

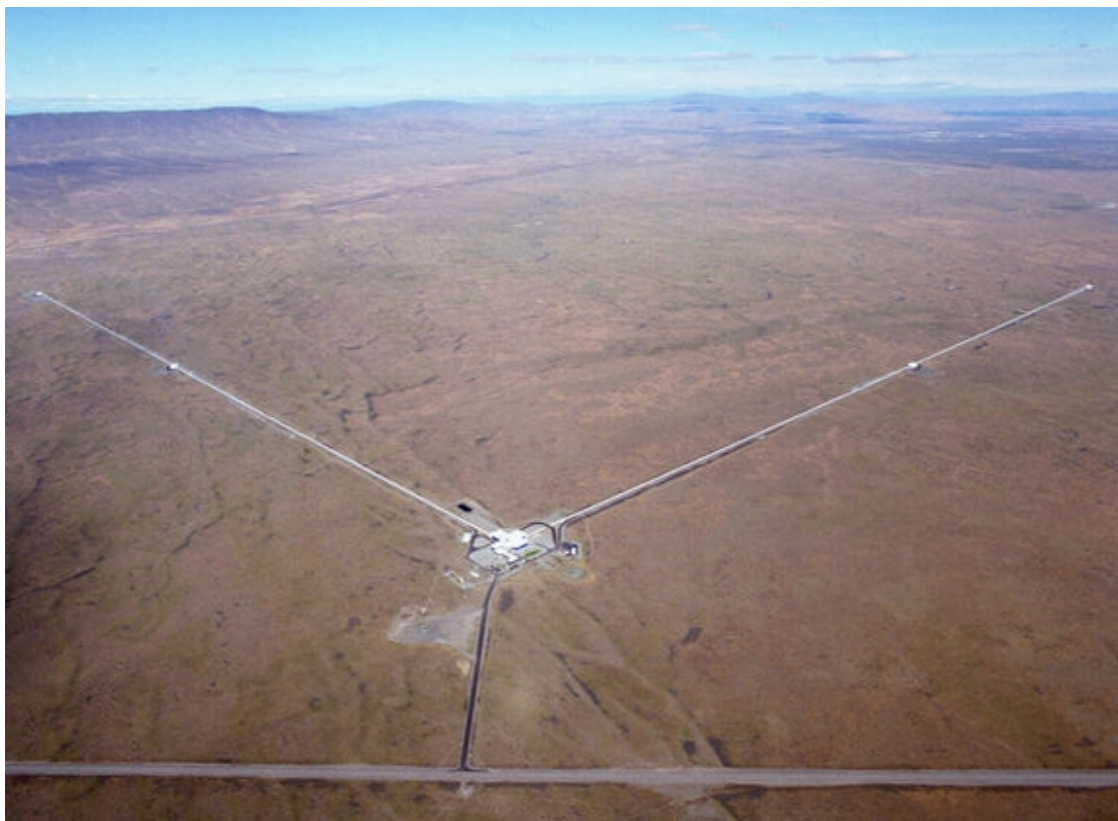
---

# LIGO's Use of Condor

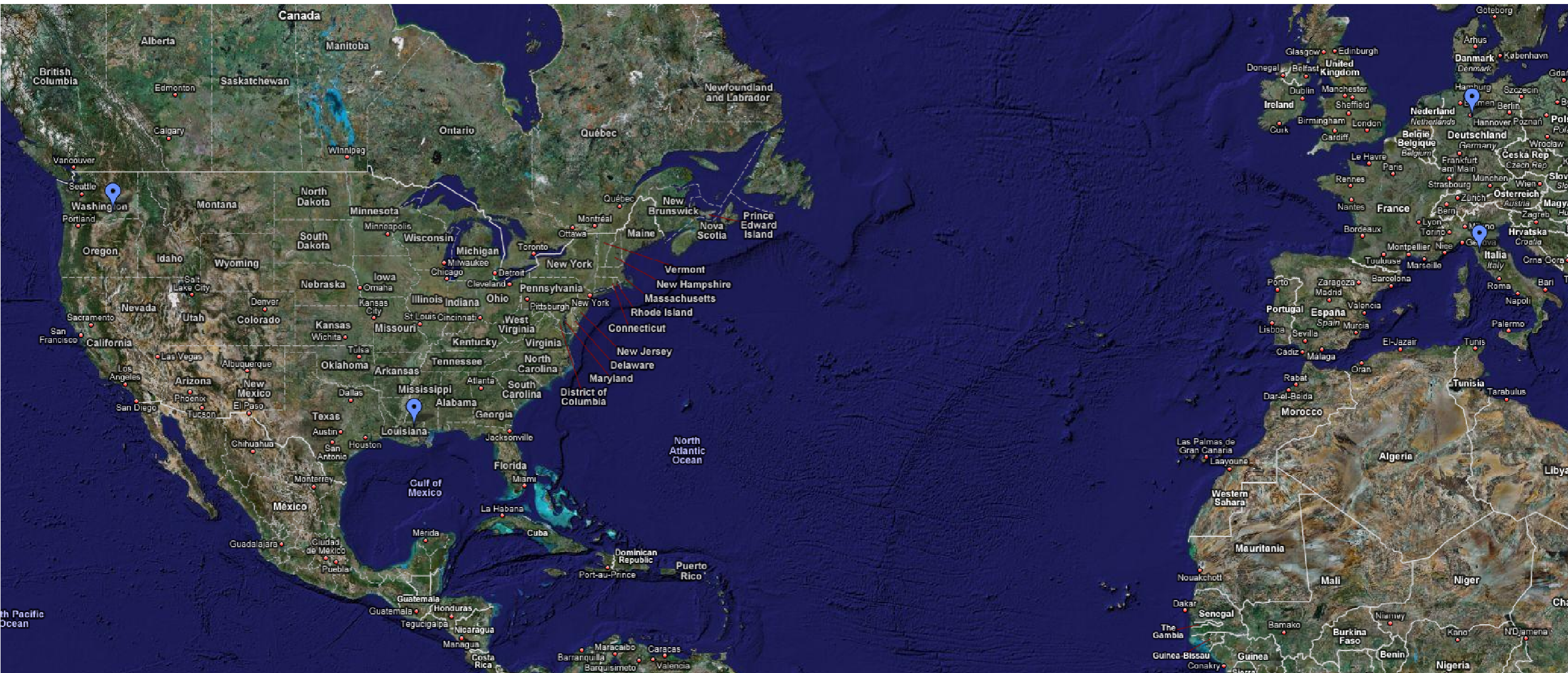
Scott Koranda and Duncan Brown  
LIGO Scientific Collaboration

# Gravitational Wave Astronomy

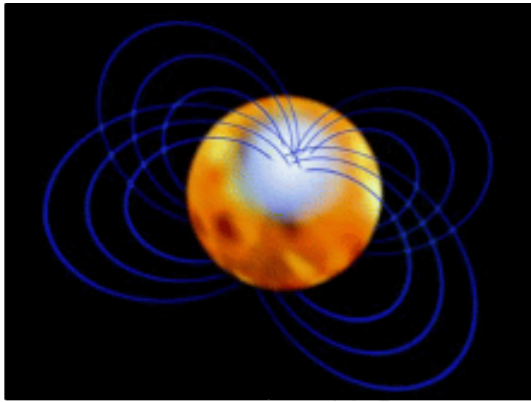
- The Laser Interferometer Gravitational Wave Observatory (LIGO) is an ambitious NSF-funded project to detect gravitational waves and use them to study the Universe



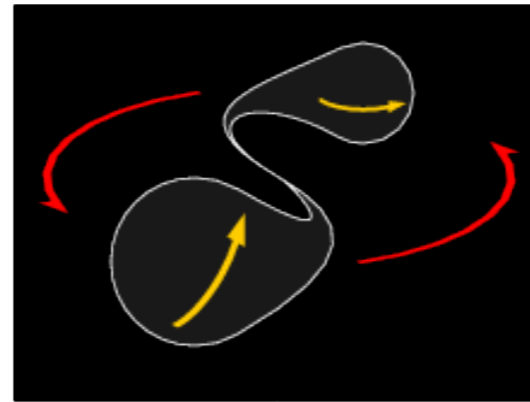
# Location of Observatories



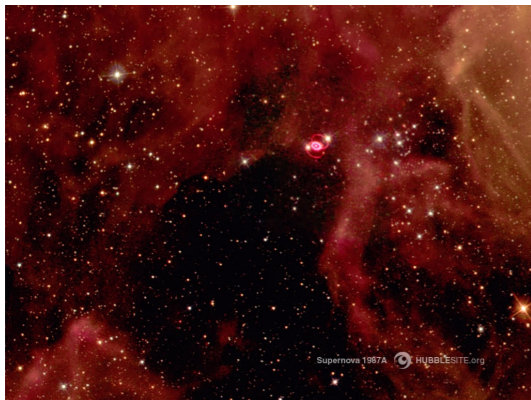
# Sources of Gravitational Waves



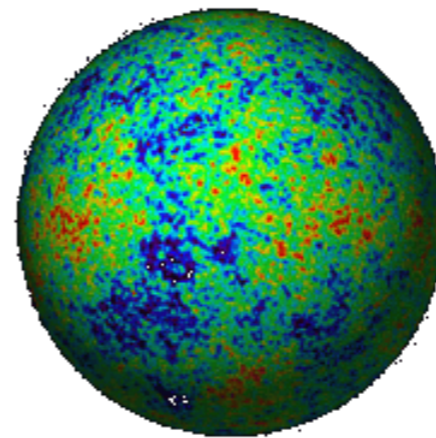
**Continuous Sources:**  
spinning  
neutron  
stars



**Compact binary coalescence (CBC):** inspiral, merger and ringdown of black holes and neutron stars



**Short bursts:**  
supernovae,  
unmodeled  
transient sources



**Stochastic sources:**  
gravitational  
wave background  
from the big bang

# Use of Condor

---

- The LIGO Scientific Collaborations relies heavily on Condor for most of its gravitational-wave searches
- Condor is deployed on over 11,000 cores across 8 compute centers on the LIGO Data Grid
- Condor schedulers manage submission and execution of (almost) all our data analysis pipelines

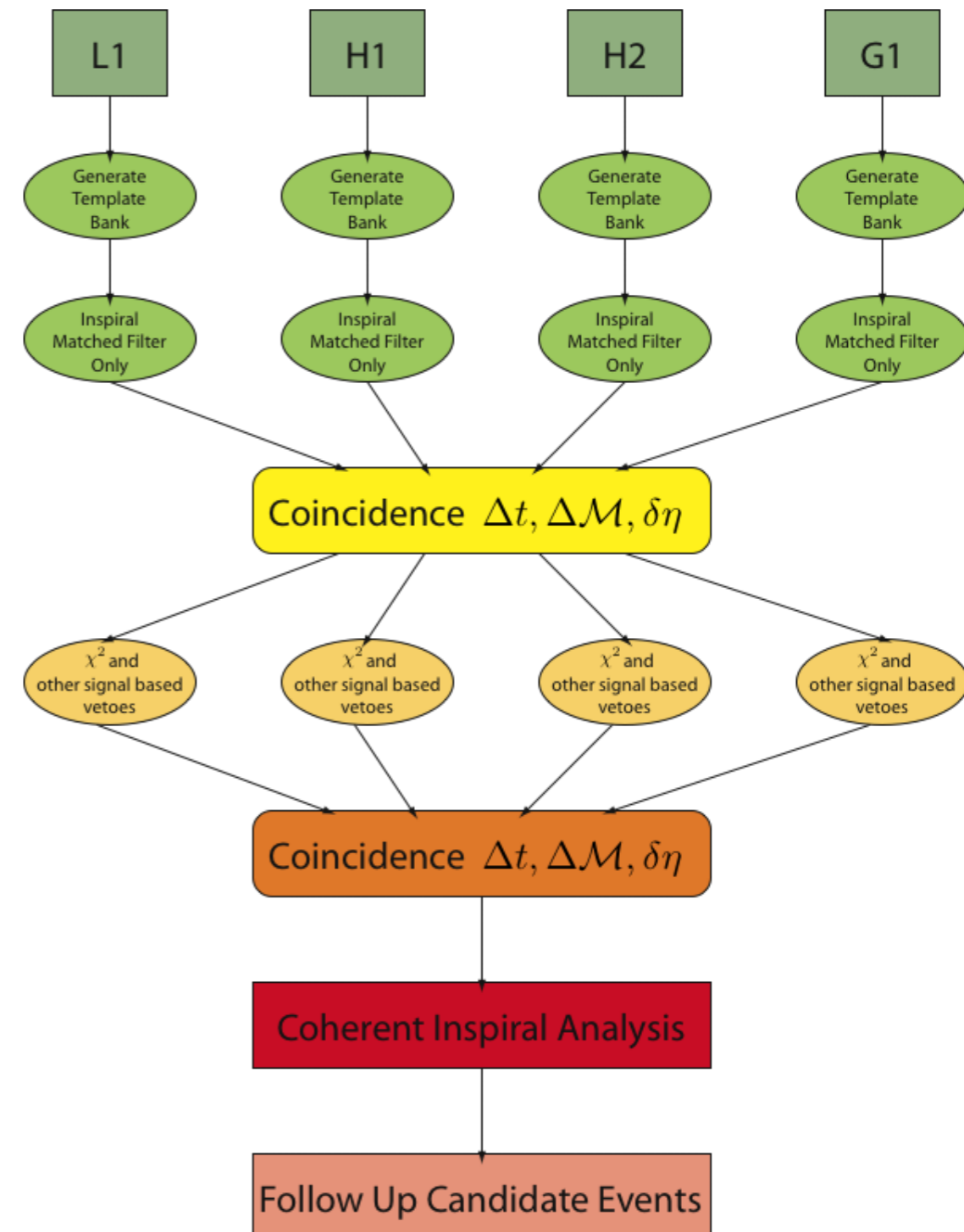
# Management of Workflows

---

- Gravitational wave searches require complex workflows to ensure that data from multiple observatories is analyzed in the correct sequence
- All our data must successfully be analyzed without failures to complete a search
- We are heavily dependent on Condor DAGMan to accomplish this

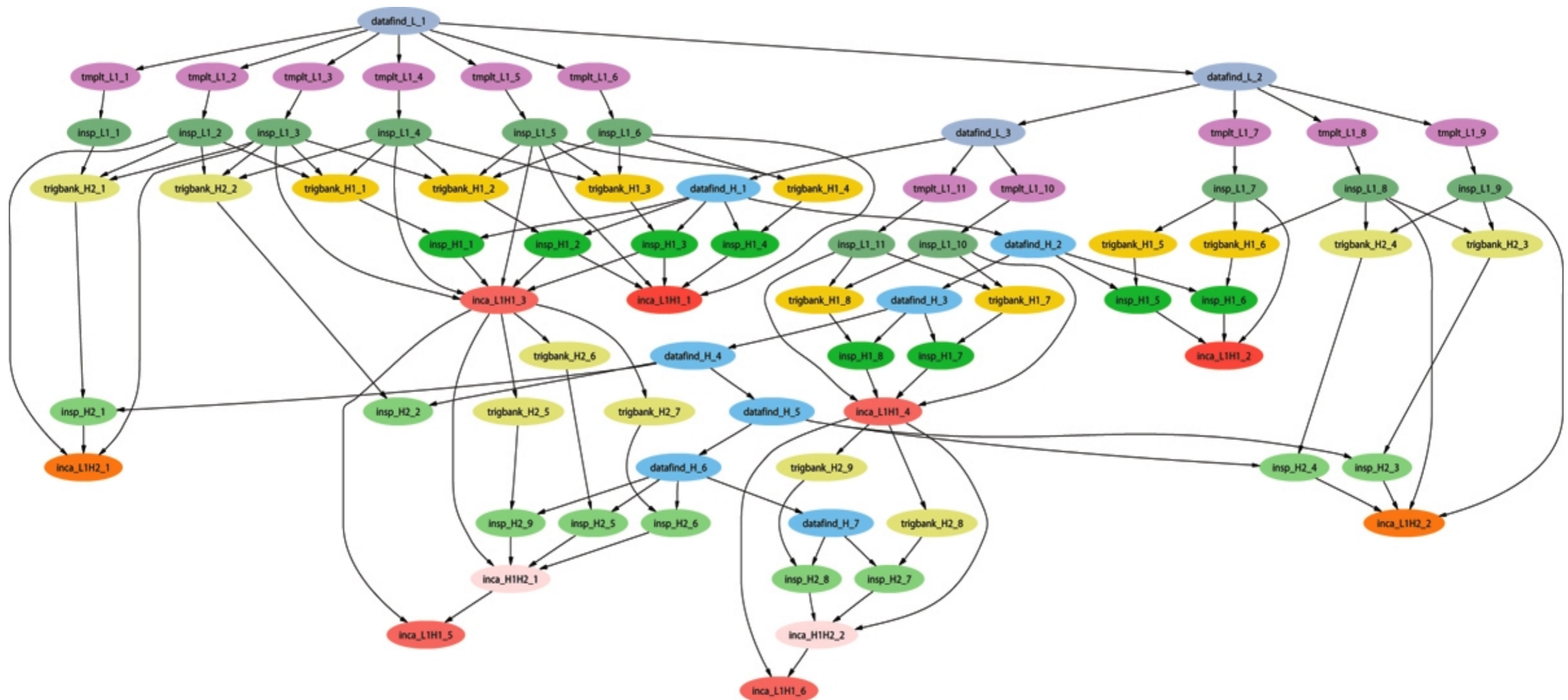
# Example Workflow

- Search for gravitational waves from compact binaries
- Iterate over all data from four detectors
- Coincidence tests compare results of searches in individual detectors to construct list of candidates



# Workflow Construction

- The LSC has developed a toolkit (GLUE) to write out workflows as Condor DAGs





# More Complicated Workflows

---

- We have become more sophisticated in our use of Condor DAGs
- To analyze one month of data for “low-mass” binaries requires 5 analysis DAGs containing ~ 45,000 jobs and 10 plotting DAGs containing ~ 50 jobs
- These DAGs are sub-DAGs of an “uber-dag” that executes the entire workflow
- Runs over entire year of data done with 400K+ nodes in uber-DAG

# DAGMan Enhancements

---

- Enhancements to the robustness of DAGMan's handling of sub-dags have made it much easier to handle our workflows (thanks!)
- Better generation of automatic rescue DAGs
- $O(100x)$  faster startup and rescue recovery for large DAGs
- We are working towards taking advantage of new DAGMan technologies like SPLICE to make management of our workflows even easier

# Checkpointing

---

- Some LIGO codes are C and run happily in the standard universe
- However... many other codes are Python, ROOT, Matlab, etc. which we cannot recompile with checkpointing
- We are eager to test out new checkpointing methods for codes that cannot be re-compiled

# Grid Computing

---

- A big goal for us is to have our jobs “flow” around the LIGO Data Grid to available resources
- We have been experimenting with Condor-C to hook together our Condor pools
- We have had success with the “plumbing” now we need to figure out how to get jobs to flow around the grid. (PegaDAGMan?)

# Other Improvements

---

- We have noticed great improvements in our pool performance in the 7.x series schedd
- But we'll continue to push for more improvements...
- Thanks to the Condor team for being so responsive (and patient) to our requests!