• What is GMTSAR?

• Precise orbits and accurate timing simplifies software and enables seamless mosaicing.

• Geometric validation of ERS, Envisat, and ALOS.
An InSAR processing system based on GMT

David Sandwell - Scripps Institution of Oceanography
Rob Mellon - San Diego State University
Xiaopeng Tong - Scripps Institution of Oceanography
Meng Wei - Scripps Institution of Oceanography
Paul Wessel - University of Hawaii
Looking for volunteers to develop scripts

GMTSAR is an open source (GNU General Public License) InSAR processing system designed for users familiar with Generic Mapping Tools (GMT). The code is written in C and will compile on any computer where GMT and NETCDF are installed. The system has three main components:

1. a preprocessor for each satellite data type (e.g., ERS, Envisat, and ALOS) to convert the native format and orbital information into a generic format
2. an InSAR processor to focus and align stacks of images, map topography into phase, and form the complex interferogram
3. a postprocessor, mostly based on GMT, to filter the interferogram and construct interferometric products of phase, coherence, phase gradient, and line-of-sight displacement in both radar and geodetic coordinates.

GMT is used to display all the products as postscript files and KML images for Google Earth. A set of C-shell scripts has been developed for standard 2-pass processing as well as image alignment for stacking and time series. ScanSAR processing is also possible but requires a knowledgeable user. Users are welcome to contribute to this effort. In particular contributions using other scripting languages such as Perl and Python are desired.

ACKNOWLEDGEMENTS This research was supported by ConocoPhillips, Scripps Institution of Oceanography, and San Diego State University.
Two-stop shop for InSAR:

- **Software** - http://gmt.soest.hawaii.edu/projects/gmt5sar/wiki
- **Documentation**
- **Orbits** (ERS and Envisat)
- **Example data sets**
- **GNU public license, no passwords, no registration, no restrictions**
SOFTWARE INSTALLATION INSTRUCTIONS
http://gmt.soest.hawaii.edu/projects/gmt5sar/wiki

Installation of GMT5 and GMT5SAR

1) Install GMT5 first with all optional libraries (GDAL and PCRE, plus FFTW3 and LAPACK for Linux).
   - Linux: See Building Instructions
   - OS X (Macports):
     ```
     sudo port install gdal +curl +geos +hdf5 +netcdf +tiff
     sudo port install gmt5
     ```
   - OS X (Fink):
     ```
     sudo fink install libtiff5
     sudo fink install gmt5
     ```

2) Install extra libraries. Note that depending on your OS version the actual version numbers in some of the packages below may differ:
   - Ubuntu 14.04 LTS:
     ```
     sudo apt-get install csh subversion autoconf libtiff5-dev libhdf5-dev gfortran
     sudo apt-get install gmt
     ```
   - CentOS 7:
     ```
     sudo yum install svn autoconf gcc-c++ libtiff-devel libhdf5-devel
     sudo yum install gmt
     ```
   - Fedora 23:
     ```
     sudo yum install csh subversion autoconf gcc-c++libtiff-devel hdf5-devel
     (need fftw-developer for faster execution.)
     sudo yum install gmt
     ```

3) Download and install orbit files and place in suitable directory (e.g., /usr/local/orbits):
GMTSAR Processing Modes

2-pass processing
   no ground control
   seamless abutment of frames along track

stacking for time series
   one master and many slaves
   primary, secondary, tertiary, image alignment

ScanSAR Interferometry
2-pass processing with no ground control

p2p_ALOS.csh IMG-HH-ALPSRP207600640-H1.0__A IMG-HH-ALPSRP227730640-H1.0__A config.alos.txt

# processing stage#
# 1 - start from pre_proc.csh
# 2 - start from align.csh
# 3 - start from topophase.csh
# 4 - start from int.csh
# 5 - start from snaphu.csh
# 6 - start from geocode.csh
proc_stage = 1

# parameters for pre_proc.csh#
# # num of patches (usually 1~3)
num_patches = 2

# earth radius
earth_radius = 6370634.080057

# near_range
near_range = 860807

# Doppler centroid
fd1 = 54.6
2-Pass Processing Example

Feb 27, 2010, M8.8, Maule Chile Earthquake ALOS SAR data

Each frame is processed independently on a different CPU and recombined later in radar or lon-lat coordinates. This requires a consistent geometry and no geometric adjustments.

[Tong et al., 2010]
Each frame processed independently on a different CPU and recombined later in radar or lon-lat coordinates.

This requires a consistent geometry and no geometric adjustments.

Snaphu phase unwrapping has a consistent zero level of $+/- \ 2\pi N$ so unwrapped phase can be recombined.

Feb 27, 2010, M8.8, Maule Chile Earthquake
ALOS SAR data
Each frame processed independently on a different CPU and recombined later in radar or lon-lat coordinates.

This requires a consistent geometry and no geometric adjustments.

Snaphu phase unwrapping has a consistent zero level of +/- $2\pi N$ so unwrapped phase can be recombined.

Feb 27, 2010, M8.8, Maule Chile Earthquake
ALOS SAR data
Each frame processed independently on a different CPU and recombined later in radar or lon-lat coordinates.

This requires a consistent geometry and no geometric adjustments.

Snaphu phase unwrapping has a consistent zero level of $\pm 2\pi \text{N}$ so unwrapped phase can be recombined.

Feb 27, 2010, M8.8, Maule Chile Earthquake
ALOS SAR data
Each frame processed independently on a different CPU and recombined later in radar or lon-lat coordinates.

This requires a consistent geometry and no geometric adjustments.

Snapahu phase unwrapping has a consistent zero level of +/- $2\pi N$ so unwrapped phase can be recombined.

Feb 27, 2010, M8.8, Maule Chile Earthquake
ALOS SAR data
Each frame processed independently on a different CPU and recombined later in radar or lon-lat coordinates.

This requires a consistent geometry and no geometric adjustments.

Snaphu phase unwrapping has a consistent zero level of $\pm 2\pi N$ so unwrapped phase can be recombined.

Feb 27, 2010, M8.8, Maule Chile Earthquake
ALOS SAR data
GMTSAR Config Files

• /GMT5SAR/gmtsar/csh/
  – config.alos.slc.txt SLC_factor
  – config.alos2.slc.txt SLC_factor
  – config.envi.txt
  – config.rs2.txt
  – config.alos.txt
  – config.csk.txt
  – config.ers.txt
  – config.tsx.txt
  – config.s1a.txt

• filter → filter_wavelength
GMTSAR Config File Example

- # processing stage #
- #--------------------------------------------------#
- # 1 - start from preprocess
- # 2 - start from focus and align SLC images
- # 3 - start from make topo_ra
- # 4 - start from make and filter interferograms
- # 5 - start from unwrap phase
- # 6 - start from geocode
- proc_stage = 1