

InSAR Scientific Computing Environment

Eric Gurrola, Gian Franco Sacco, Piyush Agram, Marco Lavallo,
Paul A. Rosen, Mark Powell, Darren Dao

Jet Propulsion Laboratory
California Institute of Technology

Howard A. Zebker
Stanford University

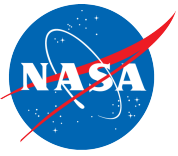
ESTF 2014
Leesburg, Virginia
October 30, 2014

Copyright 2014 California Institute of Technology. U.S. Government
sponsorship acknowledged.



JPL

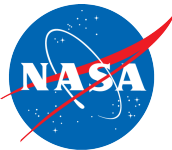




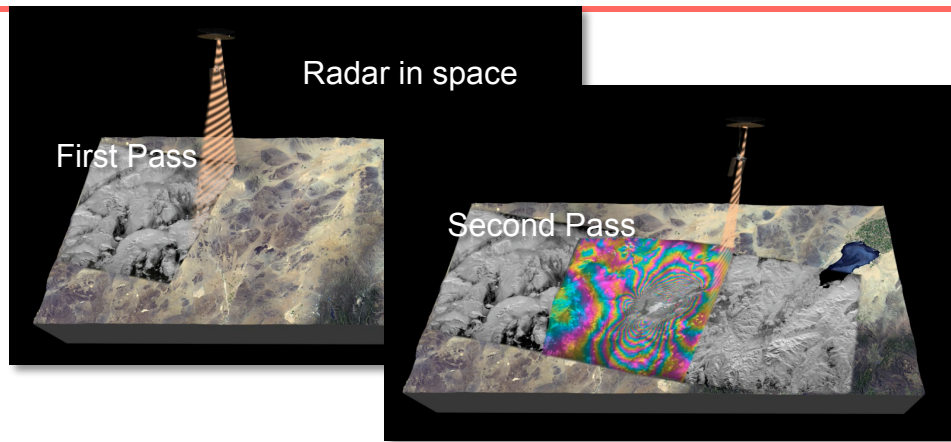
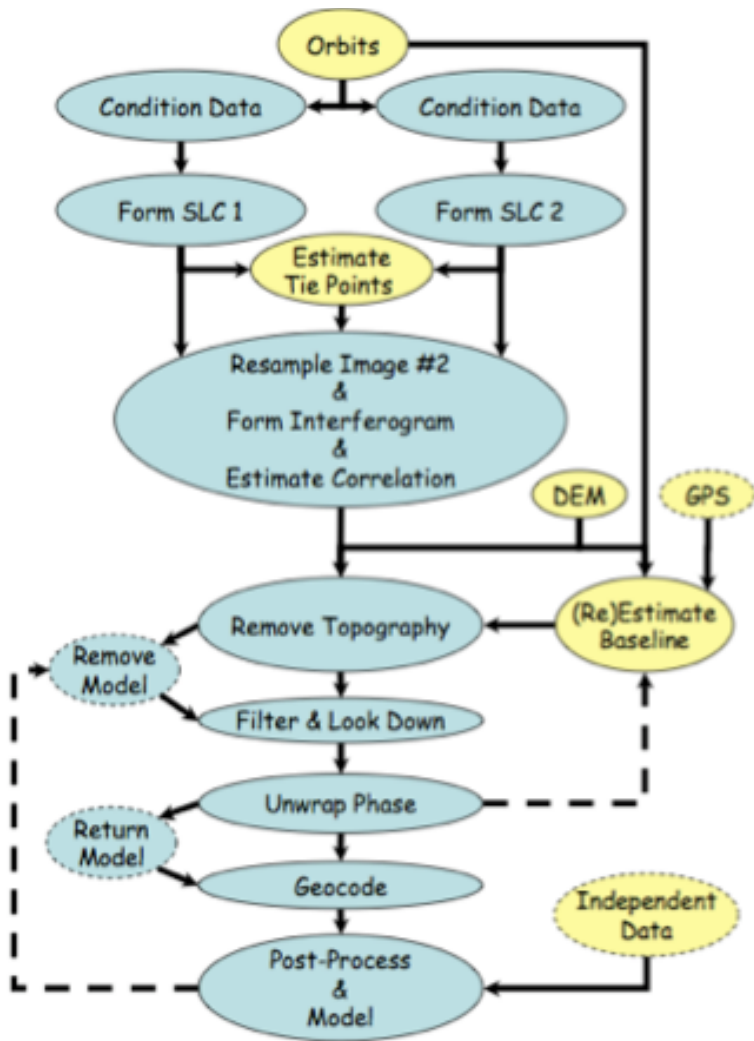
Outline

- Introduction
- Technology Description
- Functionality
- Part of the ARIA-MH Production System
- Core Technology in Earthkit and CESCRES cloud-based computing
- Baseline for proposed NISAR Algorithm Core
- Recently upgraded to support UAVSAR Interferometry and Polarimetry for Time-Series stacks
- Vision for ISCE architecture
 - What international missions it supports and how
 - How does it support the user community

Pre-decisional - for Planning and Discussion Purposes Only



Interferometric Synthetic Aperture Radar (InSAR) From Recipes to Reconfigurable Flow



```

/Users/parosen/Projects/CSK/hawaii> python3
>>> import isce
>>> from applications.insarApp import Insar
>>> a = Insar()
>>> a.configure()
  
```

```

2014-10-18 17:47:51,008 - isce.insar - INFO -
ISCE VERSION = 2.0.0_201409, RELEASE_SVN_REVISION =
1612, RELEASE_DATE = 20140918,
CURRENT_SVN_REVISION = 1647
  
```

```

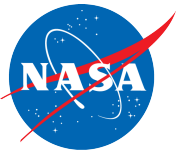
>>> a.rangeLooks = 4
>>> a.run()
  
```

Insar Application: Implements InSAR processing flow for a pair of scenes from sensor raw data to geocoded, flattened interferograms.

Fixed Prescriptive Flow

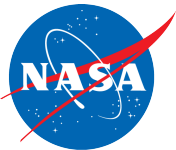


Flexible "Sandbox" ISCE Flow



ISCE Algorithms

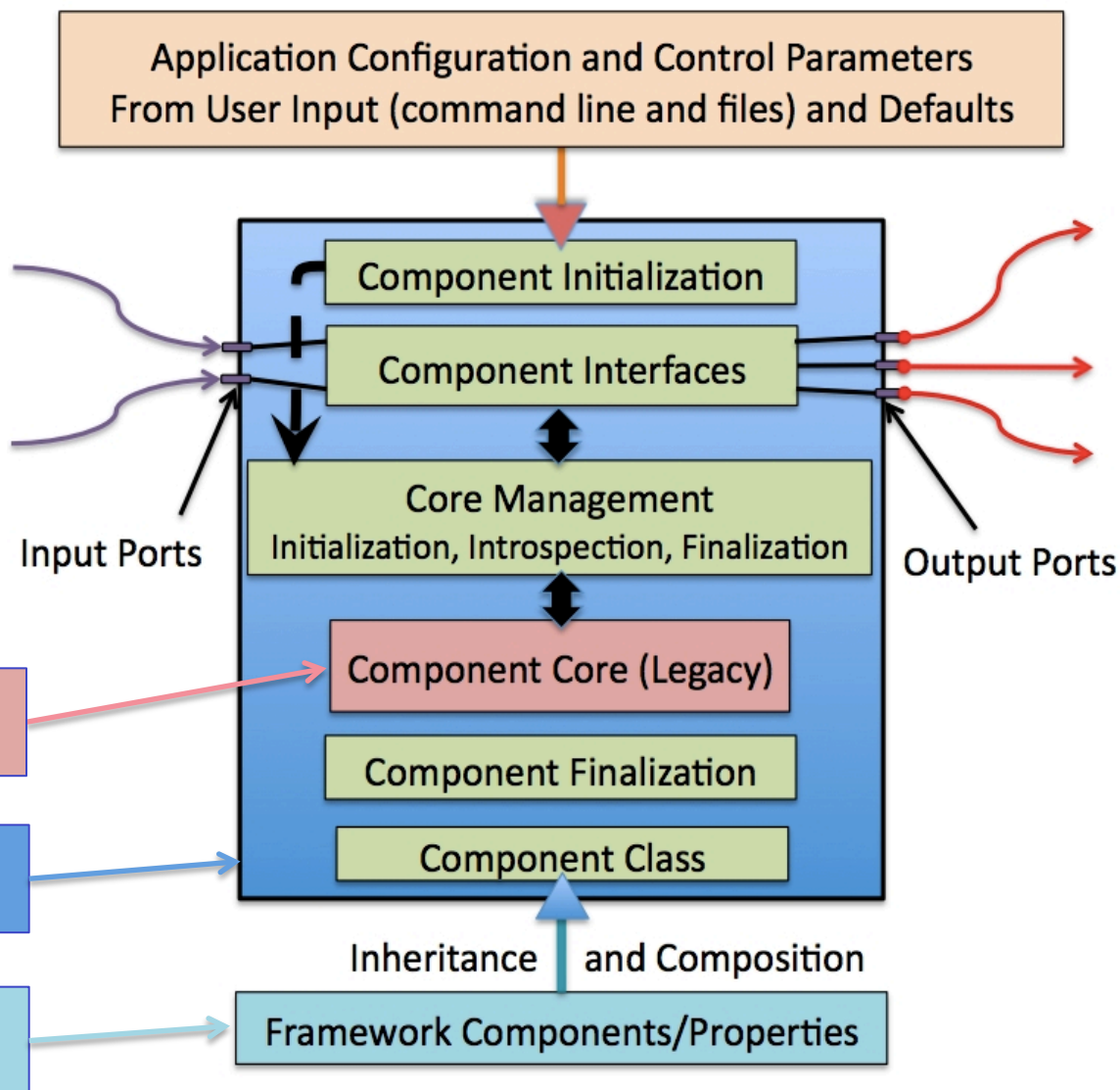
- Created accurate and fast algorithms based on GEOSAR and SRTM heritage
- Advantages of improved accuracy and speed:
 - Enable InSAR and time series methods with precise SLC alignment requirements
 - Straightforward merging SAR data with other types
 - Feasible processing of dozens of scenes by a desktop user
- Approach
 - Define a rigorous geometric framework tied to local spherical coordinate system
 - Adjust collection of images to a common reference trajectory
 - Condition data for ingestion in post-processing time-series applications such as GIAN-T



The Architectural Core of ISCE

By adopting a general object oriented framework for processing, ISCE

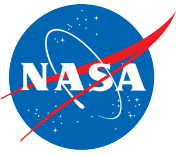
- Enables extensibility and flexibility by abstracting functions that are typically embedded in scientific codes
- Allows straightforward inclusion of scientific methods developed over many years without redevelopment of algorithms



Legacy code in Fortran, C, or other domain expert language

Object oriented processing components in Python 2.X

Framework components in Python 2.X and C++



ISCE Features

- Object orientation using Python
- A common hierarchical structure for defining components and applications
- An Image API & I/O API to abstract access to data and metadata
- Runtime polymorphism through factory pattern creation of components
- Provenance through xml metadata and python *pickle* mechanisms



Code Status and Availability

- Code supports strip mode data from
 - ERS-1, ERS-2, ENVISAT, RADARSAT-1, RADARSAT-2, Sentinel-1
 - JERS, ALOS-1, ALOS-2
 - TerraSAR-X, Cosmo-Skymed (CSK)
 - UAVSAR
- Unix/Linux build environment requires
 - gcc 4.7+, Python 3.3+, numpy, fftw 3.2, scons 2.0.1, motif
 - spiceypy (RadarSAT1 only)
 - gdal (RadarSAT2 only)
 - hdf5 (CSK only)
- UNAVCO is licensed to distribute the ISCE software to members of the WInSAR consortium
 - WInSAR members comprise a significant portion of the ISCE user base
 - WInSAR maintains a large database of InSAR raw data, as well as processing tools
- Individual research licenses can be provided by Caltech

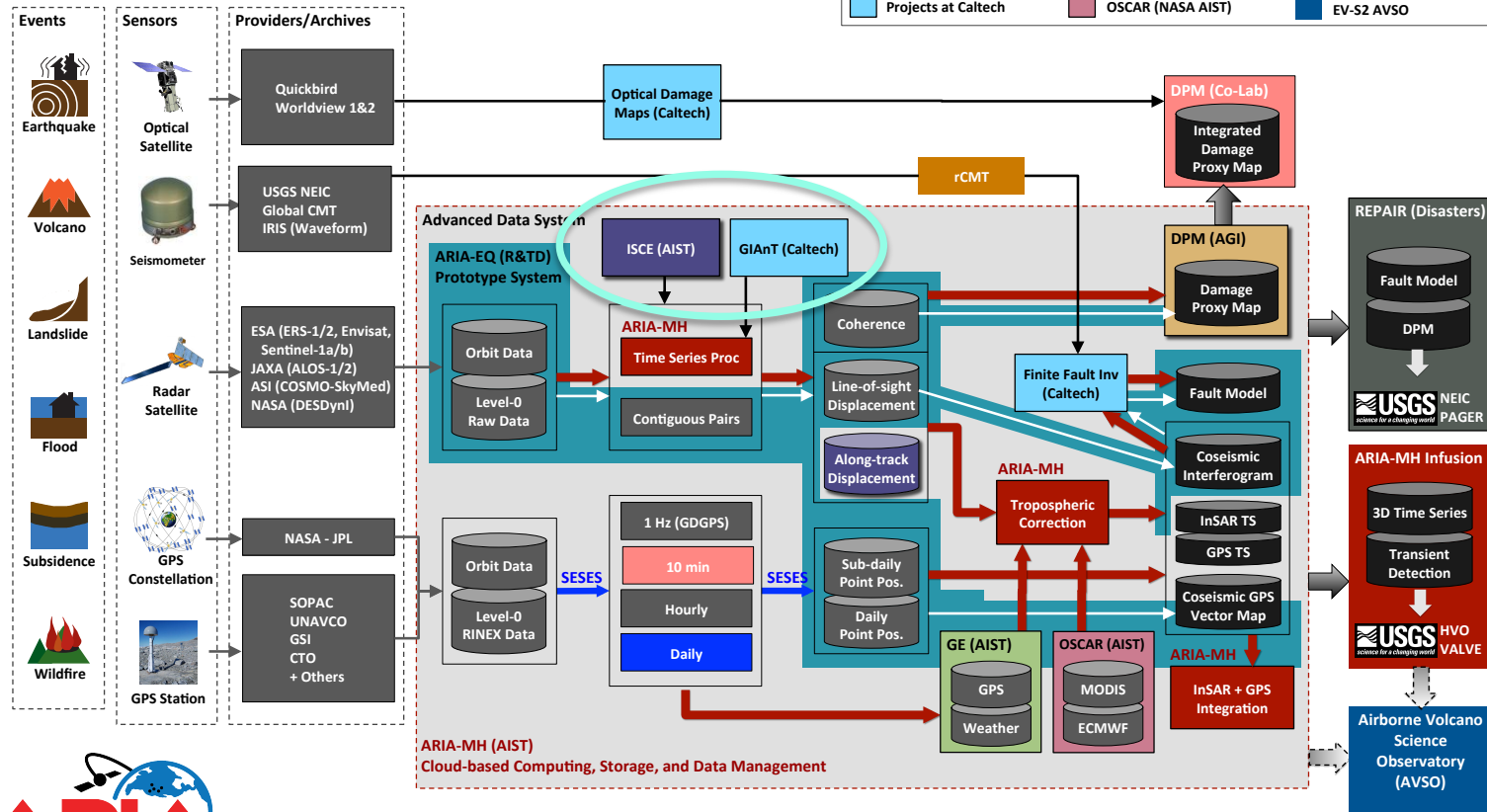


ISCE as the core of ARIA

Advanced Rapid Imaging Analysis system

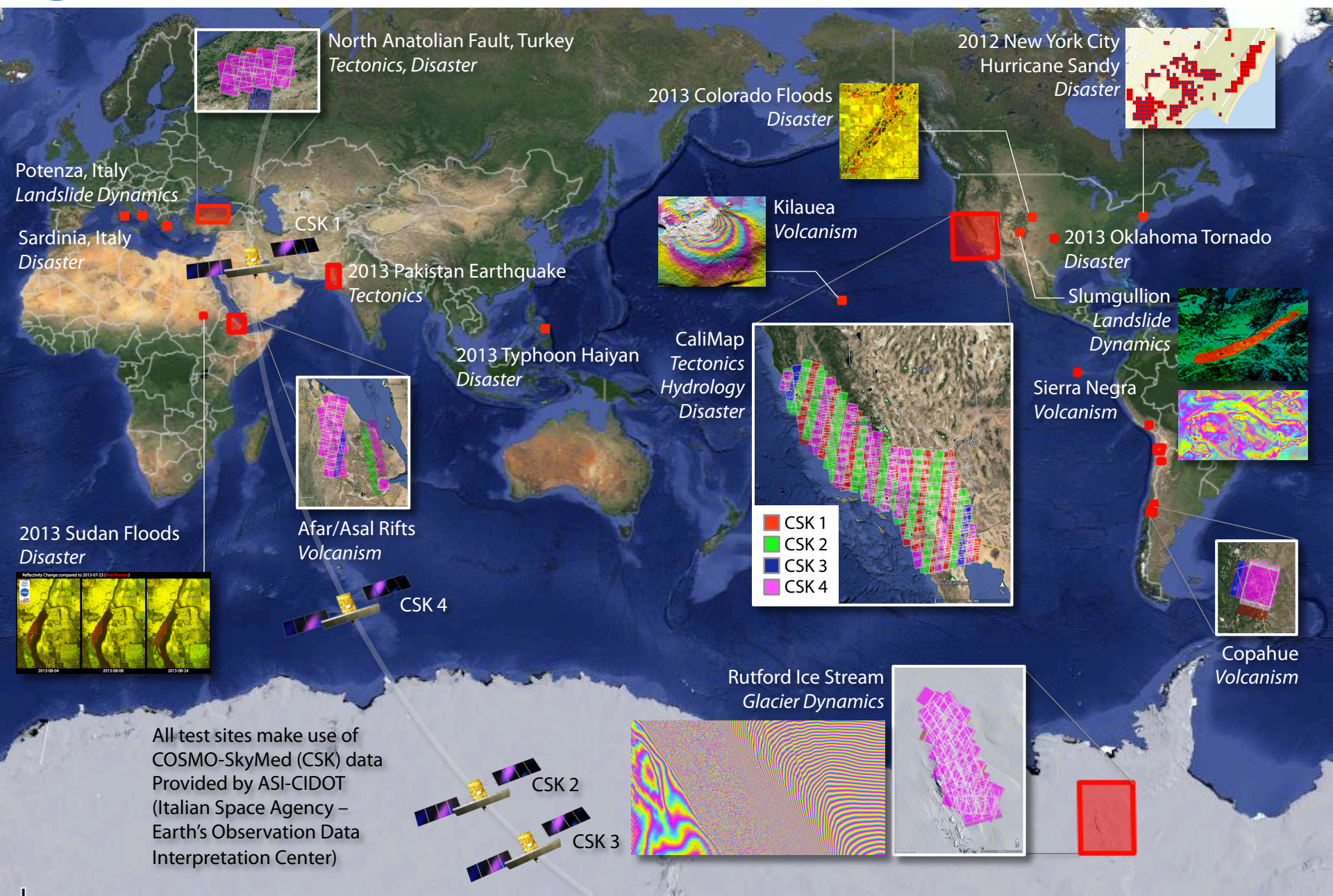
ARIA Projects 2013.07.22

ARIA-EQ (R&TD) is developing all connections in the blue regions, while ARIA-MH (AIST) will develop advanced functions for maturing the data system capabilities and enabling cloud-based hazard monitoring, data processing, management, and distribution. ARIA Co-Laboratory (PDF) is advancing low-latency GPS, optical satellite imagery and fault modeling integration.





JPL/Caltech – ASI/CIDOT Collaboration



North Anatolian Fault, Turkey
Tectonics, Disaster

2012 New York City
Hurricane Sandy
Disaster

2013 Colorado Floods
Disaster

Potenza, Italy
Landslide Dynamics

Kilauea
Volcanism

Sardinia, Italy
Disaster

CSK 1

2013 Pakistan Earthquake
Tectonics

2013 Oklahoma Tornado
Disaster

2013 Typhoon Haiyan
Disaster

CaliMap
*Tectonics
Hydrology
Disaster*

Slumgullion
*Landslide
Dynamics*

Sierra Negra
Volcanism

2013 Sudan Floods
Disaster

Afar/Asal Rifts
Volcanism

CSK 4

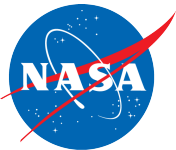
Rutford Ice Stream
Glacier Dynamics

Copahue
Volcanism

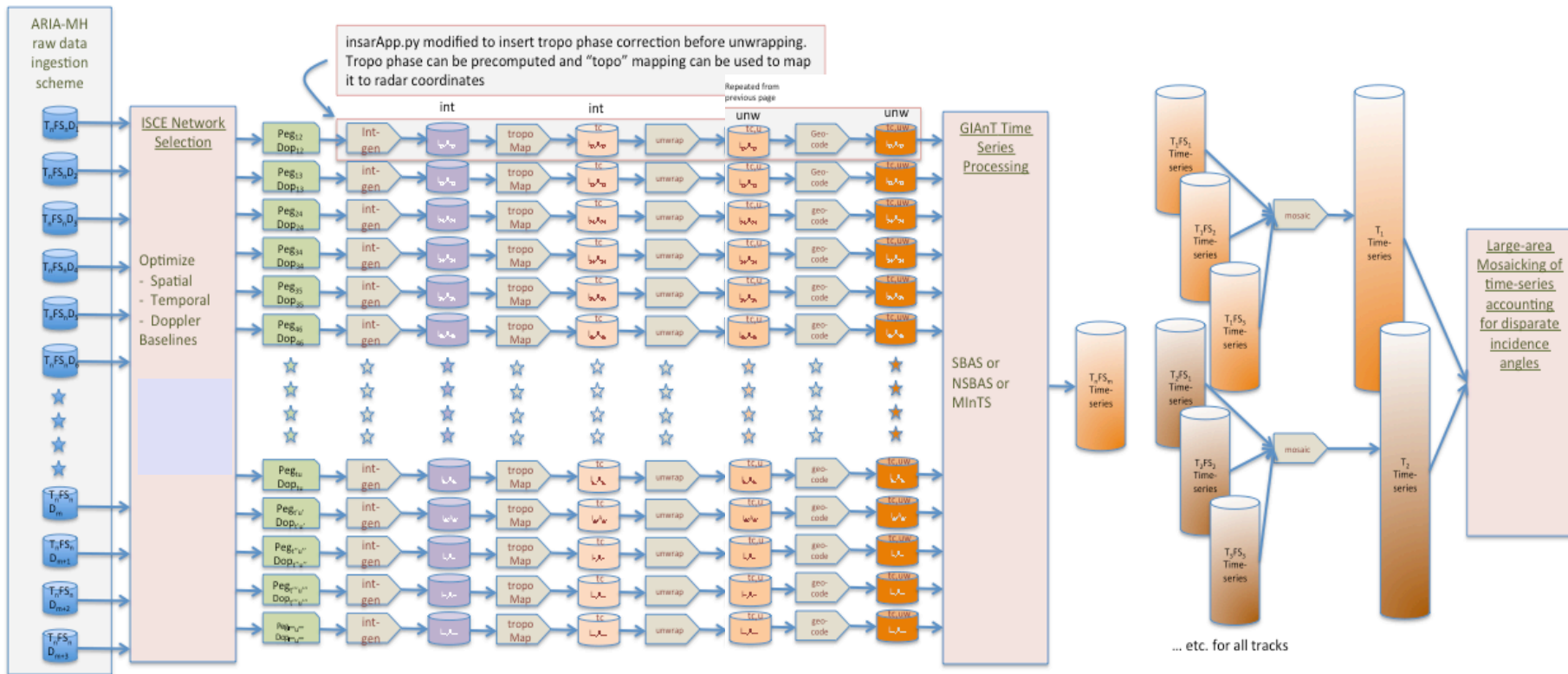
All test sites make use of
COSMO-SkyMed (CSK) data
Provided by ASI-CIDOT
(Italian Space Agency –
Earth's Observation Data
Interpretation Center)

CSK 2

CSK 3



Automated InSAR Processing Architecture for Large Scale Analysis

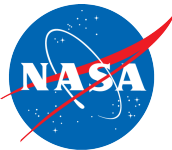


ARIA data system uses

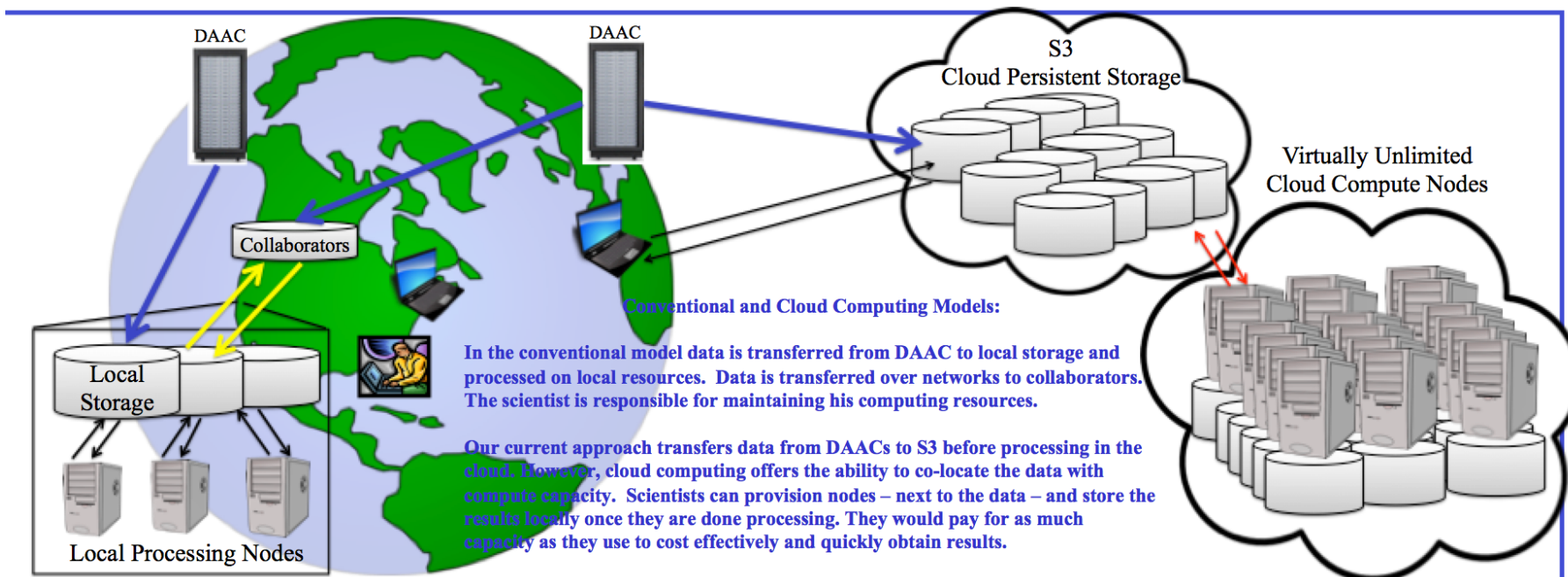
- InSAR Scientific Computing Environment (ISCE) software
- GIANT software for time series analysis

To generate

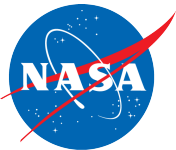
- Interferograms
- Tropospheric corrected interferograms
- Velocity maps



CESCRE – Cloud Enabled Scientific Collaborative Research Environment (Mark Powell, PI)



- CESCRE developed a framework for processing data and sharing results on the Amazon cloud
- ISCE was the fundamental test environment for CESCRE collaboration
- CESCRE allows a user to sit at their laptop and harness the full power of the amazon cloud



Earthkit Tutorials for ISCE/GIAnT

Mark Powell, PI

EarthKit Lab beta Home egurrola

12.2. GIAnT: UAVSAR Inversion Launch

EarthKit: Getting Started

Start here to learn how to use the EarthKit self-led lab system

[Start Lab](#)

1. Getting started with ISCE

Getting started with ISCE hands-on

[Start Lab](#)

1.1 Number formats in ISCE

A reference document for ISCE number formats

[Start Lab](#)

2. Using MDX

Using the visualization tool MDX to analyze interferograms

[Start Lab](#)

3. Using insarApp.py

Processing Interferometric Data Sets using insarApp.py - Overview

[Start Lab](#)

3.1. Processing ALOS PALSAR

Learn to generate interferograms from ALOS PALSAR data

[Start Lab](#)

3.1.1. ALOS input file examples

A deeper look into authoring input for ALOS data processing

[Start Lab](#)

3.2 Files and Formats

Understanding output files and formats

[Start Lab](#)

3.3 insarApp options

Exploring the insarApp.py processing option space

[Start Lab](#)

3.4. SAR Data Access

Running SSARA to query and download data

[Start Lab](#)

4. Processing ERS

Learn to generate interferograms from ERS data

[Start Lab](#)

4.1. ERS input file examples

A deeper look into authoring input for ERS data processing

[Start Lab](#)

5. Processing EnviSAT

Learn to generate interferograms from EnviSAT data

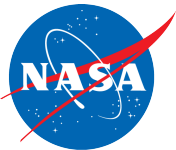
[Start Lab](#)

6. Processing COSMO-SkyMed

Learn to generate interferograms from COSMO-SkyMed data

[Start Lab](#)

- Extensive, cloud-based, interactive tutorials for learning about ISCE and GIAnT



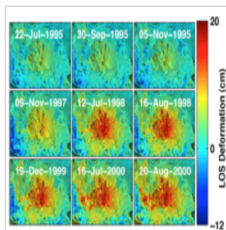
Earthkit Tutorials for ISCE/GIAnT



7. Processing from SLC

Processing from SLC: COSMO-SkyMed, TerraSAR-X, RadarSAT-2, and others

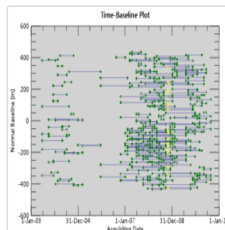
[Start Lab](#)



8. Stack processing

Learn how to process stacks of Interferograms for GIAnT

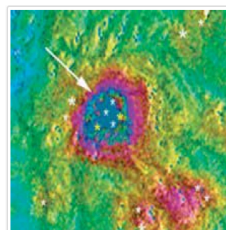
[Start Lab](#)



9. GIAnT: Understanding Inputs

A look at the inputs used in time series analysis with GIAnT

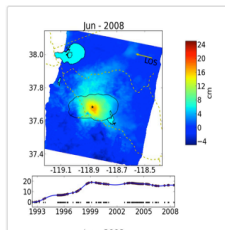
[Start Lab](#)



9.1. GIAnT: Reading the Stack

A deeper look at GIAnT time series analysis

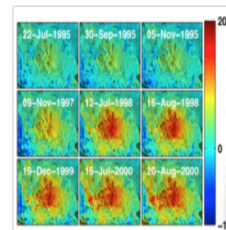
[Start Lab](#)



9.2. GIAnT: Time Series Inversion

Running the time series inversion with GIAnT and looking at the results

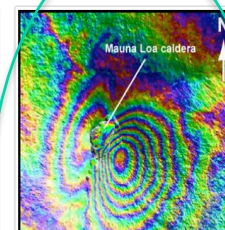
[Start Lab](#)



10. Polarimetric Processing

Polarimetric UAVSAR Data Processing for Land-cover Land-use Change Applications

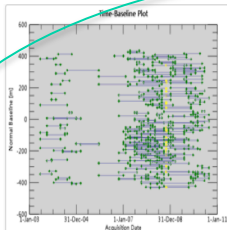
[Start Lab](#)



11. Post-processing Stacks

Post-Processing UAVSAR Stack data with IsceApp.py

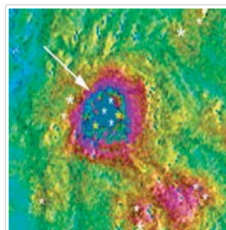
[Start Lab](#)



12. GIAnT: UAVSAR Understanding Inputs

A look at the inputs used in time series analysis with GIAnT

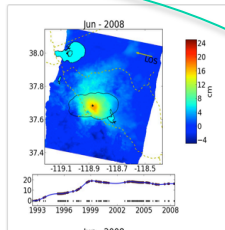
[Start Lab](#)



12.1. GIAnT UAVSAR reading the stack

A deep dive into GIAnT time series analysis

[Start Lab](#)

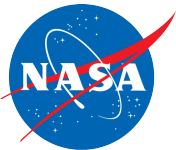


12.2. GIAnT: UAVSAR Inversion

Running the time series inversion with GIAnT and looking at the results

[Start Lab](#)

- Including new modules for UAVSAR stack processing



Earthkit Tutorials for ISCE/GIAnT



EarthKit Lab beta

Home

egurrola ▾

11. Post-processing Stacks

Remote Desktop

SSH

Shut down

1. Post-Processing UAVSAR Stack data with isceApp.py

1 / 6 ▾

In this lab, you will learn how to process UAVSAR Stack data, while learning about the ISCE application `isceApp.py`. Previous labs have used the application `insarApp.py` to process pairs of raw or single look complex data acquired on two different dates using spaceborne sensors into interferograms and geocoded products. In this lab we will work with the application `isceApp.py` to post-process several data sets from the same flight track acquired at different times with the UAVSAR radar flown on an airplane. The data downloaded from the UAVSAR website have already been processed to single look complex images by the UAVSAR team. Post-processing includes the following steps: forming interferograms from the slc data, removing topographic phase, filtering, unwrapping, and geocoding.

To get started change directory to the `/data/lab11` directory (click the "Launch" button if you haven't already done so),

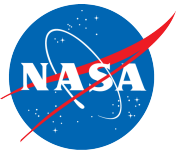
```
> cd /data/lab11
```

Take a look at the directory contents with the "ls -l" command,

```
> ls -l
demLat_N38_N39_Lon_W123_W121.dem.wgs84.xml
incoming -> /data/sites/Napa_uavsar_stack/incoming
isceApp.xml
precooked -> /data/sites/Napa_uavsar_stack/
SanAnd_05510_01_BC.dop -> incoming/SanAnd_05510_01_BC.dop
SanAnd_05510_12128_000_121105_L090HH_01_BC.ann ->
incoming/SanAnd_05510_12128_000_121105_L090HH_01_BC.ann
SanAnd_05510_12128_000_121105_L090HH_01_BC_s1_1x1.slc ->
incoming/SanAnd_05510_12128_000_121105_L090HH_01_BC_s1_1x1.slc
SanAnd_05510_13089_001_130508_L090HH_01_BC.ann ->
incoming/SanAnd_05510_13089_001_130508_L090HH_01_BC.ann
SanAnd_05510_13089_001_130508_L090HH_01_BC_s1_1x1.slc ->
```

```
puts.NUMBER_AZIMUTH_LOOKS = 16
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw.in
puts.MAXIMUM_LONGITUDE = -121.88182067871094
#####
#####
2014-10-28 18:55:52,321 - isce.isceProc.runGeocode - INFO -
#####
#####
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw
w - Outputs
-----
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw.ou
tputs.LONGITUDE_SPACING = 0.00027777777777777778
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw.ou
tputs.MINIMUM_GEO_LONGITUDE = -122.40805555555556
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw.ou
tputs.LATITUDE_SPACING = -0.00027777777777777778
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw.ou
tputs.MAXIMUM_GEO_LATITUDE = 38.07138888888889
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw.ou
tputs.MAXIMUM_GEO_LONGITUDE = -121.88166666666667
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw.ou
tputs.GEO_LENGTH = 1408
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw.ou
tputs.MINIMUM_GEO_LATITUDE = 38.46222222222222
runGeocode.uav10_uav11_hh../uav10_uav11/uav10_uav11_hh.filt_topophase.unw.ou
tputs.GEO_WIDTH = 1896
#####
#####
> █
```

- Guided training through browser interface
 - Integrated step-by-step instructions and a Unix command line prompt to play with code and data.
 - No installation of code and software dependencies



UAVSAR Enhancements for ISCE



ESTO supported enhancements in ISCE to enable scientists to work with UAVSAR products:

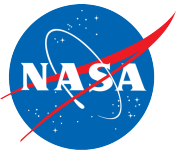
- post-process SLC images (provided by the UAVSAR project) into stacks of interferograms with options to filter, unwrap and geocode.
- enable scientists to use the ISCE framework and components to create work flows needed for their work with the data.



ISCE for the NASA-ISRO SAR (NISAR) Mission

- The proposed NISAR mission plans to use the ARIA infrastructure for elastic computing of NISAR data to L0B (raw), L1 (Images), and L2 (interferometric and polarimetric) data levels
- ISCE will be augmented by the project to incorporate new functionality that would be needed for NISAR
 - Image focusing processor for the proposed NISAR mission
 - Calibration/Validation tools
 - Science algorithms for biomass, deformation, time-series analysis, and mosaicking
 - Integrated tutorials and training modules
- This functionality will be available to individual scientists on their local computing devices or through cloud services

Pre-decisional - for Planning and Discussion Purposes Only



Vision for Future Development of ISCE

- The ISCE framework has been designed with flexibility and extensibility that can work in a variety of contexts.
- ISCE can support the community at many levels
 - Individual users working on their laptop or local computing system
 - Users interacting with the cloud through VMs and staging methods for data and processor deployments
 - On-demand processing through web-based queries to a central server
 - Project-sponsored code core for production of multi-petabyte data sets
- Coordinated development across these levels will ensure a robust, community vetted science tool for NASA science – from ROSES to missions like the proposed NISAR

Pre-decisional - for Planning and Discussion Purposes Only
