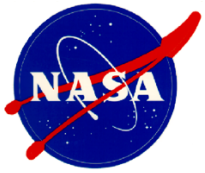
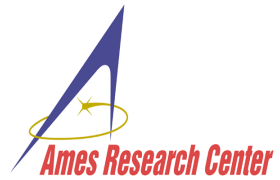




Anomaly Detection and Analysis Framework for Terrestrial Observation and Prediction System (TOPS)

ESTF 2011

June 21, 2011

Petr Votava

Ramakrishna Nemani

Andrew Michaelis

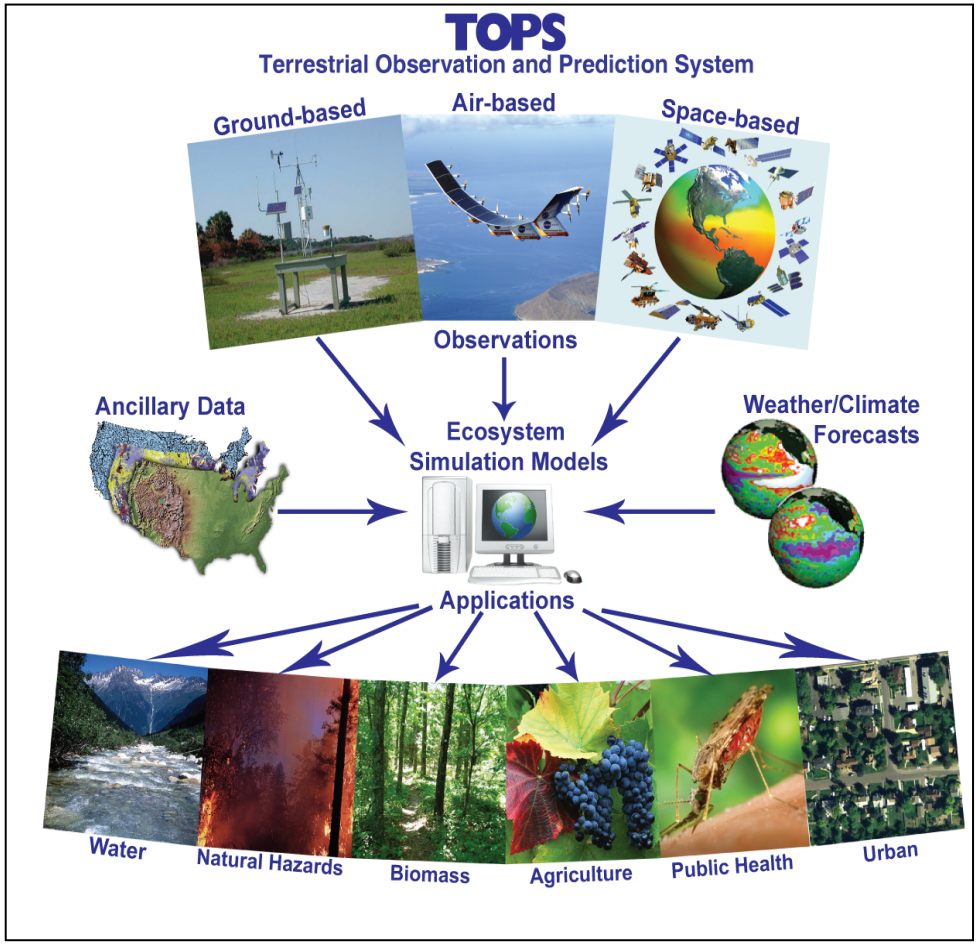
Hirofumi Hashimoto

Project Goal

Provide a framework for automated anomaly detection in large heterogeneous Earth science data sets as well as on-demand data analysis integrated with the Terrestrial Observation and Prediction System (TOPS)

Terrestrial Observation and Prediction System (TOPS)

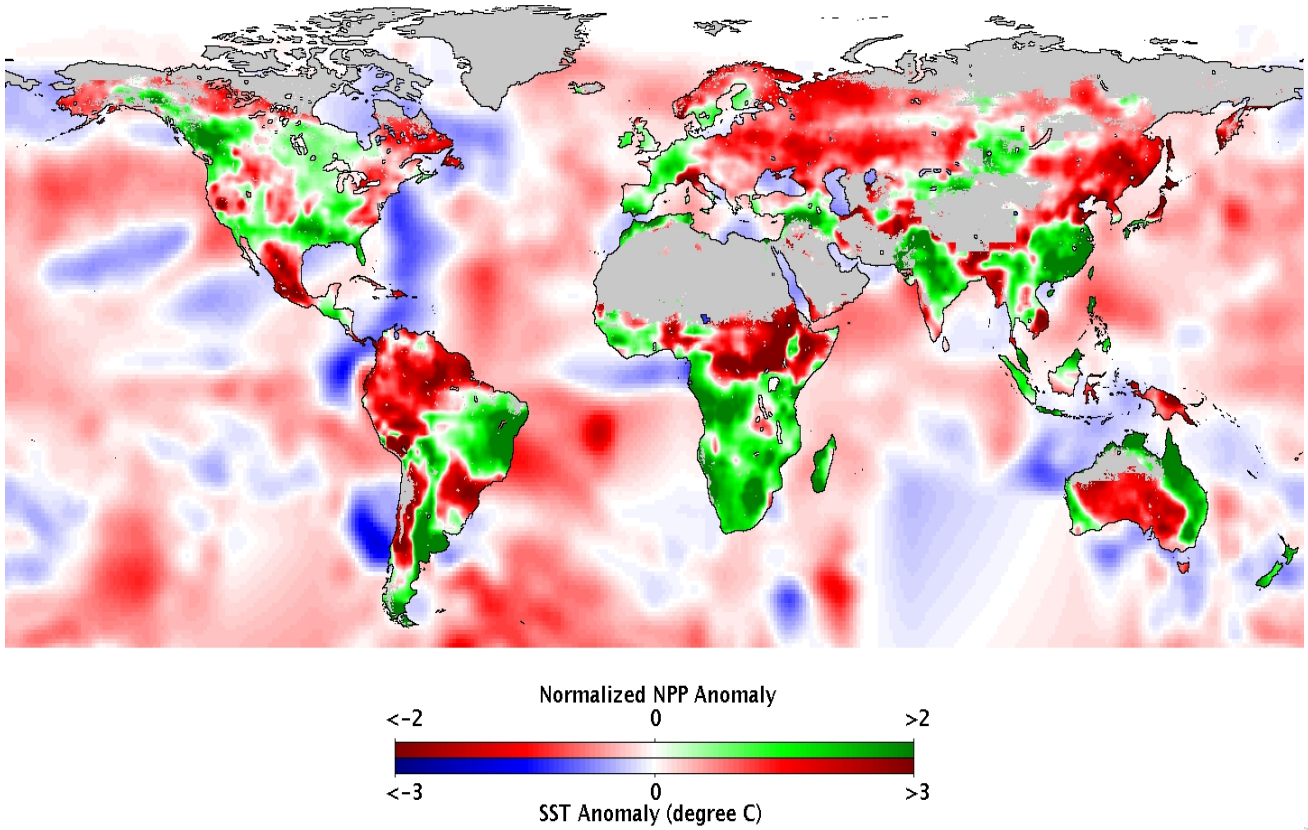
- An ecological forecasting system for developing nowcasts and forecasts of ecosystem conditions for use in a range of applications.
- Data and modeling software system designed to integrate data from satellite, aircraft, and ground sensors with climate and application models.
- Ecological Forecasting (EF) predicts the effects of changes in the physical, chemical, and biological environments on ecosystem state and activity



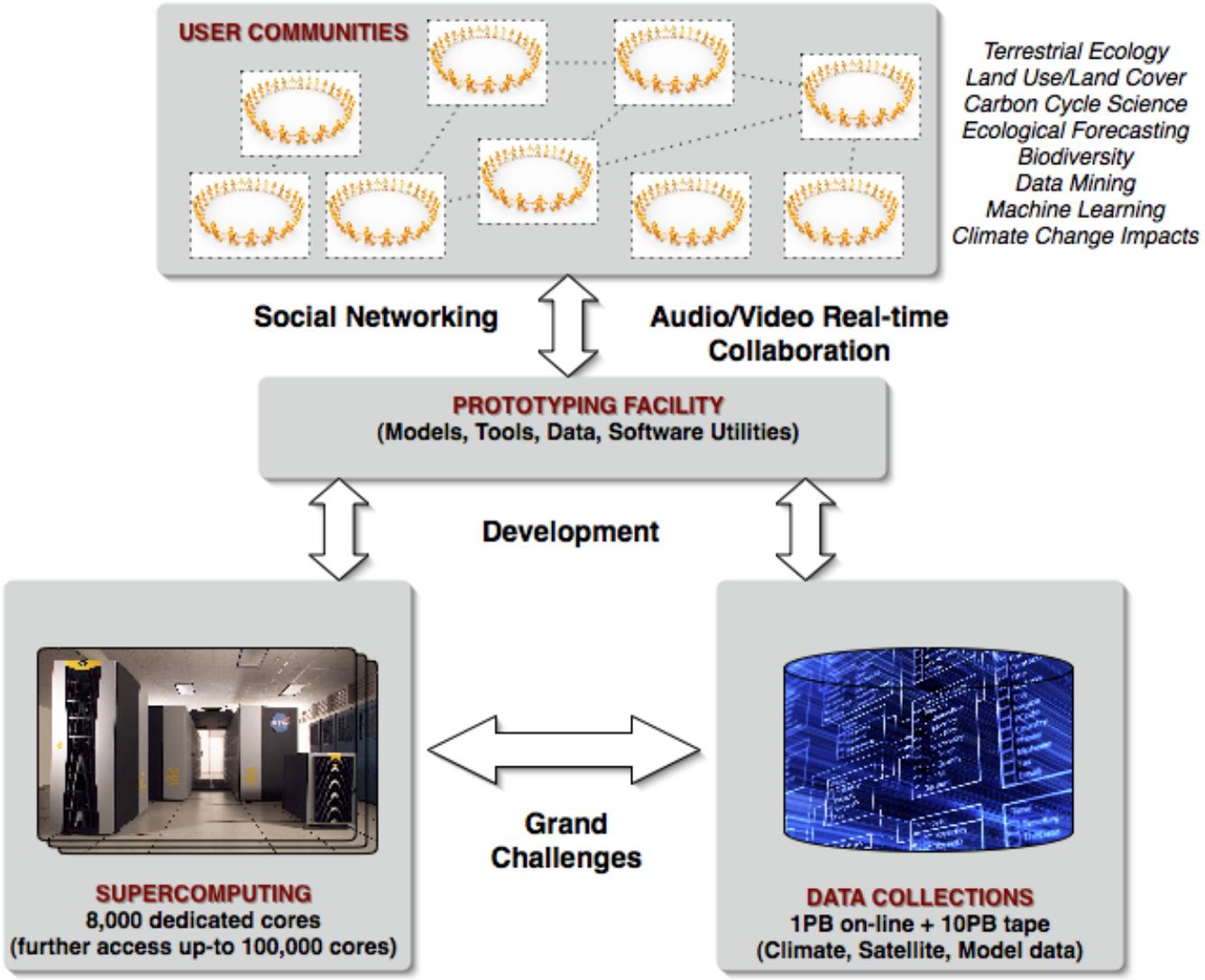
Near-realtime Monitoring of Global NPP Anomalies

Mapping changes in global net primary production

depiction of the droughts in the Amazon and Horn of Africa, May 2005



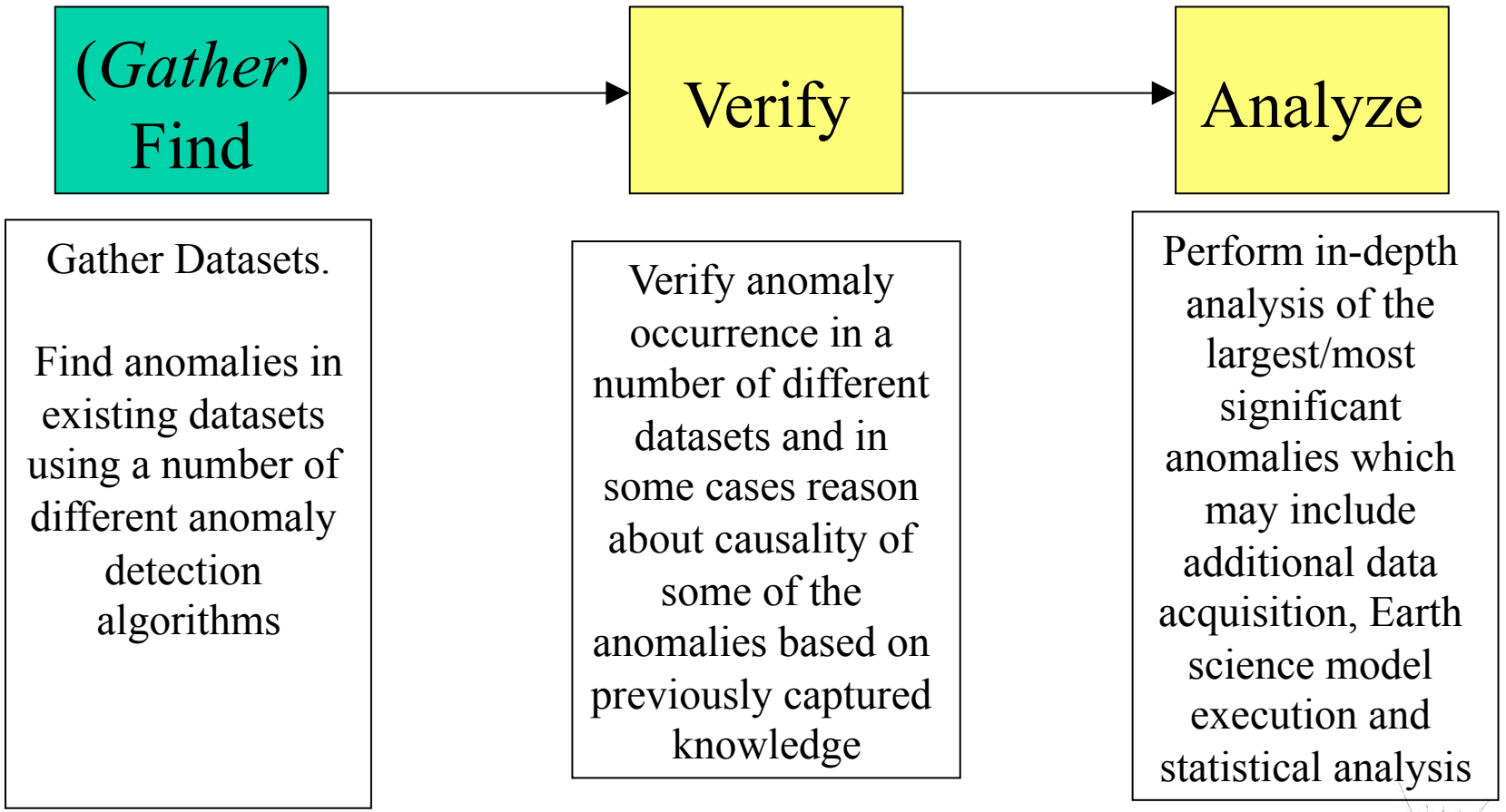
Scaling things up - NASA Earth Exchange (NEX)

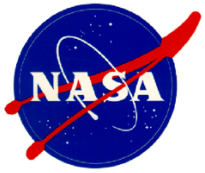


Ames Research Center

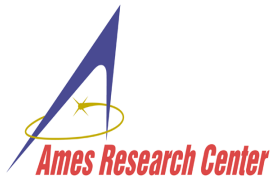
ESTO
Earth Science Technology Office

Anomaly Detection System Overview





Key System Components



- Knowledge Base
  - Capture information on data/model hierarchies and direct and indirect relationships
  - Ancillary data on observed events
- Anomaly Detection Framework
  - Plug-in infrastructure for anomaly detection algorithms
- Verification module
  - Provides logic for re-running analysis on related datasets
- Analysis module
  - Management of anomaly analysis workflows

Anomaly Detection Framework Objectives

1. Develop a flexible and extensible framework for integration of a number of anomaly detection algorithms operating on large volumes of Earth science datasets.
2. Integrate several existing anomaly detection algorithms with the framework.

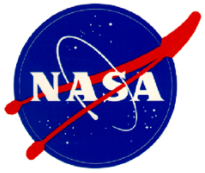
Infrastructure for deploying anomaly detection algorithms on multivariate spatio-temporal datasets from the climate change and ecosystem domains.

Anomaly Detection Framework

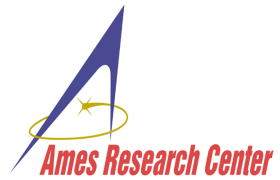
- Three main components
  - Data
    - Historical data
    - Current observations
    - Baseline pre-processed climatologies
    - Ancillary data (development, city boundaries, …)
  - Framework Infrastructure
    - Software that integrates data with algorithms in a flexible way (minimize coding requirements)
  - Algorithms
    - Performs the anomaly detection tasks

Data

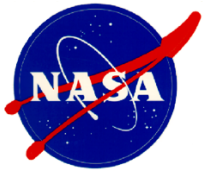
- Most important and most time-consuming component of the TOPS system
  - In our past experience, it takes about 80% of the time to put the baseline datasets together before a study can be executed
  - Baseline data must be cleaned up and often composited (remove clouds, select better quality pixels etc.) in order to be to be useful
    - The decision on how to do the compositing is **different for each dataset** and involves QA processing, filling missing data, statistical analysis, and provenance
    - This process can be considered "training" as we create a baseline typical view of the dataset, so that it can be used to detect outliers
- We have on-demand acquisition of global data from MODIS, AMSR-E and Landsat
- Currently have all MODIS land data, over 1 million Landsat scenes and many other datasets



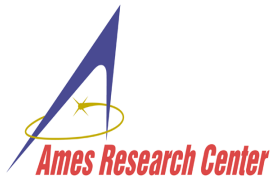
Framework Infrastructure



- Key requirements
  - Manages number of anomaly detection algorithms
  - Manages number of models produced during the learning phase of the algorithms
    - Models are representations of typical state of the environment from which disturbances can be distinguished
  - Facilitates execution of the algorithms
  - Enables integration of new algorithms



Implementation Language



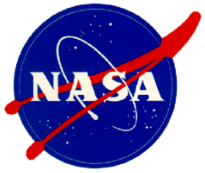
- We selected Python as our language of choice for the framework implementation for several reasons:
  - It provides a very rapid prototype environment, while also being fairly stable for production environment with good exception/error handling capabilities and good performance
  - Most of the TOPS team uses Python on daily basis
  - Over 50% of the TOPS production code is already in Python and more is being currently converted, so it will provide for an easy integration with the rest of the TOPS system and utilities

Algorithm Integration

- The framework will integrate algorithms in most languages through command-line wrappers and it can integrate algorithms written in Python directly through their APIs
- In order to integrate new algorithm into the system, the wrapper must extend and abstract class called *DetectionAlgorithm, and* implement several methods - *__init__(), set_parameters(), run() and get_anomalies()*
  - This determines the steps taken during the algorithm initialization
  - This is also where the algorithm specific parameters are set and it is executed
  - The execution itself can either call a command line implementation, or execute an externally supplied Python module through the module's API
- Additionally if there are algorithm-specific requirements for data formats, the data conversion should be part of the wrapper
  - TOPS provides a large number of tools and utilities that are already running in operational environment to facilitate many data transformations
  - However, due to the unknown and non-unified characteristics of number of external anomaly detection algorithms, parts of the integration will be manual

Integrated Algorithms

- **Spatial**
  - *Spatial anomaly from climatology*
  - *Normalized anomaly*
    - Models/baselines are climatologies and long-terms statistics for the specified region
- **Time-series**
  - *Inductive Monitoring System (IMS)*
    - Model is a set of vectors representing corners of N-dimensional hypercube created during the learning phase
    - Good for on-line/real-time anomaly detection
  - *OrcaExpress*
    - Fast distance-based outlier detection in N-dimensional space
    - No prior model is built



Other Considerations

- MODIS Global Disturbance Algorithm
  - Land surface temperature and vegetation anomaly detection
- Recursive Merging
  - For landcover change detection
- Distributed Orca
  - Improved performance using peer-to-peer algorithm across number of computing nodes

Knowledge/Anomaly Base

- Keep track of top anomalies discovered during execution
- Keeps track of "known events" so that they can be used to filter out observed anomalies
  - Fires
  - Development
  - Other Landcover change data
  - List of text reports from NOAA related to extreme events

NOAA Reports

- Monitor NOAA advisory feeds
- Extract location and event(s)
  - Using Natural Language Processing techniques
  - Using Lingpipe and NLTK + custom components
  - Combine locations with our location hierarchies component so that we can aggregate data over multiple regions
  - In a way, this is a great source of anomaly/extreme events that could verify our observations, BUT it is a trickier to process than our satellite data

Knowledge Base Semantic Component

- Capture of data and model relationships such as similarity and compatibility so that different components can be dynamically selected and matched during analysis.
- OWL-based semantic descriptions
  - Simple relationship hierarchy (isDerviedFrom, hasImpactOn, …)
- Currently served by our Sesame server, but we're also looking at Parliament and Dydra (cloud-based solution)

Knowledge Base

- Currently in the middle of an update to reflect:
  - Tighter integration with the new release of SWEET
    - Will enable us to focus much more on data/event connections and knowledge capture rather than defining concepts
  - Addition of MODIS quality information for Terra and Aqua Land products
  - Possible addition of dynamic geo-information hierarchy
    - Staring from GCMD location taxonomy
    - Addition of Yahoo GeoPlanet on-demand location look-up and caching
  - Inserting verified anomalies themselves back to the KM system
    - Investigating switching to Parliament (open-source) or Dydra (cloud) infrastructure (from Sesame) to efficiently handle much larger anomaly ontology

VisTrails-based Workflow Management System

- Comprehensive provenance infrastructure for computational tasks
  - Data and Process Provenance
  - Results can be automatically reproduced
- Support for exploratory tasks in distributed collaborative environment
  - Collaborative design of scientific processes
  - Any point in the process can be independently explored
  - Interactive parameter space exploration
- Strong visualization component
- Easy integration with both legacy software and web services
  - Far more that service orchestration
  - Integrated entire Landsat processing pipeline in 1 day with added visualization and provenance capabilities!!

Process Design and Knowledge Capture

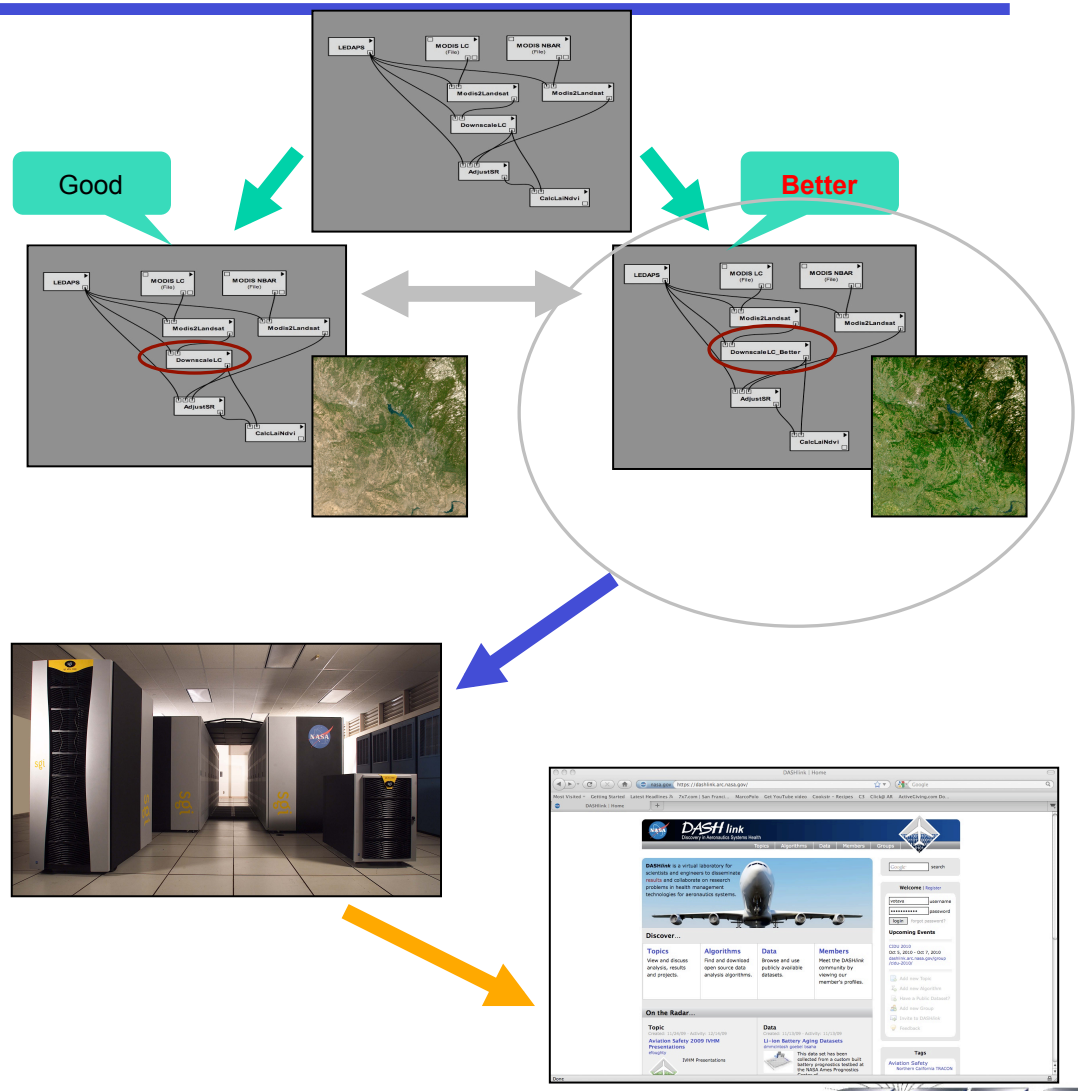
Ames Research Center

Process Design

Iterative Refinement/
Distributed Collaboration

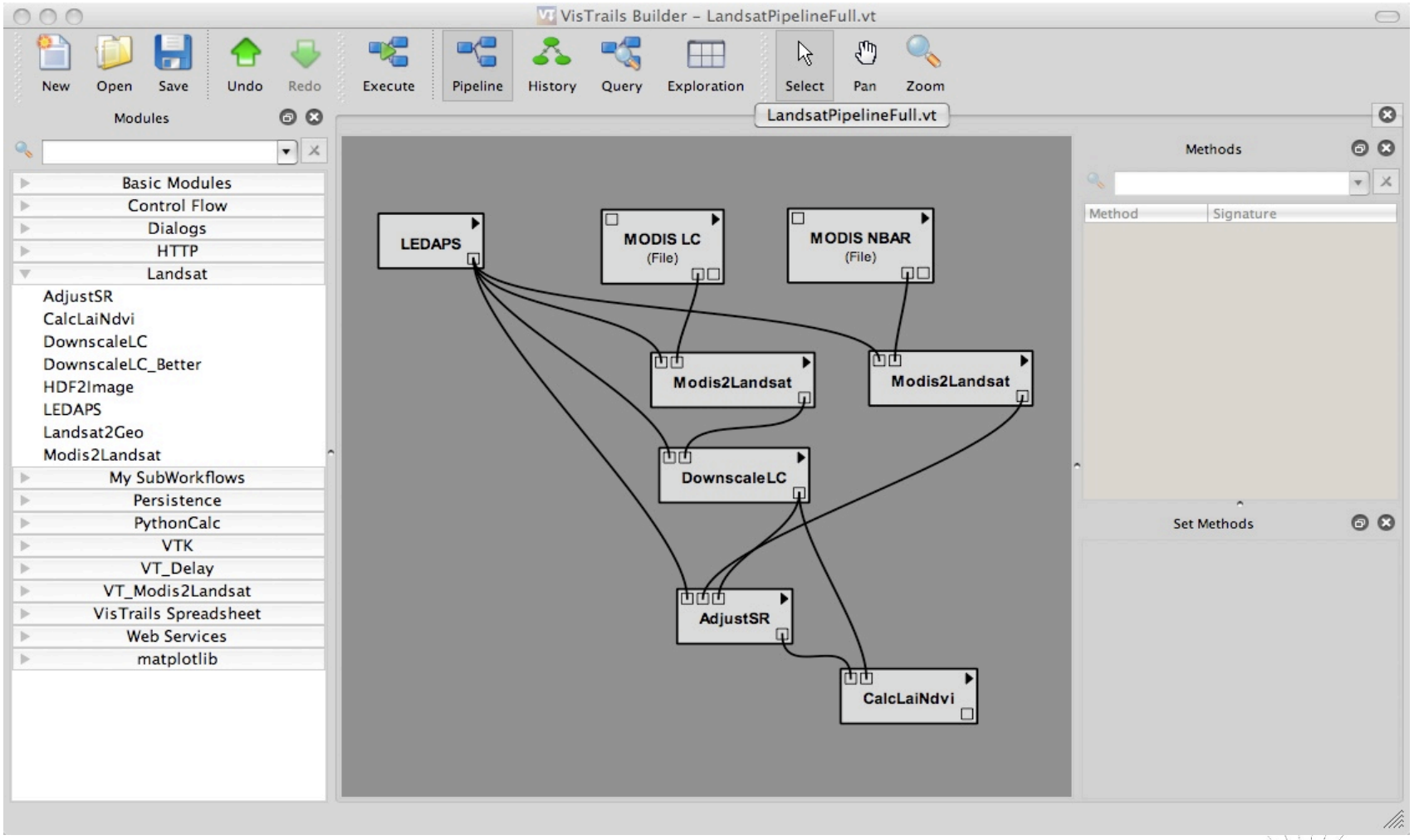
Full-scale Execution

On-Line Result Sharing and
Process Re-use

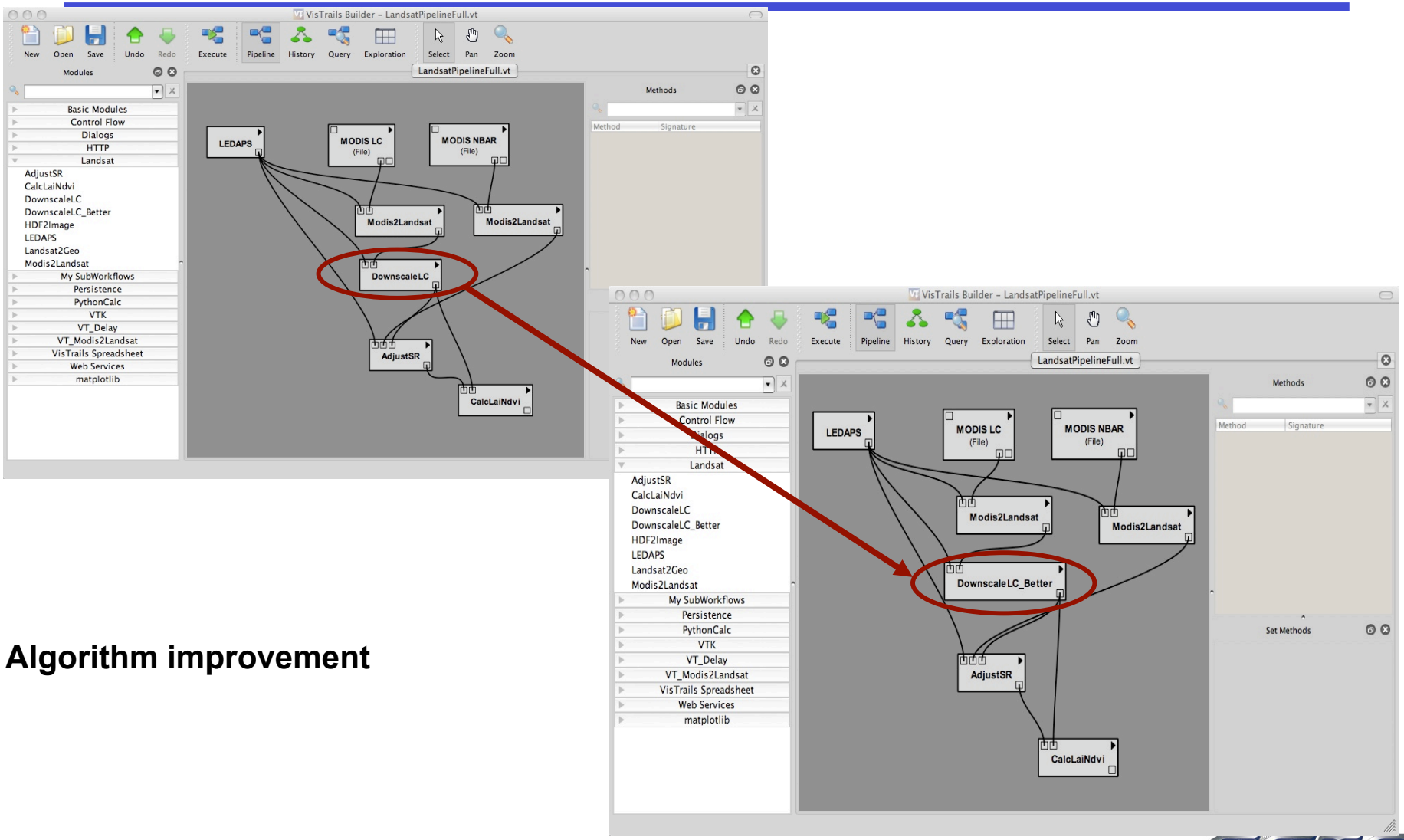


ESTO
Earth Science Technology Office

Design

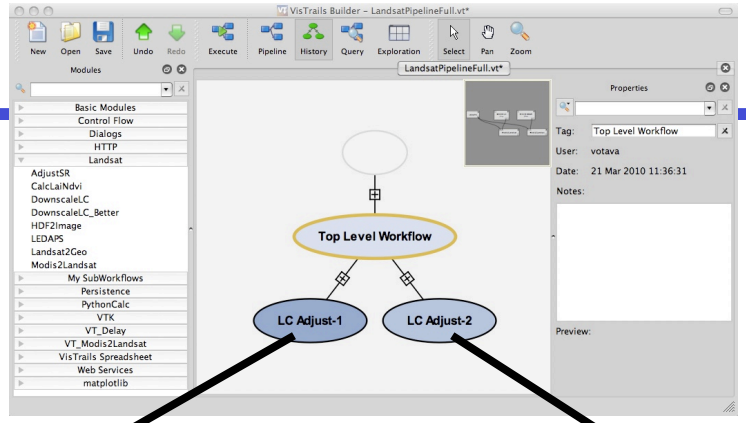

VisTrails Builder - LandsatPipelineFull.vt
New
Open
Save
Undo
Redo
Execute
Pipeline
History
Query
Exploration
Select
Pan
Zoom
Modules
Basic Modules
Control Flow
Dialogs
HTTP
Landsat
AdjustSR
CalcLaiNdvi
DownscaleLC
DownscaleLC_Better
HDF2Image
LEDAPS
Landsat2Geo
Modis2Landsat
My SubWorkflows
Persistence
PythonCalc
VTK
VT_Delay
VT_Modis2Landsat
VisTrails Spreadsheet
Web Services
matplotlib
LandsatPipelineFull.vt
LEDAPS
MODIS LC
(File)
MODIS NBAR
(File)
Modis2Landsat
Modis2Landsat
DownscaleLC
AdjustSR
CalcLaiNdvi
Methods
Method
Signature
Set Methods

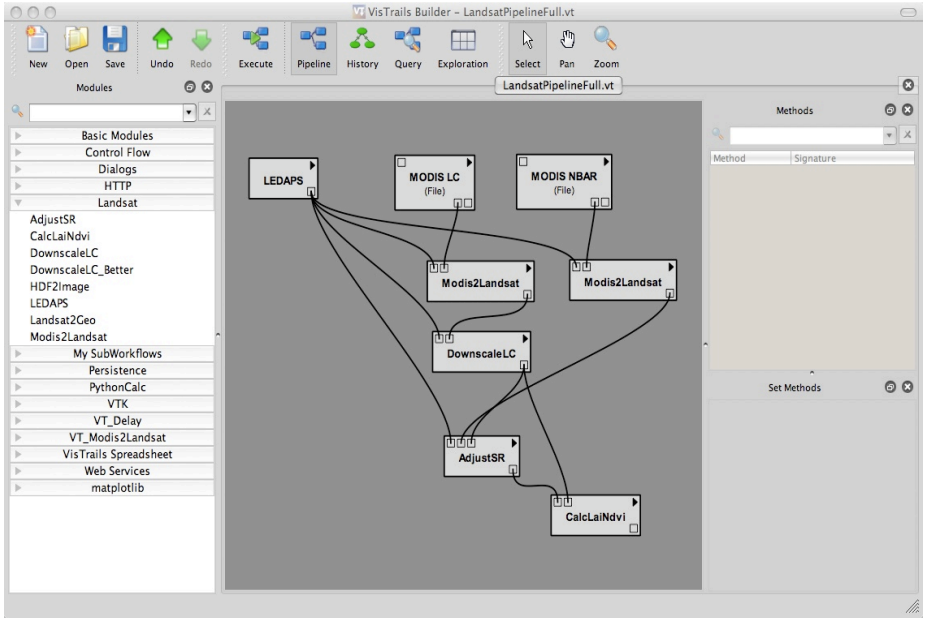
Iterative Refinement

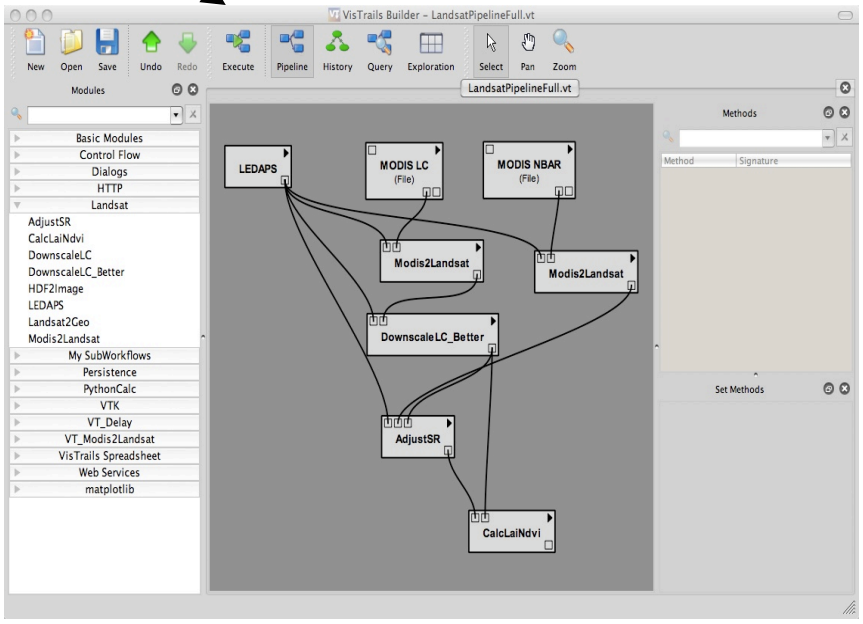


Algorithm improvement

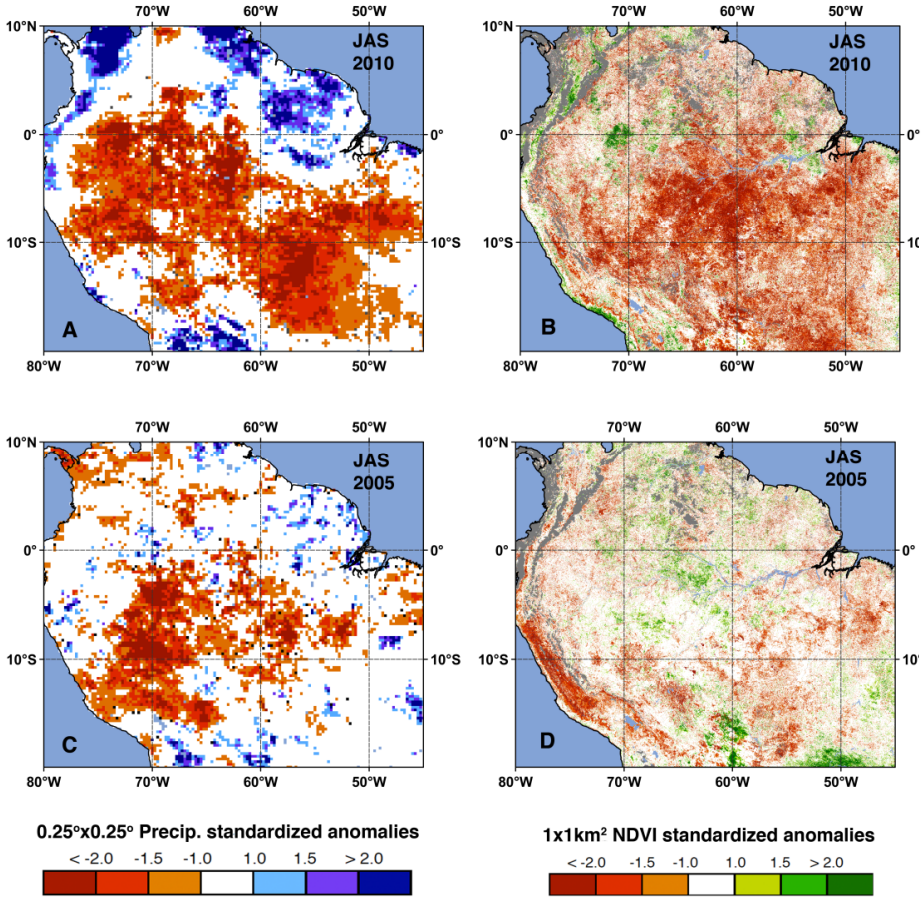
Track Process History





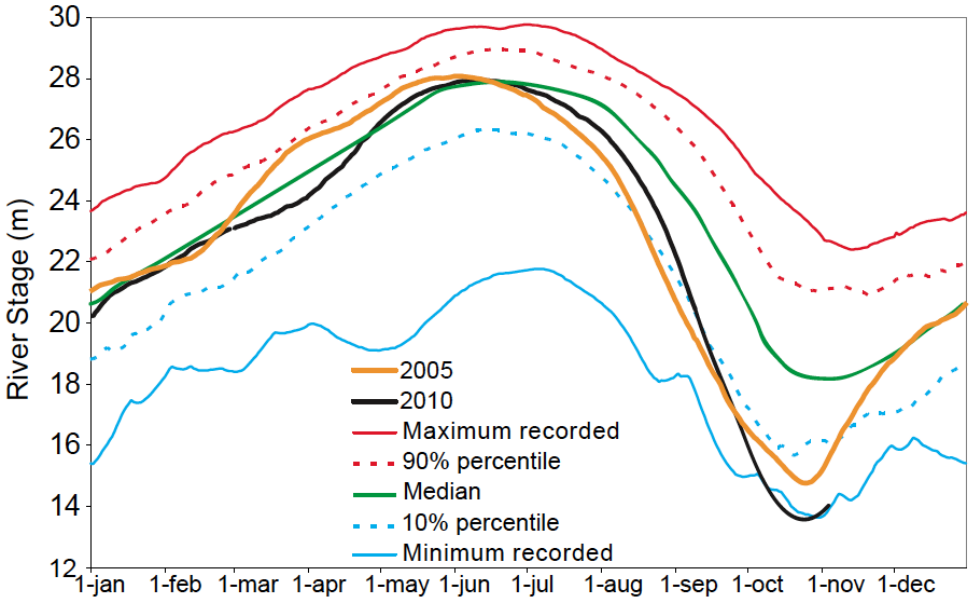


Use Case: 2010 Drought in the Amazon



TRMM MODIS

River stages of the Rio Negro near Manaus harbor



A brevia in review at Science

Data Needed for the Study

- River stage measurements
- Local weather sensor(s)
- Rain satellite estimates (TRMM)
- Vegetation satellite data (MODIS NDVI, EVI, LAI)
- Aerosols data (MODIS Aerosols)
- Fire data (MODIS fire)
- Overall Volume of data = 130GB
  - The project goal is to be able to monitor anomalies and if necessary repeat the analysis in over 30,000 places world-wide

Analysis Process

- Start with river stage time-serie
  - It shows that in 2010 it is the driest year during the dry part of the season
- Verify
  - Process TRMM data from 1998 to 2010 and look at anomalies from August, Sept, Oct
  - Next Look at vegetation from MODIS (NDVI, EVI, LAI)
    - Resolve inconsistencies through knowledge about different products response to aerosols
  - Verify increased levels of aerosols from satellite data
    - Big reason for aerosols (fires)
  - Verify fire distribution over the region during that time
  - Must consider QA of all the products

Analysis

- Much thought went into the scientific process
- For 2005 season (the first study), it took almost 2 years to complete thorough study
- In 2010 we were able to do it in less than 20 days
- The goal of this project is to greatly speed up the first-time analysis by automating much of the process and then being able to replicate the study around the world even with different datasets
- We should be able to execute this particular study over thousands of measuring stations worldwide

Putting It All Together (Amazon2010 Use Case)

1. TOPS periodically runs anomaly detection algorithm on a list of river level measurements
2. System runs the timeseries algorithm and returns a list of anomalies.
3. System runs the same algorithm on a similar dataset (for example using precipitation from a nearby station because it relates to water level in the river)
4. Anomalies from both sources are compared and ones that appear in both sets for the specific time-period are selected.
5. Additional verification can be performed using satellite TRMM rain data.
6. The discovery of the anomaly, its magnitude and location may trigger a pre-defined analysis workflow
   - The workflow can have number of steps from on-demand data acquisition to Earth science model execution
7. If no workflow is found, the system may try to reason for a selected number of steps if possible (and info available in the *Knowledge Base*)
   - Rain has effect on vegetation so, vegetation in the area should also show signs of disturbance from normal
8. Finally a report is generated detailing actions taken during the entire process that will contain preliminary provenance information about each of the anomalies

Challenges

- **Provenance and versioning (somewhat out of scope, but super important)**
  - We are looking at a better way to provide versioning between models and algorithms. Additionally, the versioning represents an important part of the data and process provenance that we want to track throughout the entire process. We are also looking at best ways to formalize this provenance information.
- **Combination of spatial and time-series data**
  - The Amazon2010 is a good case to try to work it out. We are considering multiple levels of statistical hierarchies at multiple resolutions (e.g. 1/2 degree, 1 degree etc.). Information can be kept in a databases and easily aggregate on-demand. The challenge is that this is also product dependent, for example the fire product aggregation will look different than vegetation anomalies.

Future Applications

- Climate QA
  - We plan to use this framework for providing anomaly detection as a part of our QA process during climate gridding
  - Daily execution on 30,000 stations worldwide
  - Will provide important improvement in the quality of our gridded climate datasets - the most requested data from our group (distributed over 10TB in 2010)
- NASA Earth Exchange
  - Plan to integrate the system as a part of the NASA Earth Exchange (new NASA collaboration platform), where it will be available for testing of new algorithms on over 1PB volumes of Earth science data.

Project Website

https://c3.nasa.gov/nex/projects/2/

You can join, get updates, documents, participate in the discussions and more!!