment, upload output directory to experiment resource
Main analysis script
Embed
XTOLM script
Save output for each row in directory named after XNAT experiment column
Save main script text output for each row in the spreadsheet
Local output directory
Text & numbers
Other resources
Processed images
Statistical analyses, plots, etc.
Output spreadsheet

XTolm experiment context

The basic XTolm data organization unit is an auto-populated "context". Context is a collection of variables specific to a single row in input spreadsheet. XTolm supports syntax to easily iterate through arbitrary subsets of spreadsheet indices, automatically initializing local script variables to current spreadsheet row values.

Adapting analysis scripts to XTolm

```
#!/bin/bash

# XTOLM reads each spreadsheet column into an array variable
# named after column header. Header names must comply
# with shell variable naming rules.
load_data <spreadsheet file>
...
# Main cycle where each spreasheet row is processed. XTolm
# (1) creates and cd's to experiment processing dir, and (2)
# initializes column variables to values from the current row.
# $range indicates which rows to process, e.g. 2-5 or 2,3,4-10
xt_for <subject_column> <experiment_column> $range

    # load DICOM's from XNAT experiment to current dir.
    # scan id's may be fixed or stored in spreadsheet column
    load_scans <id1>[id2,...]

    #execute analysis script and assign output variables, e.g.
    dcm2niix –f brain.nii –o . study2
    bet brain brain_mask -m
    brain_vol=`fslstats -M brain_mask.nii`

    # save computed variables to local experiment context
    save_vars <variable1[,...]> #e.g. save_vars $brain_vol

    # save any subdir of the current processing dir as
    # XNAT experiment resource.
    save_dir <dir_name>

# Finish the row processing cycle and cd out of the experi-
ment dir
xt_done

#write the spreadsheet with computed variables.
summary <label> [column]
```

Multiple variable session contexts

XTolm supports one nested level of context per experiment, i.e. each experiment may have multiple named contexts. Useful for e.g. measurement of multiple lesions per brain.

Built-in variables

Built-in variable names start with (xt_) and form three groups:

(a) XNAT credentials: XNAT server, user, password, jsession
(b) Context variables: XNAT project, subject, experiment
(c) Utility variables: analysis ID, to run multiple analyses on the same data without overwriting variables.

Grid execution (experimental)

XTolm command line supports execution on a grid engine, primarily SLURM (SGE support retired).

Applications

XTolm is most useful if you build with it from the start. It saves development/rerunning effort over long term. Specific applications include

- Cross-sectional imaging study analyses
- Creating training sets for AI applications
- Multi-experiment GUI application tasks that can load data from command line, such as FSLView
  - output quality review, quality control
  - labeling

Conclusion

Xtolm is a local Linux console environment that simplifies:

- Batch execution of image analyses
- Serialization of experiment scans and resources with XNAT
- Re-running analyses (optionally, on XNAT-archived data)
- Creating spreadsheets to build summary graphs and downstream statistical analyses.

[1] In "XTolm", "X" is inherited from XNAT; "Tolm" comes from "translator" in German (dolmatscher) and old Russian (толмач)