GAMIT/GLOBK In A Day*

M. Floyd and T. Herring

Massachusetts Institute of Technology

GAGE/SAGE Science Workshop
Pre-Workshop Course
9–10 August 2021

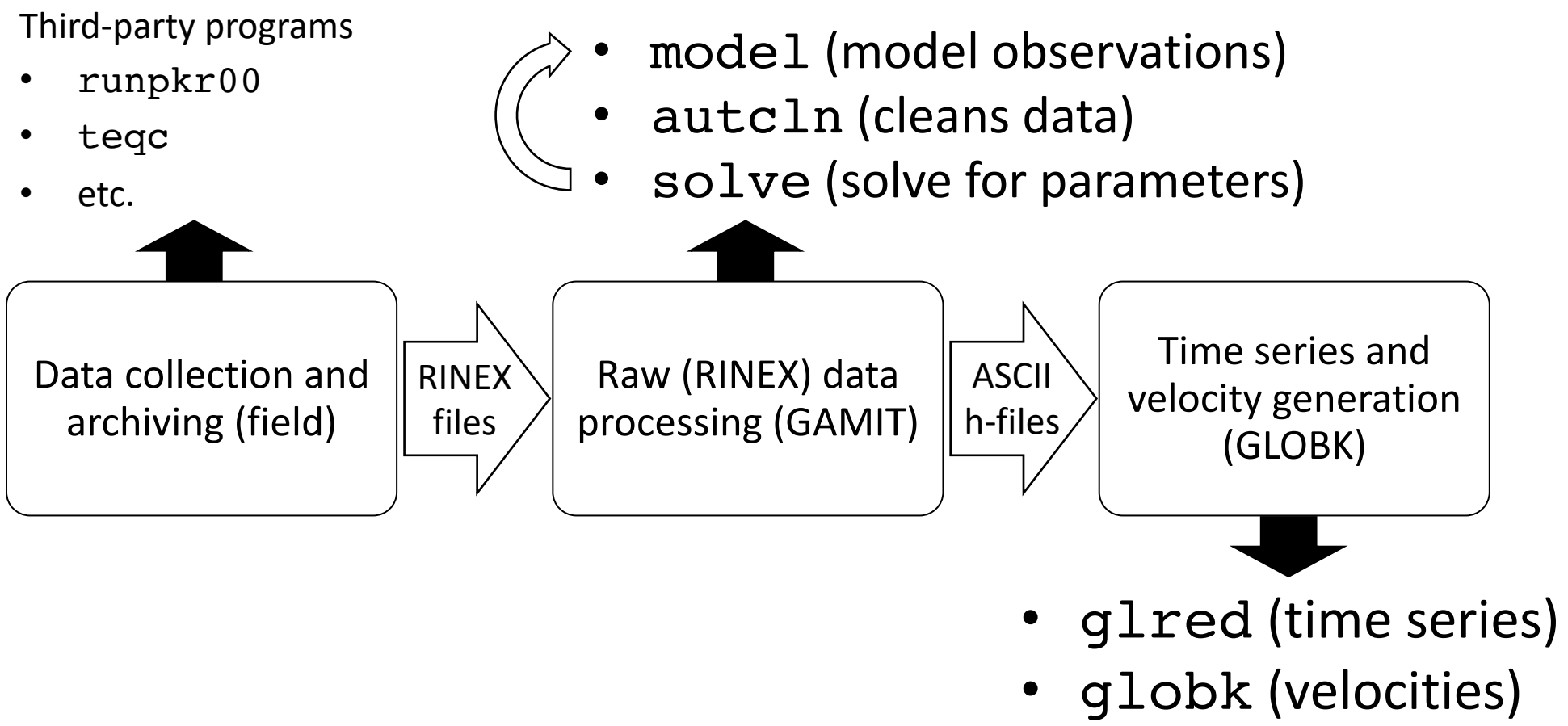
http://geoweb.mit.edu/gg/courses/202108_GAGE-SAGE/

Material from R. W. King, T. A. Herring, M. A. Floyd (MIT) and S. C. McClusky (now at ANU)

*Or two half-days in the world of virtual workshops

Setup and batch processing with `sh_gamit`

Basic stages of GAMIT/GLOBK for geoscience



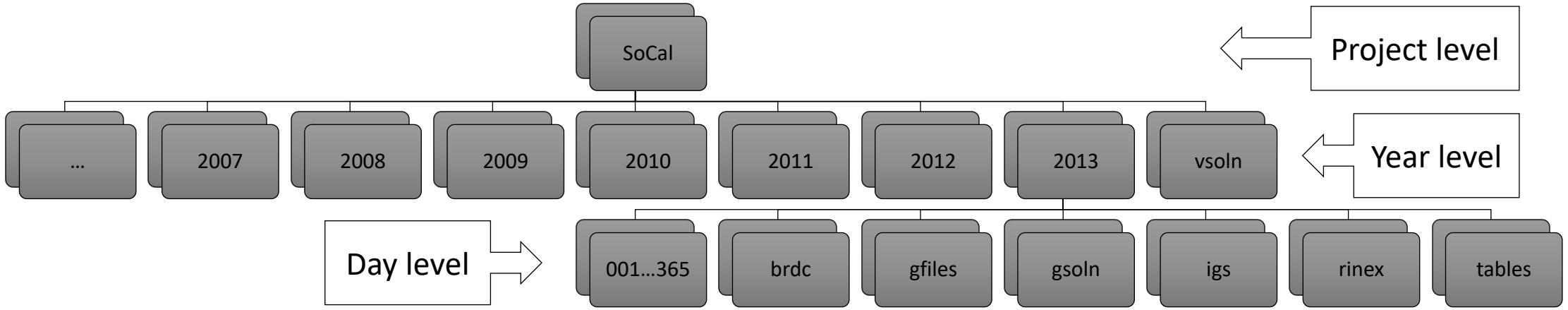
Directory structure

- Top level: global tables and survey directories
- Within each top-level directory:
  brdc/ gfiles/ glbf/ gsoln/ igs/ rinex/ tables/
  001/ 002/ … (these directories are created as needed)
- Generally 50-60 sites is the largest network processed in GAMIT; networks larger than 99 stations require sub-netting of sites (see `netsel`, `global_sel` and `sh_network_sel`)
- Tables are linked from day directories to experiment tables/ and then to ~/gg/tables/
- GAMIT processing occurs in the day (<DDD>) directories
- GLOBK processing occurs in gsoln/

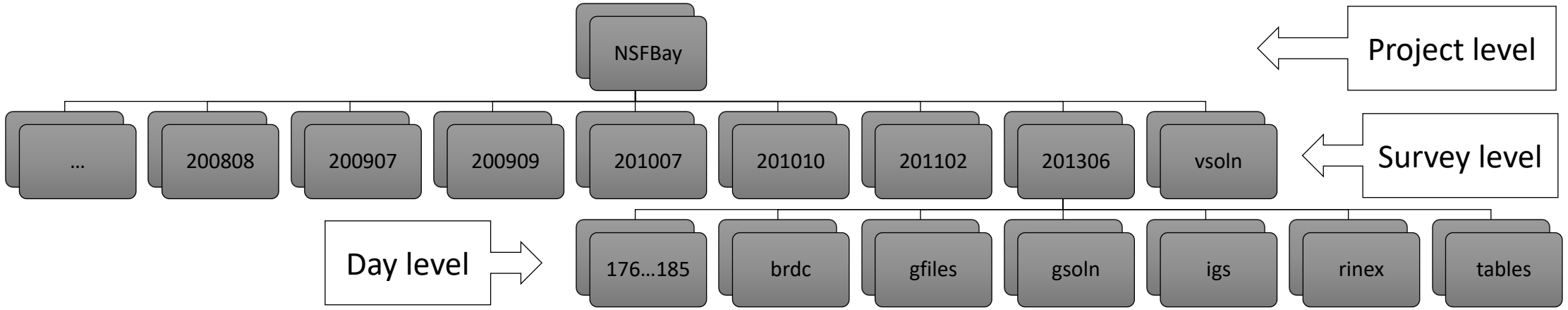
Processing directory

- The processing directory will not have the same structure as the main installation directory
- Choose a different location, do not process in your main installation directory
- We will, however, be copying or linking to the main installation tables (via symbolic link or “shortcut” ~/gg/tables)

Example continuous GPS structure



Example survey GPS structure



GAMIT

1. Run `sh_setup`
   - Check all links, especially to grid files (otl.grid, atl.grid, map.grid, met.grid; see sestbl. for what is “switched on”)
2. Edit sites.defaults to include all publicly available global sites you wish to process and place any other RINEX data to be processed in rinex/ directory
3. Prepare *and verify* station.info, e.g. `sh_upd_stnfo`
   - Covered in previous lecture
4. Prepare *and verify* apr-file, e.g. `sh_rx2apr`
   - Covered in previous lecture
5. Run `sh_gamit`

sh_gamit

- `sh_gamit` is the top-level script for running GAMIT
- The following files are important to verify and/or edit (e.g. after `sh_setup`)
  - autcln.cmd (probably unnecessary to edit)
  - process.defaults (not necessary to edit much, if anything)
  - sestbl. (controls experiment observations and models; defaults OK but may want to edit)
  - sites.defaults (list of sites to process in experiment)
  - sittbl. (controls a priori constraints on sites; probably unnecessary to edit)
  - station.info (*very* important file to get right)
  - .apr-file (*very* important file to get right)

Phase data processing: GAMIT

- Preprocessing
  - Download (sh_get_orbits) and prepare (sh_sp3fit) orbits
  - Make clock files (makej)
  - Download publicly available sites (sh_get_rinex) and convert RINEX files to GAMIT internal format for phase-and-pseudorange observations (makex)
  - Write batch (“b”) files
- Iterative solution (run b-files)
  - Calculate synthetic observations from a priori parameters and models (model)
  - Create observables (LC, L1+L2, etc.), clean data (autcln)
  - Fit calculated to observed by solving for parameter estimates (solve)
  - Update a priori information if large adjustments
- All the above command steps are run for the user by sh_gamit
  - Although preparation of orbits (sh_get_orbits/sh_sp3fit) and RINEX files (sh_get_rinex) often done manually, depending on resources

Suggestions for multi-GNSS processing strategies

- If you wish to combine data from different GNSS, process each system in a separate experiment directory, e.g. /2017G and /2017E for GPS and Galileo
- Download the RINEX files in advance to check for availability of GNSS signals
- For `sh_gamit` use the "`-gnss`" option to specify the GNSS and "`-orbit codm`" ("`-orbit igsf`" OK for GPS)
- Check the orbit fit rms files in the /igs directory to assess the orbit quality
- Combine the resulting h-files in GLOBK to produce a single result (time series or velocities)

Multi-GNSS in GAMIT

- Several scripts now have an additional option ("`-gnss`") that sets the type of GNSS
  - Most likely to use directly: `sh_gamit`, `sh_get_orbits`, `sh_sp3fit`
  - Less likely to use directly: `sh_preproc`, `sh_bcfit`, `sh_rxscan`, `sh_get_times`, `sh_makexp`
- Valid arguments are (only one of)
  - `G` (GPS)
  - `R` (GLONASS)
  - `E` (Galileo)
  - `C` (BeiDou/COMPASS)
  - `J` (QZSS)
  - `I` (IRNSS)
- The default is still "G" (GPS)

Things that will really save you a headache

In our experience, 90% of user issues come down to one of the following:

- Not having the latest version or associated updates installed
- Incorrect or missing records in station.info
- Inaccurate or missing records in the “apr”-file

So, please:

- Update your installation on a regular basis (incremental updates are wrapped into a tar-file on the first day of each month)
- Take the time verify your setup files and do not rely on `sh_gamit` or other automated scripts to get everything correct without some manual intervention

Files you need to worry about

- RINEX files: local plus list in sites.defaults
- Control files
  - process.defaults : minor edits for each survey
  - sestbl. : experiment and models setup (unchanged for most processing)
  - sites.defaults : sites to include or omit and source of metadata
  - sittbl. : sites constrained for ambiguity resolution
  - globk.cmd : `use_site`, `apr_neu`, `apr_svs`, `apr_wob`, `apr_ut1`, `sig_neu` and `mar_neu` commands
  - glorg.cmd : `apr_file`, `pos_org`, `stab_site` commands
- a priori coordinates (.apr-file, l-file)
- Metadata (station.info)
- Differential code biases (dcb.dat); download current values once per month
- Satellite characteristics (svnav.dat); download current with each new launch
  - Now linked to igs_satellite_metadata.snx, which is maintained by the IGS

process.defaults

- Controls:
  - Data and processing directory structure
  - Some session parameters (e.g. start time, length and data interval, and apr-file name)
  - Peripheral book-keeping (e.g. files to compress, archive or delete, and email address for summary)

sites.defaults

- Controls sites to be in included in experiment of given name
  - Whether or not these sites should be downloaded from a public data server (use “ftprnx” flag)
  - Whether or not these sites should have their metadata updated from the RINEX file header (not recommended, particularly for continuous sites; use “xstinfo” flag)
- May use one sites.defaults file with multiple experiment names and use “`-expt`” option in `sh_gamit` to process only certain sites
- May use a different experiment setup for different days
  - `sh_gamit` will look first for “sites.defaults.YYYYDDD”, then a generic “sites.defaults” file

autcln.cmd

- Controls all parts of the phase cleaning algorithm
- Defaults generally work well for all experiments
  - May occasionally wish to change:
    - elevation mask
    - criteria to keep more data from sites with bad a priori co-ordinates

.apr-file

- Controls a priori (input) coordinates of sites
- Convergence of (non-linear) processing is about 1:1000, i.e. 10 m accuracy for a priori co-ordinate will result in final coordinate accurate to about 10 mm
  - Important to have good a priori coordinates
- Utilities include: `sh_rx2apr`
- .The experiment l-file is initialized each day with the coordinates in the .apr-file specified in process.defaults (while retaining any entries added during prior processing for sites not in the .apr_file

station.info

- Controls site occupation metadata
  - Site name
  - Start and stop times of occupation
  - Receiver and antenna information (types, serial numbers, firmware, heights)
- Utilities include `sh_upd_stnfo` which invokes program `mstinf`
- Options for metadata include
  - Pre-prepared station.info (`sh_upd_stnfo`, `make_stnfo`)
    - Must set “xstinfo” in sites.defaults
  - RINEX headers (`sh_gamit` default but may change soon)
    - Update station.info unless an entry already exists for the day being processed or “stinf_unique” is set to “-u” in process.defaults and entry has not changed
    - Can be used with non-standard receiver and antenna names specified in guess_rcvant.dat (ideally your RINEX files have the IGS official receiver and antenna names. It is critical that this information is correct
- THIS IS A VERY IMPORTANT FILE!
  - If you do not get this file correct (and verified) before processing, you may lose a lot of time reprocessing phase data at the GAMIT (slowest) stage

sestbl. (“session table”)

- Controls processing setup
  - Observables to use (e.g. LC, L1+L2, etc.)
  - Experiment (orbits and EOPs) type
    - “BASELINE” solves for site coordinates only using fixed orbital parameters [default]
    - “ORBIT” solves for orbital parameters only using fixed site coordinates (from .apr-file)
    - “RELAX.” solves for both site and orbital parameters
  - Models used

sittbl. (“sites table”)

- Controls:
  - Site-specific information for processing
    - Constraint (accuracy) of a priori coordinates in .apr-file

Approximate position and stabilizing network

Accurate a priori coordinates necessary for good GNSS processing

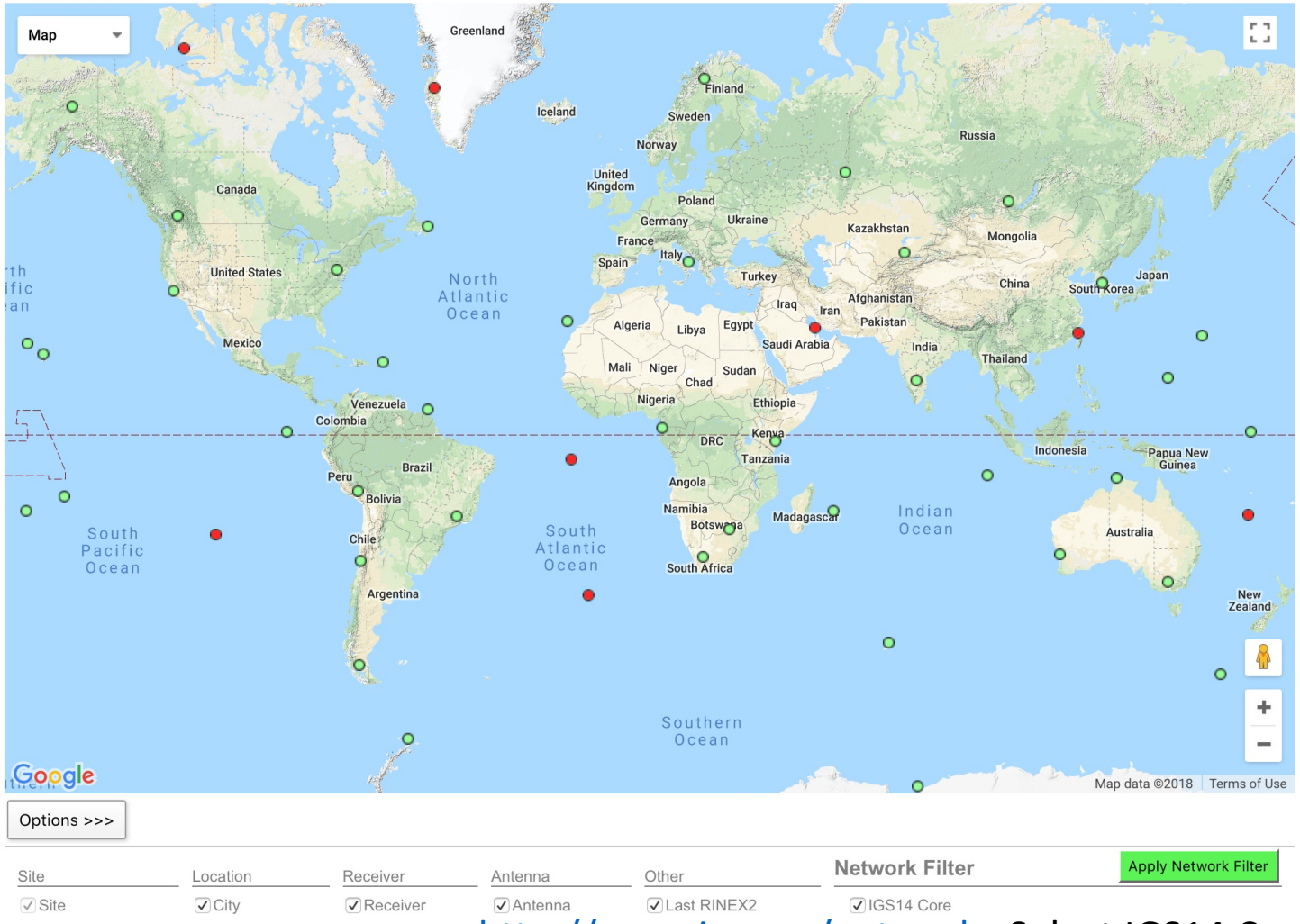
1. Run `teqc` to create RINEX observation and (broadcast) navigation files, e.g.
   ```
   teqc +nav abcd3650.14n +obs abcd3650.14o –tr d 12343650.dat
   ```
2. Run teqc in qc-mode on observation file with navigation file to get pseudorange-derived estimate of approximate coordinate, e.g.
   ```
   teqc +qc –nav abcd3650.14n abcd3650.14o
   ```

May also be done using GAMIT/GLOBK's `sh_rx2apr`

Ultimately we will use the position of known sites to define a reference frame, so we must include those sites in our network to be processed

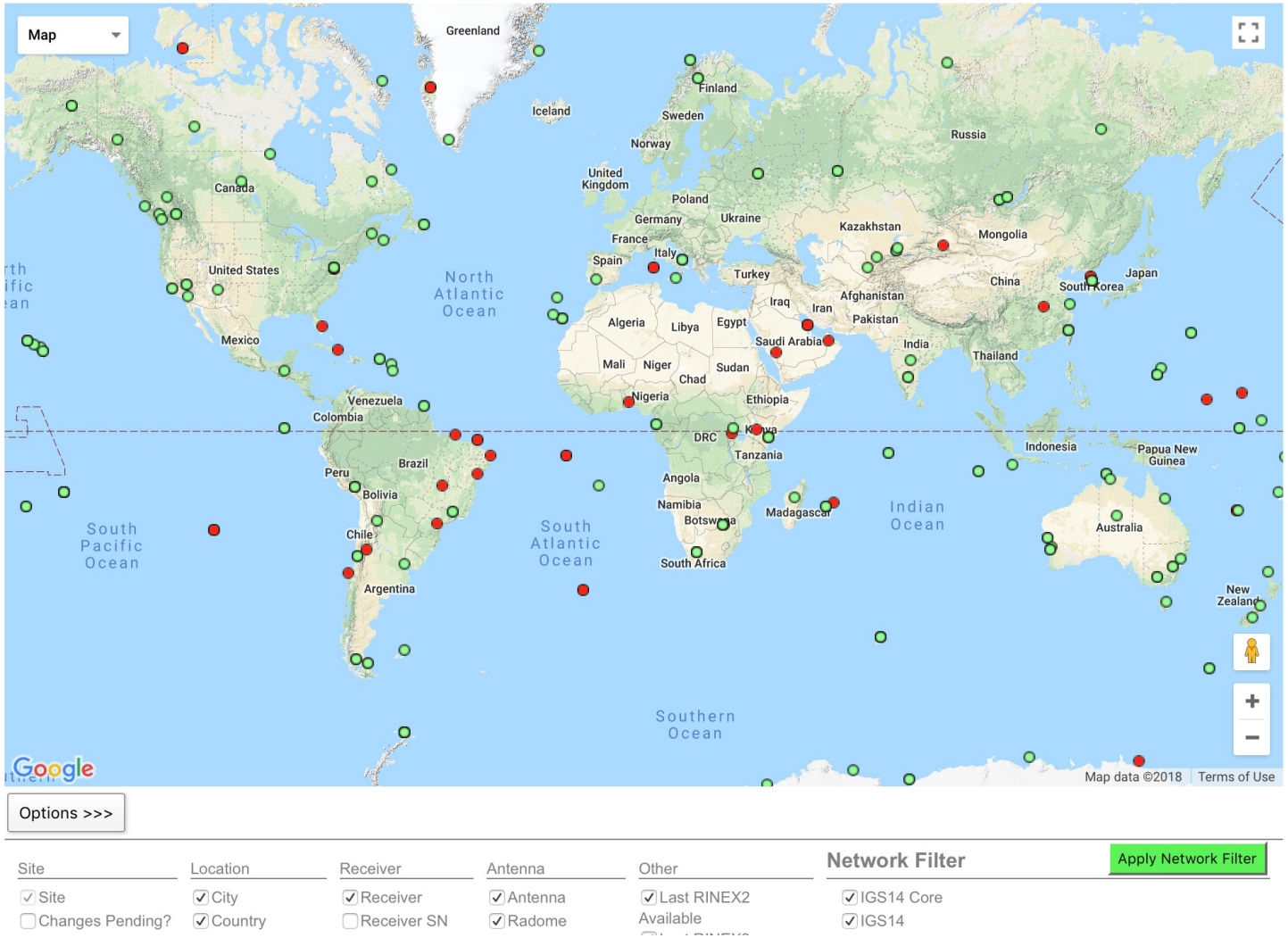
IGS (IGS14) reference frame core network

- DRAO
- GOLD
- GUAT
  KOKB
- MDO1
- MKEA
- NLIB
- PETP
- SCH2
- SCUB
- STJO
- USNO
- WHIT



http://www.igs.org/network Select IGS14 Core

IGS (IGS14) reference frame network



Reference frame implementation

- The main point of reference frame definition is to use sites with well known coordinates to stabilize your processed network
  - If your processed network already contains sites with known accurate coordinates, e.g. in ~/gg/tables/igb14_comb.apr, then there is no need to add additional sites to your processing
  - Your “stab_site” list in `glorg` command file should be guided by and closely associated with the contents of an accurate .apr-file, e.g. igb14_comb.apr by default but could be any accurate .apr-file you choose include in `globk` and `glorg` command files
    - ~/gg/tables/igb14_hierarchy.stab_site relates to ~/gg/tables/igb14_comb.apr
- Decide early in your processing whether or not you will need to include additional, e.g. IGS, sites in your GAMIT processing to use ultimately to realize your preferred reference frame
  - The first decision you must make (choosing sites to process) is greatly influenced by the ultimate processing step (reference frame realization)!

`sh_gamit` internal operation

- The following programs are run by the script:
- `makexp` and `makex` prepare the data
- `fixdrv` prepares the batch control files
- `arc` integrates GNSS satellite orbits
- `model` calculates theoretical (modeled) phase and partial derivatives of phase with respect to parameters
- `autcln` repairs cycle slips, removes phase outliers, and resolves the wide-lane ambiguities
- `solve` estimates parameters via least-squares, resolving the narrow-lane ambiguities and creating an h-file for GLOBK (user constraints are removed in the h-file to allow reference frame definition)

Steps in the standard GAMIT batch sequence

- `arc`, `model`, `autcln`, `solve` for initial solution
  - 5-minute sampling, no ambiguity resolution (GCR only)
  - update lfile. for coordinates adjusted > 30 cm
  - look at: autcln.prefit.sum; q<expt>p.ddd
- `model`, `autcln`, `solve` for final solution
  - 2-minute sampling, ambiguity resolution
  - Look at: autcln.post.sum, q<expt>a.ddd
- Final solution repeated if NRMS reduced by > 30% from initial solution, to assure good editing and linear adjustment of parameters (original final solution files overwritten)

What `solve` produces

- Print output is the q-file, which records

  in detail

  - A constrained solution without ambiguities resolved (“GCR”)
  - A constrained solution with ambiguities resolved (“GCX”)
  - These are the solutions you should examine, along with the `autcln` summary files, to assess the quality of the solution

  and in summary only

  - A loose solution without ambiguities resolved (“GLR”)
  - A loose solution with ambiguities resolved (“GLX”)
- Updated l-file for successive iterations or days
- Useful output for GLOBK is the h-file (analogous to the IGS standard SINEX file), which contains the parameters estimates and full covariance matrix.
- (There is also an o-file, which is just the q-file but in more machine-readable form, and is seldom used; and, if orbits adjusted, an updated g-file)

What can go wrong?

- Site missing (not listed)
  - No RINEX data within session span: check RINEX file and/or makex.expt.infor
  - Too few data, x-file too small and not used: check RINEX file size, change “minxf” in process.defaults
- Site in solution but no data or adjustment
  - a priori coordinates > 10 m off: check range rms in autcln.prefit.sum,
    - run `sh_rx2apr` differentially for several RINEX files
  - bad receiver: examine RINEX files or initial c-files with `cview`
- q-file nrms > 0.2
  - solution over-constrained: check GCX vs GLX nrms, rerun with only one site constrained in the sittbl.