Reproducible Research in SEP

Yang Zhang

June. 2011
Outline

- Background
- Reproducibility workflow
- Current practice
- Interactive visualization example
- Discussion
Reproducible research origin

- J. Claerbout, 1992, “Electronic documents give reproducible research a new meaning”
Reproducible research

- scientific computational research is hard to repeat
  - paper/documents cannot hold all the details
  - tricks hidden by the author
  - different computing environment
  - data availability
Non-reproducibility hurts us

- difficult for other researchers to verify
- harder to promote our research
- slowdown the technology transfer
- difficult for the followers (even ourselves) to pick up and continue at later time
Electronic reproducibility documents

- Overcome (mitigate) all the previous issues
- more flexible graphic demonstration approaches
Outline

- Background
- Reproducibility workflow
- Current practice
- Interactive visualization example
- Discussion
Software

- SEPlib
- GNU Make
- vplot
- LaTex & Scons
SEPlib

- (one of) the first seismic processing package in academia
- header-binary data storage style
- 3-D geometry handling and data regularization
- multi-dimension regular-sampled data visualization tool (Sep_cube)
GNU Make, Grammar

- **makefile** - A bunch of rules to build certain targets

  ```
  # Comments are started with the hash(#) symbol.
  target [target ...]: [component ...]
  [<TAB>command 1]
  
  
  .
  
  .
  
  [<TAB>command n]
  ```

  mymac:~zyang03$> make target
GNU Make

- source code compilation and build
- data process flow
- figure generation cmd
GNU Make

- source code compilation and build
- data process flow
- figure generation cmd

```
CC      = gcc
CFLAGS  = -g

all: helloworld

helloworld: helloworld.o
    # Commands start with TAB not spaces
    $(CC) $(LDFLAGS) -o $@ $^  

helloworld.o: helloworld.c
    $(CC) $(CFLAGS) -c -o $@ $<
```
GNU Make

- source code compilation and build
- data process flow
- figure generation cmd

$K/spike_rand_8_sub16.H: ${BINDIR}/Gen_spike_KMig.x
   ${BINDIR}/Gen_spike_KMig.x nt=128 nx=128 dt=0.01 dx=10.0 \n   n_spike=8 max_spike_sz_x=5 max_spike_sz_t=5 rand_seed=8479 > @
GNU Make

- source code compilation and build
- data process flow
- figure generation cmd

```
$\text{R/datasub.v: }$\text{K/dat-intl-kirch-rand-8.H}
< $\text{K/dat-intl-kirch-rand-8.H Window3d n3=1 f3=0 | Grey wantscalebar=y \{ft\_sz\}}$
pclip=100 title="Data collected" out=$\text{R/datasub.v}$ $(\text{dn})$
```
Reproducible research flow

SEPlib library

- develop computer code for your algorithm

build executables

- build figures on the processing result
- incorporate figures into LaTeX document

verify the algorithm by applying to data

Reproducibility test

GNU Make

Stanford Exploration Project
SEP’s reproducibility criterion on results (figures)

- easily reproducible (ER)
- conditionally reproducible (CR)
  - Computing time exceeds 20mins
  - requires unconventional (less universal) computing hardware/platform (Cluster)
  - depends on real-data availability
- non-reproducible (NR)
  - hand-drawing
  - figures from outer sources

Stanford Exploration Project
Reproducibility makefile rules

- make exec
- make ER/CR
- make clean
- make burn
- make view
Project Example

- Courtesy of Adam Halpert

```c
#include $(SEPINC)/SEP.top
LIBDIR = -L${SEP}/lib
UCPPLIBS = -lsep3d -lsep -lsepaux
UF90LIBS = -lsepeef90 -lsepf2df90 -lvplotf90 -lvplot -lsepauxf90
UF90INCLUDES = ${LIBDIR}
SRCDIR = Src/
RESULTSER =
RESDIR = Fig/
R = $(RESDIR)
BINDIR=./
dn = >/dev/null
grey = Grey $(dn)
```
Project Example

- Courtesy of Adam Halpert

```
segment_3d.x: Src/segment_3d.cpp Src/segment-image_3d.h Src/segment-graph.h Src/disjoint-set.h
            icc -g -O3 -I. -ISrc -I${SEPINC} -o $@ Src/segment_3d.cpp -lm -L${SEP}/lib -lsep3d -lsep -lse
            aux

exec:
  make segment_3d.x
  make Merge_segments.x
  make Int_input_3d.x
```
Project Example

- Courtesy of Adam Halpert

```makefile
ER:
make oct-2d
make oct-3d
make 2d-merge
make 2d-toppase
make 2d-env-orig
make 2d-env-new
make 2d-merge-new
make 3d-origseg
make oct-3d-picks
make 3d-env-orig
make 3d-env-new
make 3d-newseg
make o3d-far
make o3d-env-orig-far
make o3d-env-new-far
make 3d-origseg-far
make 3d-newseg-far
```
Project Example

- Courtesy of Adam Halpert

% env.H:
  < * . H Envelope | Scale > $ @

o2d-seg.H: o2d-env.H segment_3d.x
  < $ < segment_3d.x max_dist=4 min_size=2500 > $ @
  < $ < Merge_segments.x picks=2d-picks.H sval=50 > m1.H
  < m1.H Merge_segments.x picks=2d-wb-picks.H sval=.05 > $ @
  Rm m1.H
2d-merge: o2d-merge.H
  Sep_cube o2d.H o2d-merge.H run_history=2d-merge.txt
Project Example
Interactive pdf documents

- Sep_cube (more info on Bob Clapp web page)
- Python & socket
- demo
Some issues

- GNU Make
  - automatic removal of intermediate file (solved now)
  - does not resolve the source dependency for C/C++
  - rebuild criterion based
- learning curve
  - tab/space
  - variable assignment
  - less programmability
  - result %H (result.$(p))
  - result.%.H%.2.H (result.$(p))
  - bash
Some issues

- GNU Make
  - automatic removal of intermediate file (solved now)
  - does not resolve the source dependency for C/C++
  - rebuild criterion based on modified date
  - learning curve
    - tab/space
    - variable assignment
    - less programmability
      - $result.\%.H (result.\${par1}.H) is OK, but
      - $result.\%.1.H.\%.2.H (result.\${par1}.\${par2}.H is not supported.
    - bash

Stanford Exploration Project
Acknowledgement

- Adam Halpert
- Bob Clapp
- Sjoerd de Ridder
- Yaxun Tang
Reference


- Thank you