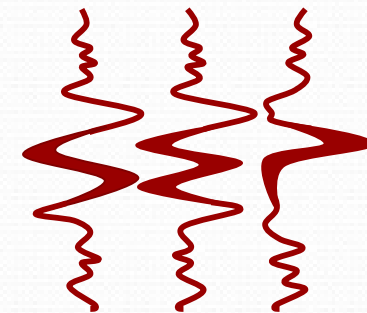


Reproducible Research in SEP

Yang Zhang

June. 2011



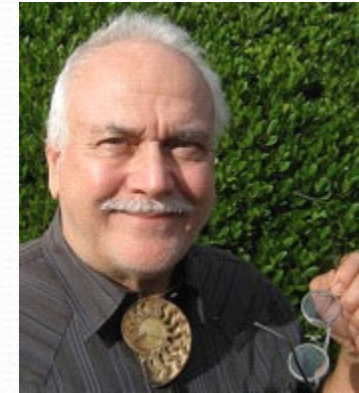
**Stanford
Exploration
Project**

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_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

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

Reproducible research flow

SEPLib library

develop computer code for your algorithm

build executables

verify the algorithm by applying to data

Reproducibility test

GNU Make

incorporate figures into Latex document

build figures on the processing result

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

Reproducibility makefile rules

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

Project Example

- Courtesy of Adam Halpert

```
1 include $(SEPINC)/SEP.top
2
3 LIBDIR = -L${SEP}/lib
4 UCPPLIBS = -lsep3d -lsep -lsepaux
5
6 UF90LIBS = -lsepgeef90 -lsep2df90 -lvplotf90 -lvplot -lsepauxf90
7 UF90INCLUDES = ${LIBDIR}
8
9 SRCDIR = Src/
10 RESULTSER =
11 RESDIR = Fig/
12 R = $(RESDIR)
13 BINDIR=./
14
15 dn = >/dev/null
16 grey = Grey $(dn)
17
```

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  
  gcc -g -O3 -I. -ISrc -I${SEPINC} -o $@ Src/segment_3d.cpp -lm -L${SEP}/lib -lsep3d -lsep -lsep  
aux
```

```
exec:
```

```
  make segment_3d.x  
  make Merge_segments.x  
  make Int_input_3d.x
```

Project Example

- Courtesy of Adam Halpert

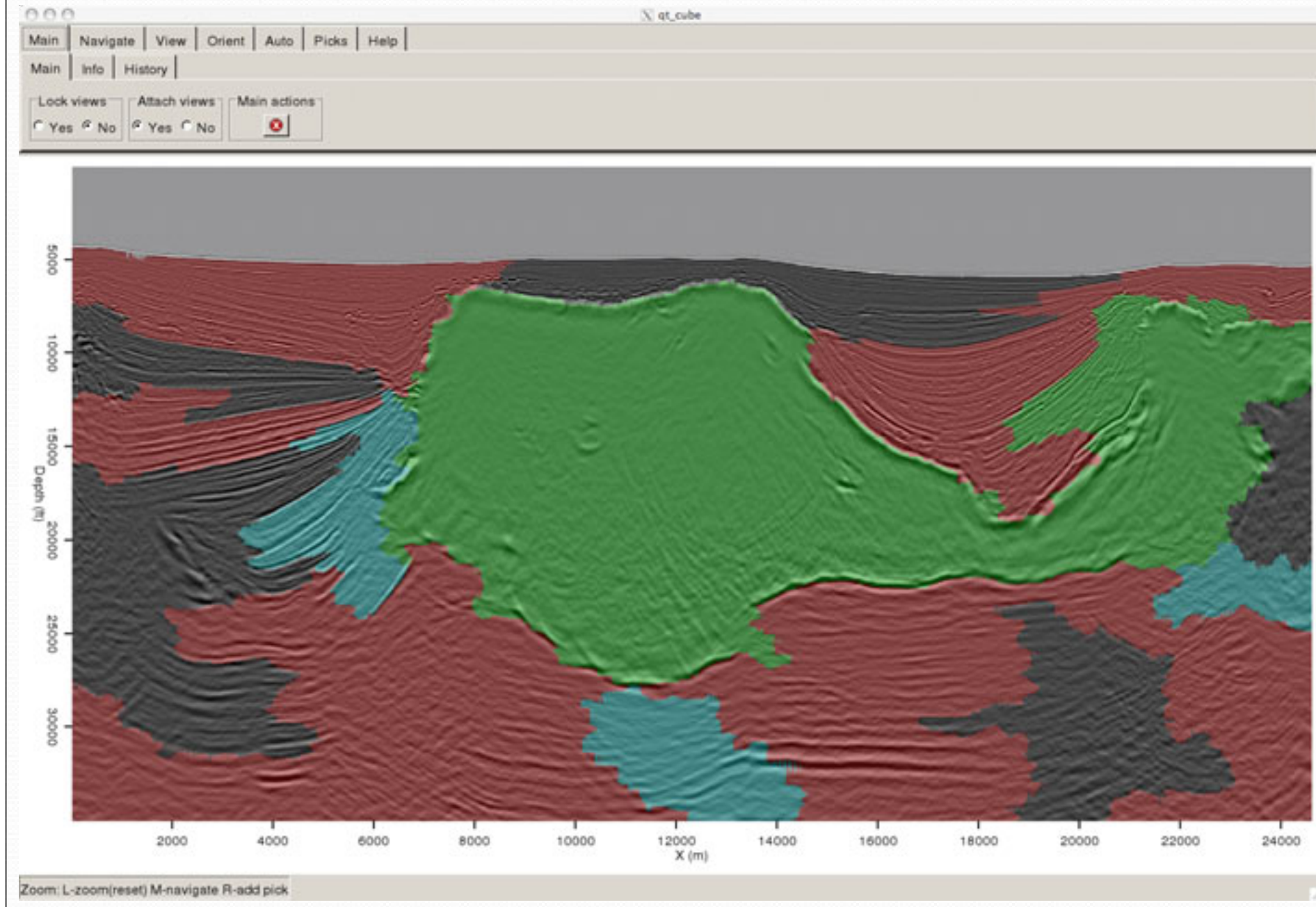
```
ER:  
make oct-2d  
make oct-3d  
make 2d-merge  
make 2d-lobbase  
make 2d-env-orig  
make 2d-env-new  
make 2d-merge-new  
make 3d-origseg  
make oct-3d-picks  
make 3d-env-orig  
make 3d-env-new  
make 3d-newseg  
make o3d-far  
make o3d-env-orig-far  
make o3d-env-new-far  
make 3d-origseg-far  
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 > $@  
o2d-merge.H: o2d-seg.H 2d-picks.H 2d-wb-picks.H Merge_segments.x  
  < $< 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

Demo

The screenshot shows a Mac desktop environment. In the foreground, a Beamer presentation window is open, displaying a slide titled "Some issues". The slide lists several points about GNU Make:

- GNU Make
 - automatic removal of intermediate file (solved now)
 - does not resolve the source dependency for C/C++
 - rebuild criterion based on modified dates, some time it is not desired.
 - learning curve
 - tab/space
 - variable assignment
 - less programmability
 - `result.%H` (result.\$(parameter))
 - `result.%1.H.%2.H` (result.\$(parameter1).\$(parameter2).H) not supported.
 - bash

Below the main list, there is a note: "GNU make is meant for software build, does not suit our reproducible research idea 100 percent." and another note: "The `result.%H` example is very useful in batch build scenario, or parameter tuning. However the support is limited. If you need more flexibility, you need to learn bash script, which is more difficult (plus less readable) than python. However people are lazy, and when they get used to it they are reluctant to switch to something new, unless you can convince them the new stuff is significant better."

Overlaid on the presentation is a terminal window showing the following commands and output:

```
~/homs/sep/adam/Setup/cube_sep ./kir/segult.img.ssfpi5.T gtxmaps-y
[glad]~/interactive: SEP_pdf book.pdf
start_pdf
acropdf book.pdf &
start_server
Fig ["store1": "/homes/sep/adam/Setup/cube_sep PATH/segult.img.ssfpi5.T gtxmaps-y", "store2": "tube R32DR/syn.v", "store": "/homes/sep/adam/Setup/cube_sep PATH/./offlag.R gtxmaps-y"]
~/homs/sep/adam/Setup/cube_sep ./kir/segult.img.ssfpi5.T gtxmaps-y
Fig D
looking for ./dlx/fig/input-dlx-real.v
Fig D
looking for ./dlx/fig/dlx-real.v
[glad]~/interactive: SEP_pdf book.pdf
start_pdf
acropdf book.pdf &
start_server
Fig ["store1": "/homes/sep/adam/Setup/cube_sep PATH/segult.img.ssfpi5.T gtxmaps-y", "store2": "tube R32DR/syn.v", "store": "/homes/sep/adam/Setup/cube_sep PATH/./offlag.R gtxmaps-y"]
~/homs/sep/adam/Setup/cube_sep ./kir/segult.img.ssfpi5.T gtxmaps-y
Fig D
looking for ./dlx/fig/dlx-real.v
[glad]~/interactive:
```


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

Acknowledgement

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

Reference

- <http://sepwww.stanford.edu/doku.php?id=sep:research:reproducible>

- Thank you