Software Discussion

- Onboard instrument software
  - Tied to hardware design
    - Onboard OS, microcontrollers, FPGAs, some combination?
  - Trigger algorithms
    - GBM uses trailing background, then binning on various timescales and energy scales
    - If large telemetry volume, send everything down
    - Triggering allows you to select intervals
    - Limitations - onboard processing
    - Add some features of GBM Untargeted search
      - Background modeling
      - Additional time/energy ranges
      - Increased phase steps
  - Commanded download of intervals of interest
  - Trigger classification
    - False trigger rejection
    - Suppress based on classification
    - Bright transients
  - Onboard localization
    - How useful in a nanosat? - It compresses data to only send that info.
      - GBM uses a simple lookup table of rates and sky position in spacecraft coordinates - single table not very relevant if orientation changing
  - Languages
    - GBM software written in C
    - Current state of the art is C++
  - Downlinked file formats
    - Portable file formats with standardized readers
- Ground Pipeline
  - Ground localization
    - Relative Rates
      - GBM uses simplified version of detector response for trial localization, chi squared fitting using database of atmospheric scattering (5 angles - GRB position with respect to spacecraft and earth), spectral template
      - Bayesian approach using a combination of directional and spectral information
      - To combine multiple nanosats - try more spectral templates?
    - Timing approach
      - If spectral response is different enough, light curves will be different

- - Model pulse on board and send parameters right away rather than photon data
  - ○ Standard file formats - types and contents
    - ■ Will be easier to combine data if it's generated the same way
  - ○ Scientific data analysis - temporal/spectral/localizations
    - ■ Common databases - atmospheric scattering
    - ■ Calibration database shared or separate
  - ○ Every team run their own pipeline, or run joint pipeline, or hybrid approach?
- ● Databases
  - ○ Have ability to reprocess data as we learn and have new versions of software?
  - ○ Documention

- ● Summary
  - ○ Write up full description of GBM trigger algorithm, maybe share some of code
  - ○ Write white paper describing trigger algorithms, source classifications and limitations, as recommendations to each mission
  - ○ Need to agree on standard file format
  - ○ Need to determine where division is between individual mission pipelines, and a common pipeline
  - ○ Open source platform with core development team
  - ○ As we develop new versions of software, do we have central database to be able to reprocess?