CMB-S4 Transient Pipeline Development and Evaluation

Felipe Menanteau *(he/him)*
CMB-S4 Collaboration
Apr 3-6, 2023

ILLINOIS NCSA

CMB-S4

Objective

- Detect and report transients events for CMB-S4
- Evaluate the SPT-3G software and SPT-3G Transient pipeline for this (https://github.com/SouthPoleTelescope/spt3g_software)
- Use CMB-S4 DC0 Simulations for this task
- Follow the path of Whitehorn (2016) SPTPol and Guns et al (2021) SPT-3G for transient detection

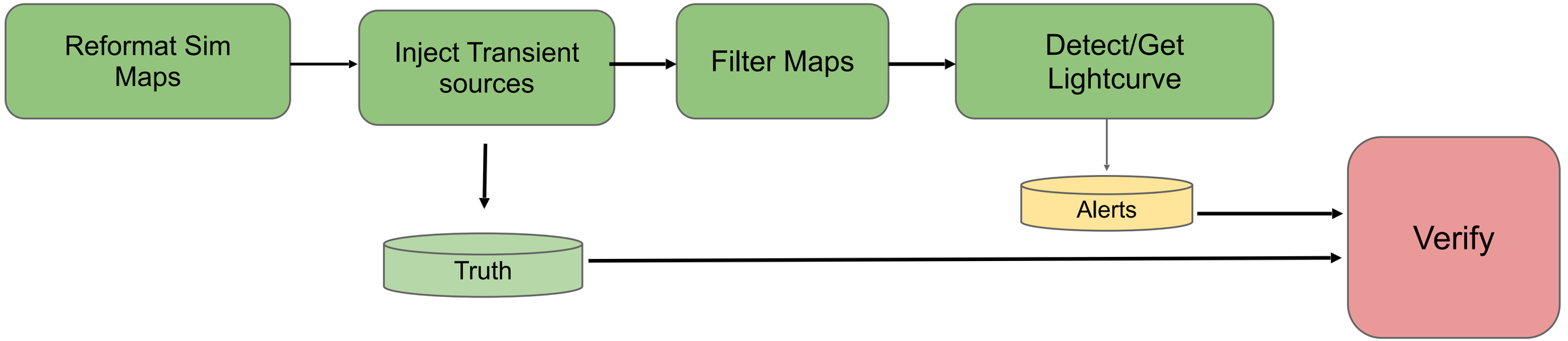
Thanks:

- Sam Guns
- Allen Foster
- Tom Crawford
- Sasha Rahlin
- Nathan Whitehorn

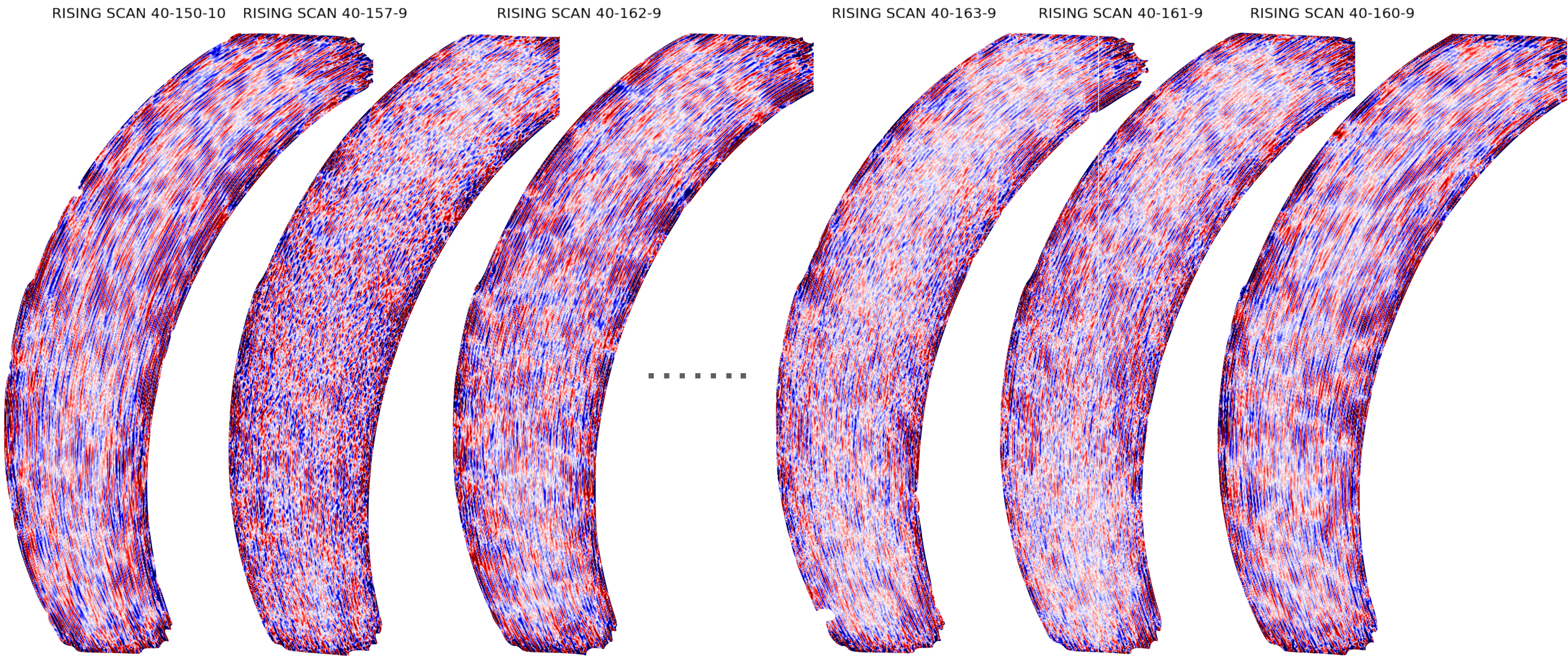
DC0 Simulations

- 5711 CHLAT SCAN maps (RISING & SETTING) transferred to from NERSC to the Illinois Campus Cluster (ICC) at the National Center for Supercomputing Applications (NCSA).
- All 5711CHLAT noise maps in `HDF5` format were transformed into `Healpix` (FITS) all-sky maps using toast (i.e. `toast_healpix_coadd`) on the ICC.
- We tiled the sky in 36 ~20deg x 20deg tiles and use these tiles for our analysis.
- Tiles use a ZEA projection
- We used the Singularity Containers (based of docker) for the batch jobs at the ICC
- Code repo for CMB-S4 Transient Pipeline: https://github.com/CMB-S4/s4trans/

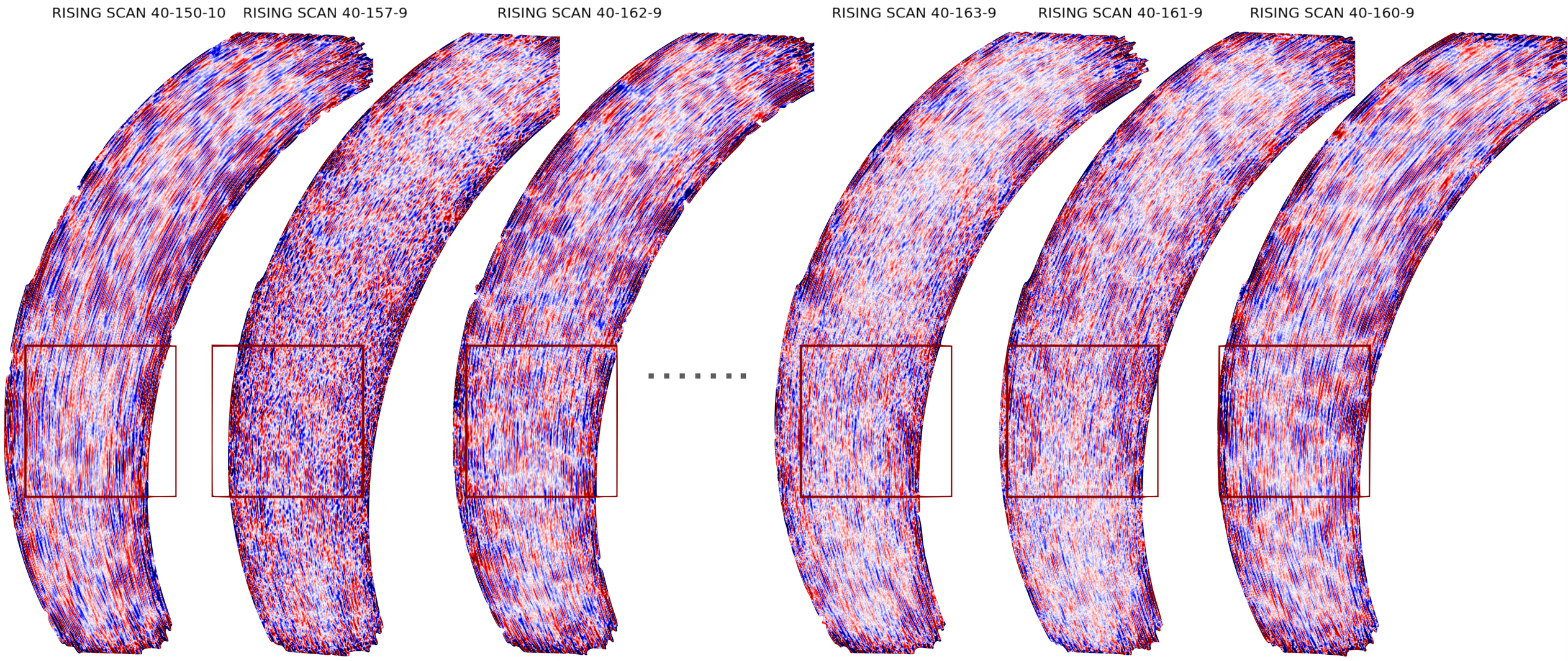
Processing workflow



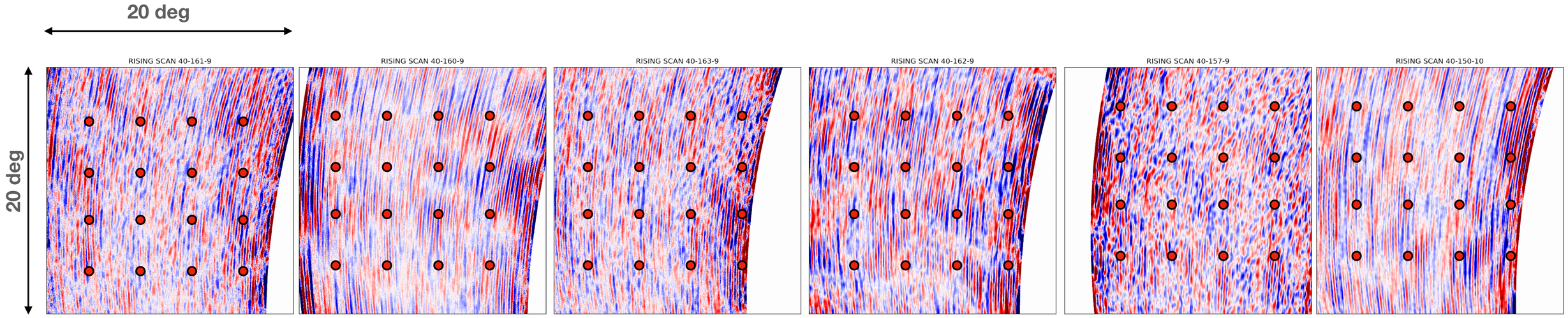
DC0 Simulations



DC0 Simulations (Tiling)

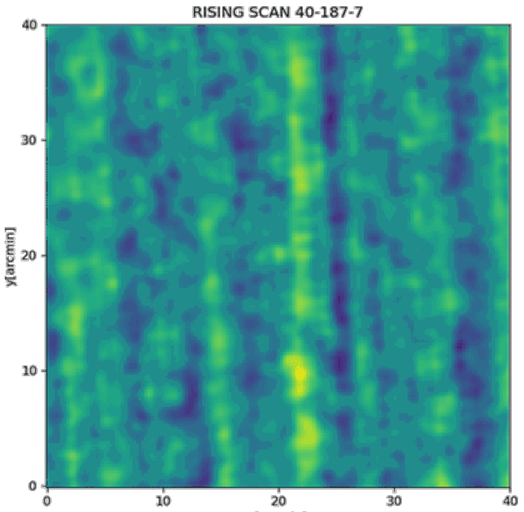


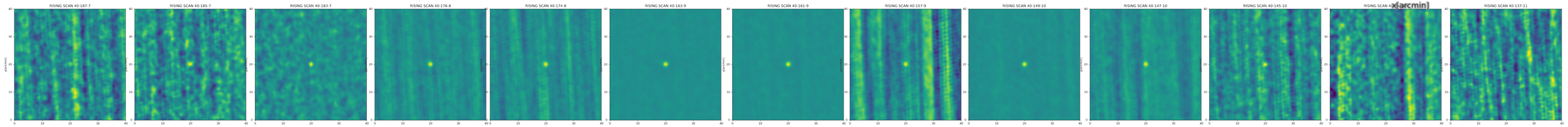
Sky Tiling on DC0 Scans (Arbitrary)

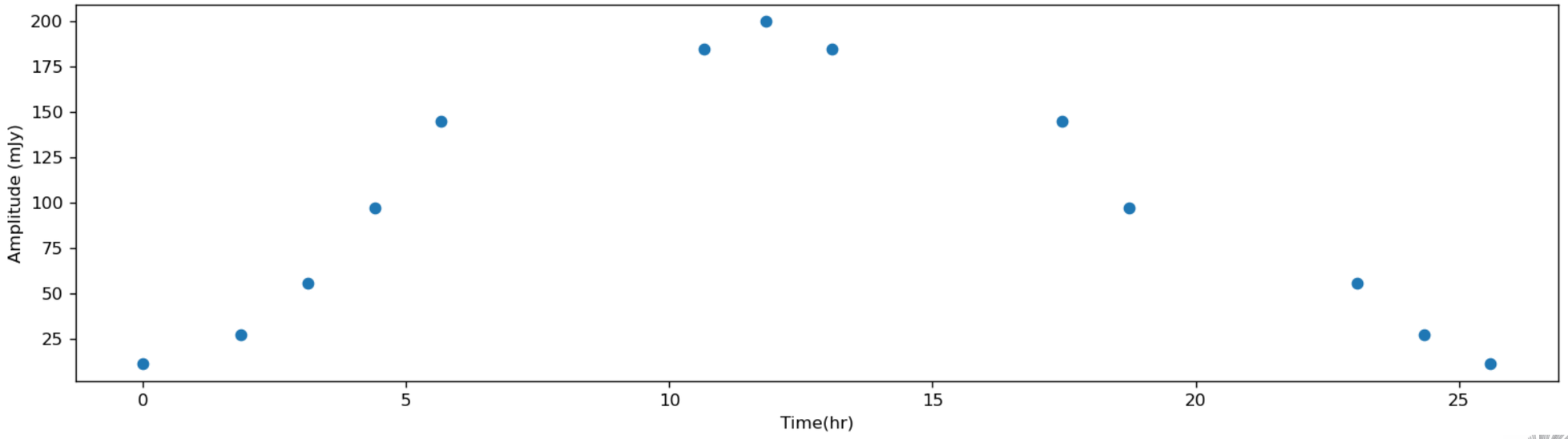


Injection of transient sources (inputs)

- Projection + Injection + Filtering takes ~12 min over 1 tile per scan/observation on ICC running x40 wide

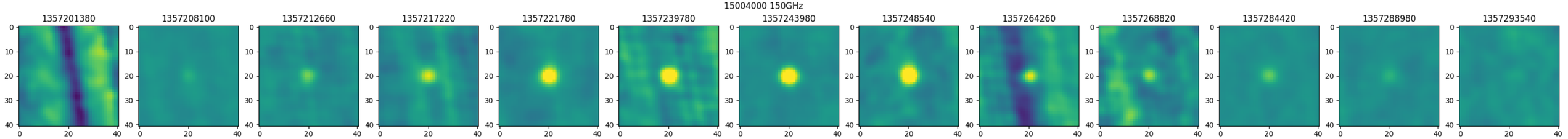


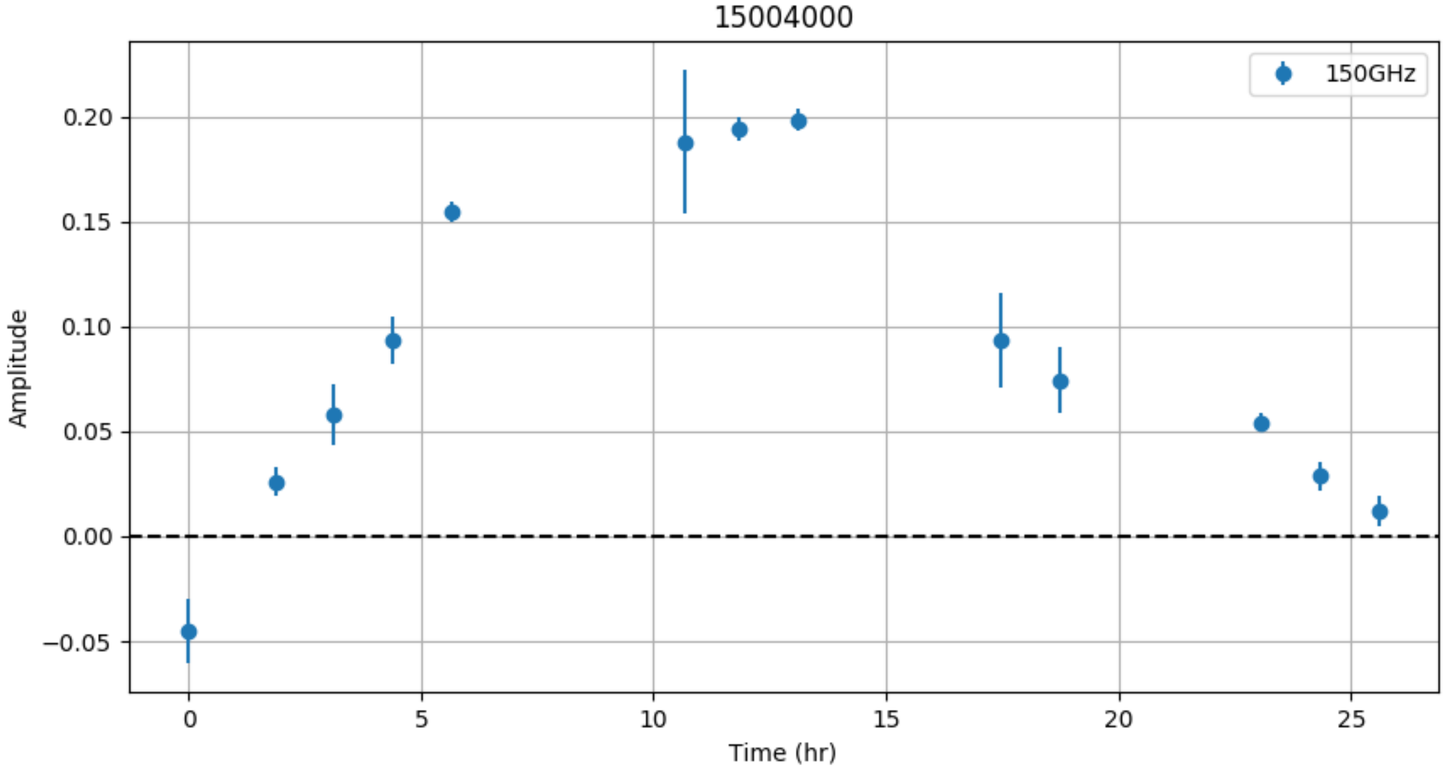




Source Detection and Light Curve Recovery

Using the SPT-3G Transient Pipeline

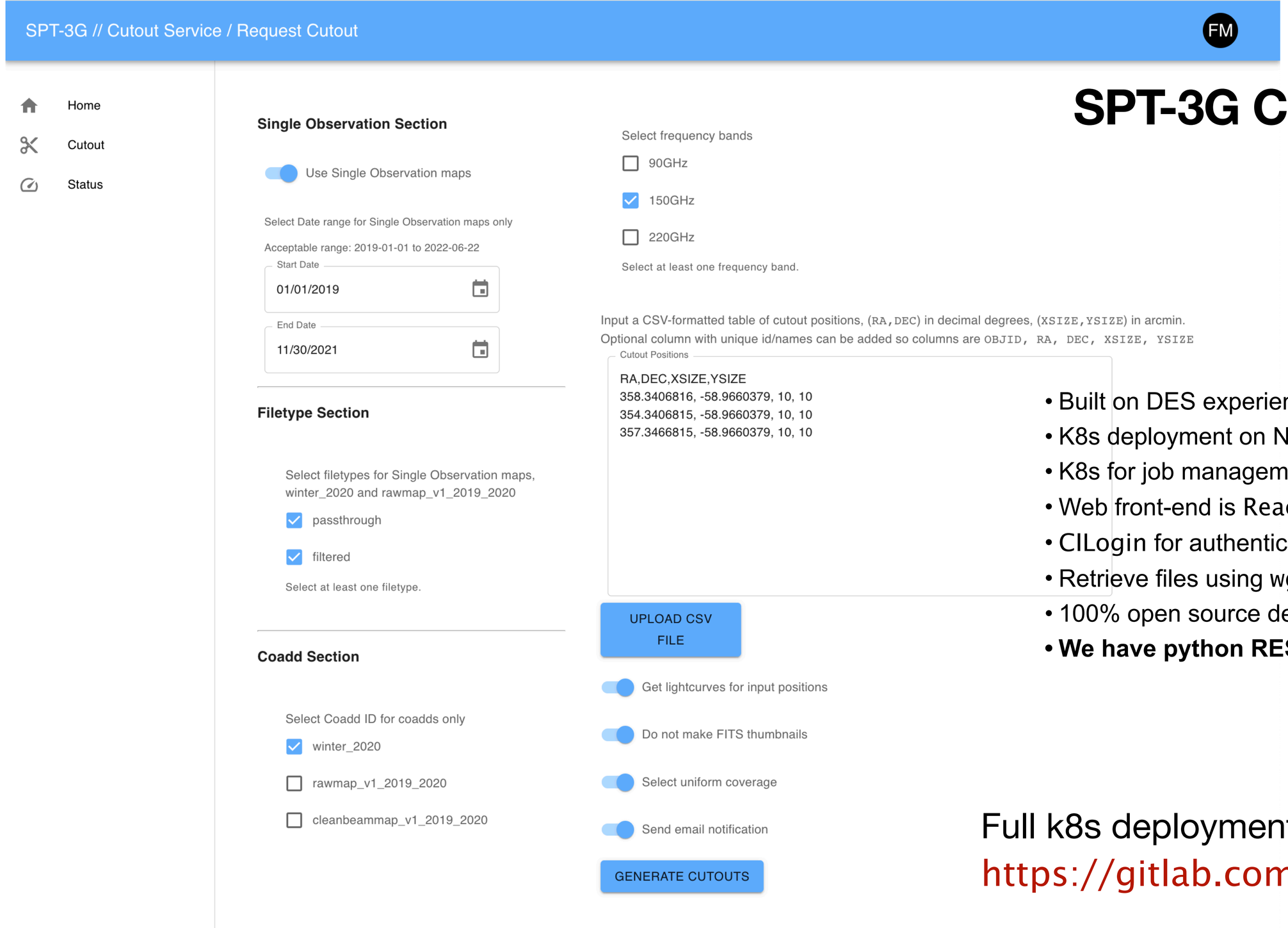




CMB-S4

Future Work

- Run Source injection and LightCurve extraction at scale (all sky?) at ICC
- Inject sources at various time-scales and fluxes
- Verify/Distribute/Asses DC0 results — How??
- Follow steps of SPT-3G collaboration and develop a cutout/lightcurve server for transients like the “SPT-3G Cutout Web App”



SPT-3G Cutout Web App

- Built on DES experience
- K8s deployment on NCSA radiant (open stack)
- K8s for job management an resources
- Web front-end is React JS
- CILogin for authentication
- Retrieve files using `wget`
- 100% open source deployment
- **We have python REST API**

Full k8s deployment repo home:
https://gitlab.com/spt3g/kubernetes

South Pole Telescope (SPT) at NCSA

Launch app Learn more

Conclusions

- We have demonstrated that we can use the SPT-3G software to load, project and filter DC0 scan maps
- Starting point for SPT-3G software are `Healpix` maps.
- We have developed code to insert transient sources of arbitrary flux/times
- We have used SPT-3G software to project (ZEA) and filter the DC0 scans over ~40 observations. Run times are about 15 min/tile.
- Can run over all tiles in 15hrs
- We have demonstrated that we can successfully run the SPT-3G transient pipeline to recover the injected sources.