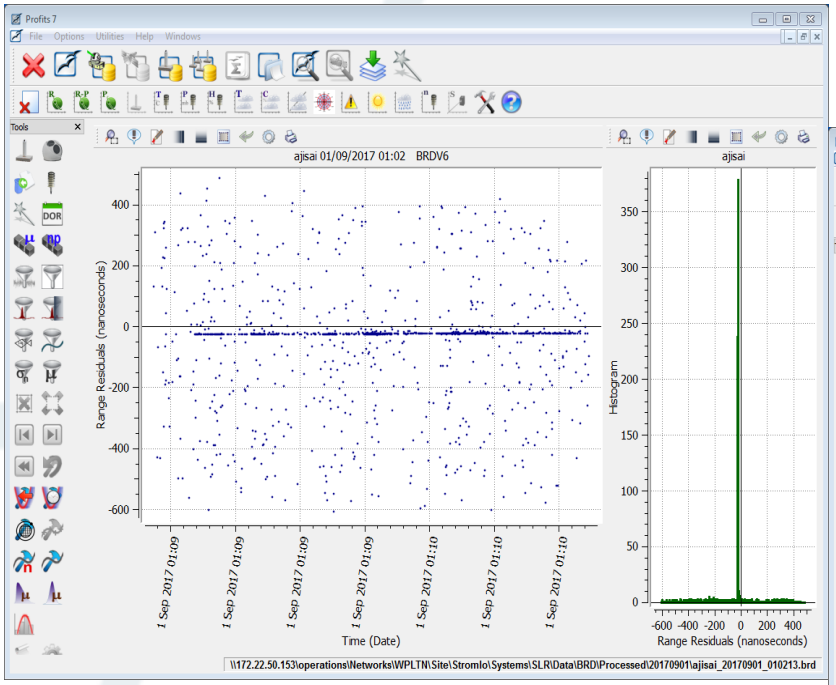
Autonomous Post Processing

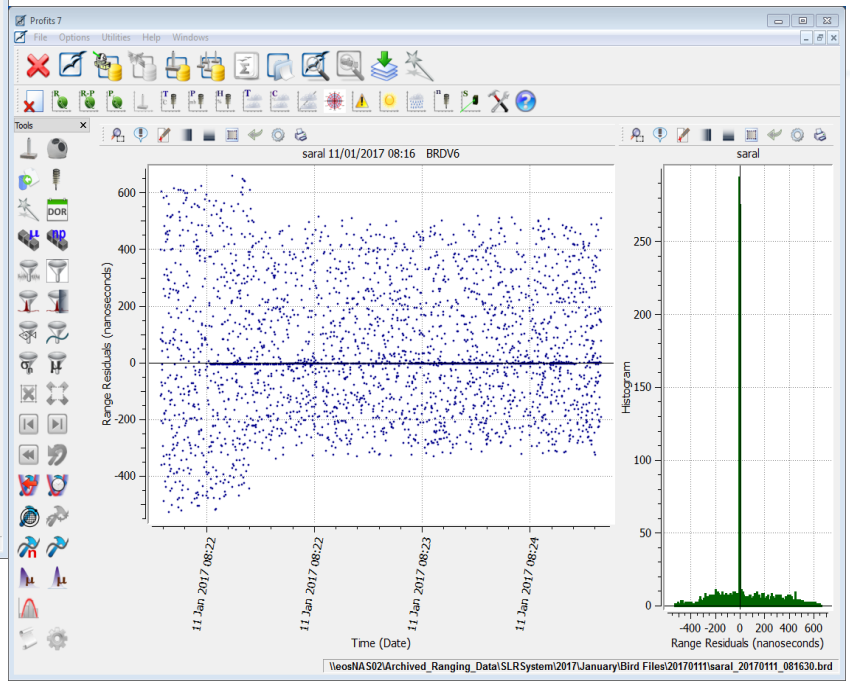
- Characteristics of Mt Stromlo System SLR ranging data;
  - Timing event epochs (using CSPAD).
  - Data collected in BRD files and processed when pre- and post-cals are available
  - Cal data is also collected in BRD files and processed as another target.

- What are the challenges?
  - Managing and applying calibration data
  - Identifying returns from satellites vs noise

Autonomous Post Processing

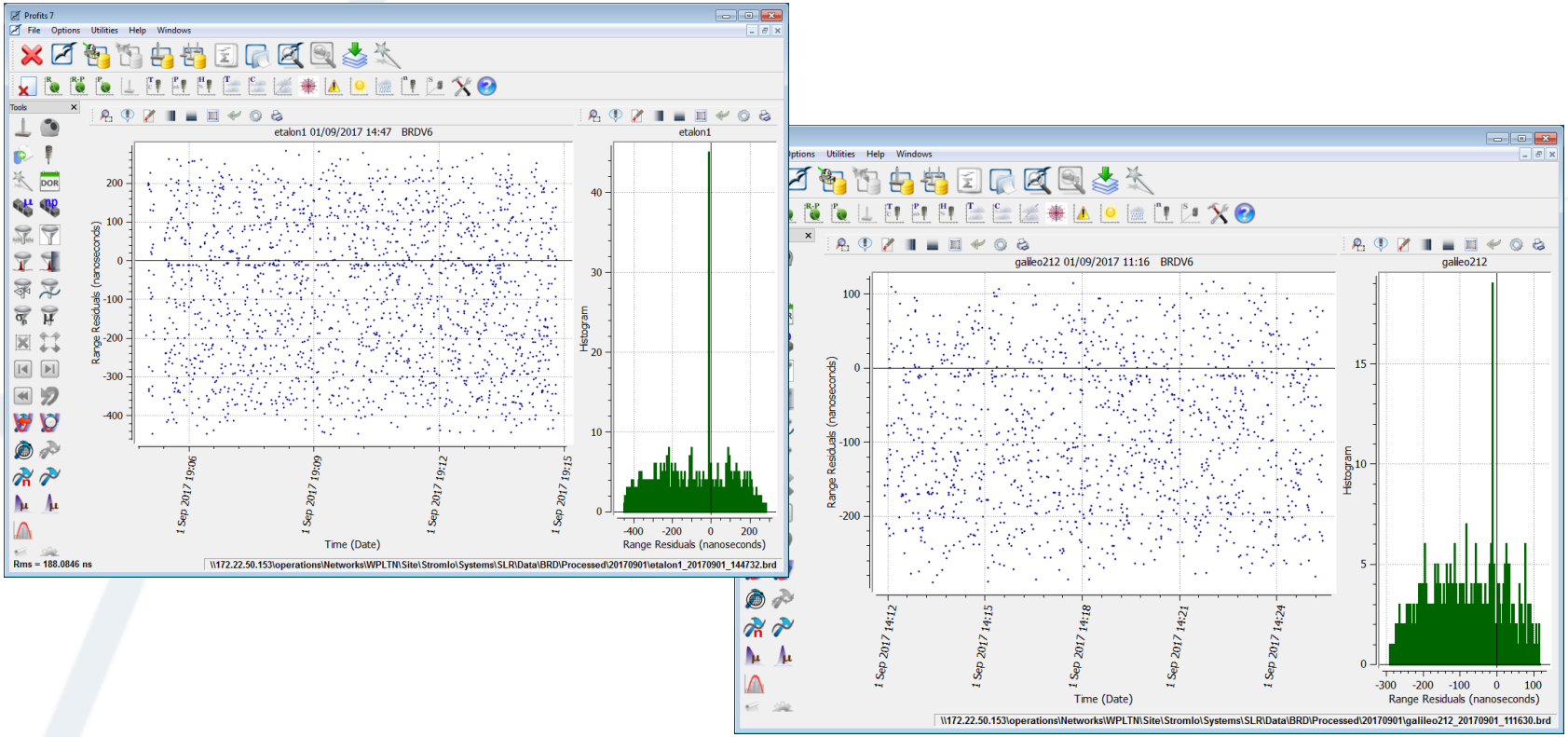
Examples of reasonably strong, flat signals in moderate noise. Easy to identify and extract the required signal.





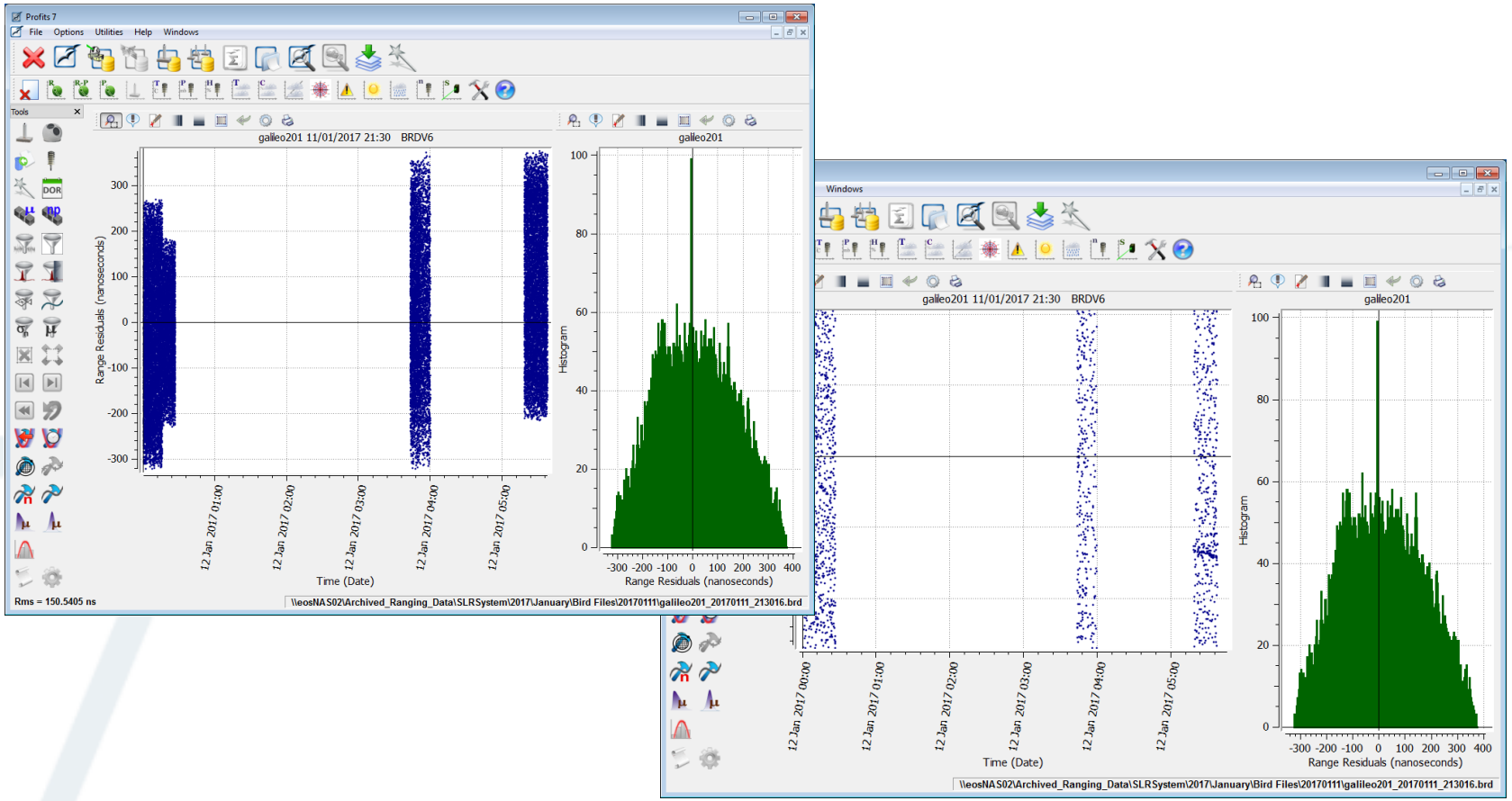
Autonomous Post Processing

Examples of weak, flat signals in moderate noise. Still reasonably easy to identify and extract the required signal.



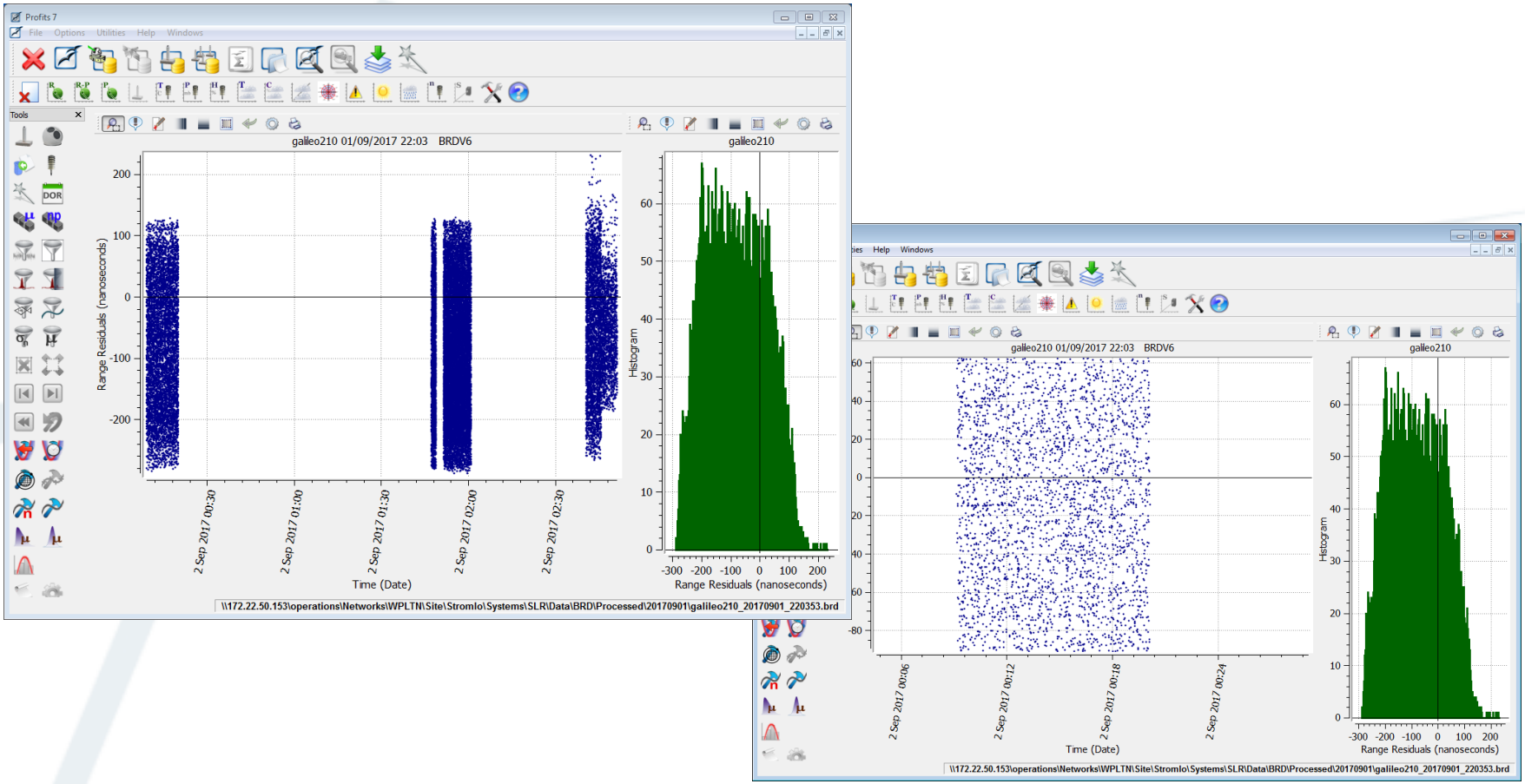
Autonomous Post Processing

Example of a reasonable strong signal in strong noise. Still able to identify and extract the required signal.



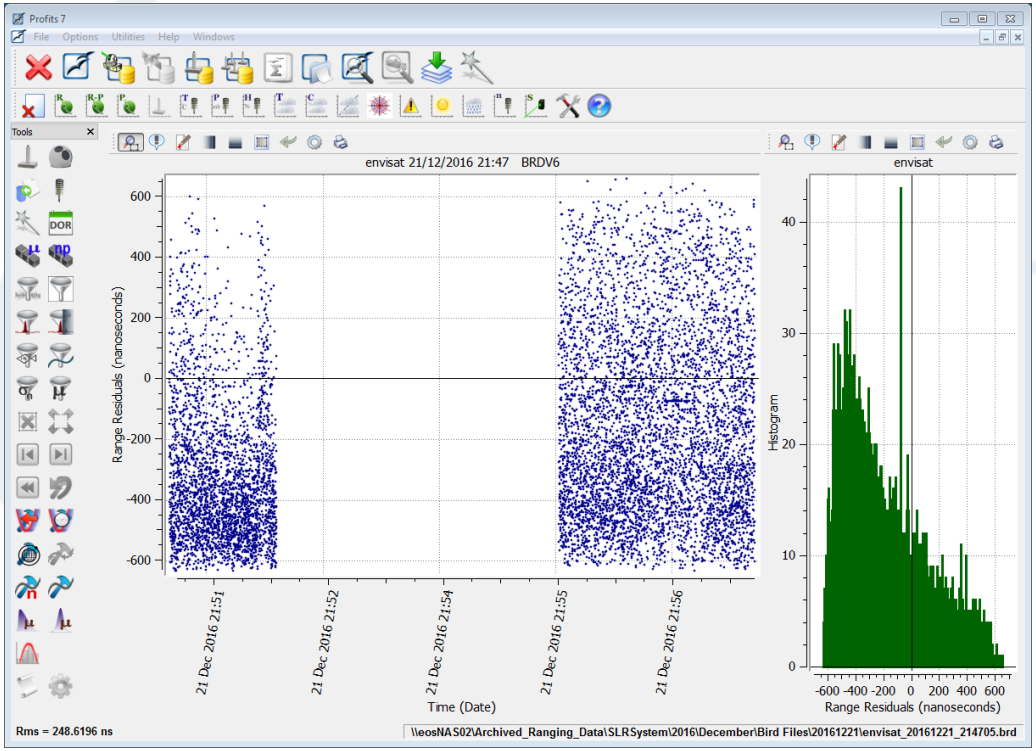
Autonomous Post Processing

Example of a weak signal in strong noise. Identification and extraction of the required signal is very difficult.



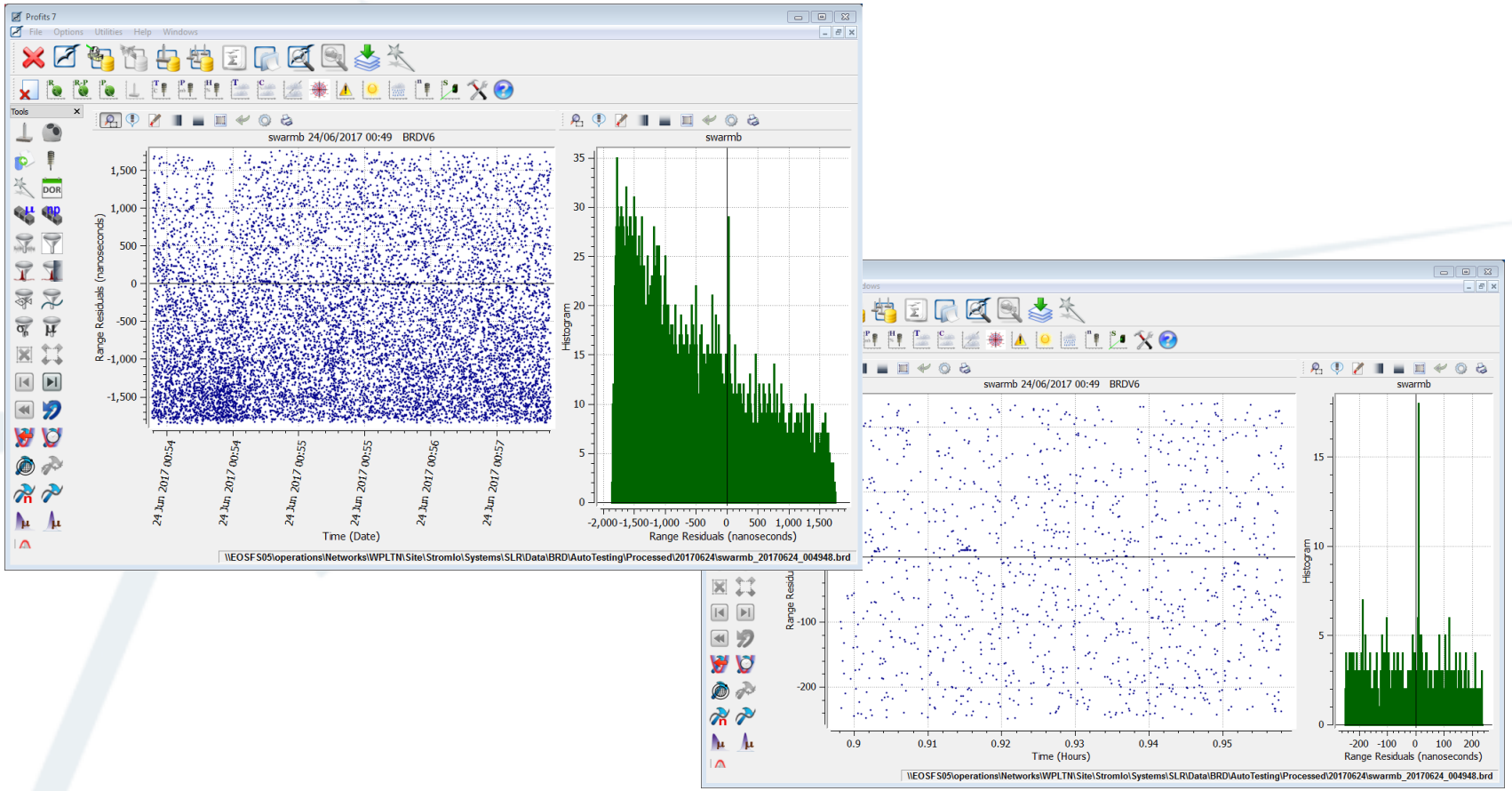
Autonomous Post Processing

Examples of a signal in cloud noise. More difficult to identify and extract the required signal.

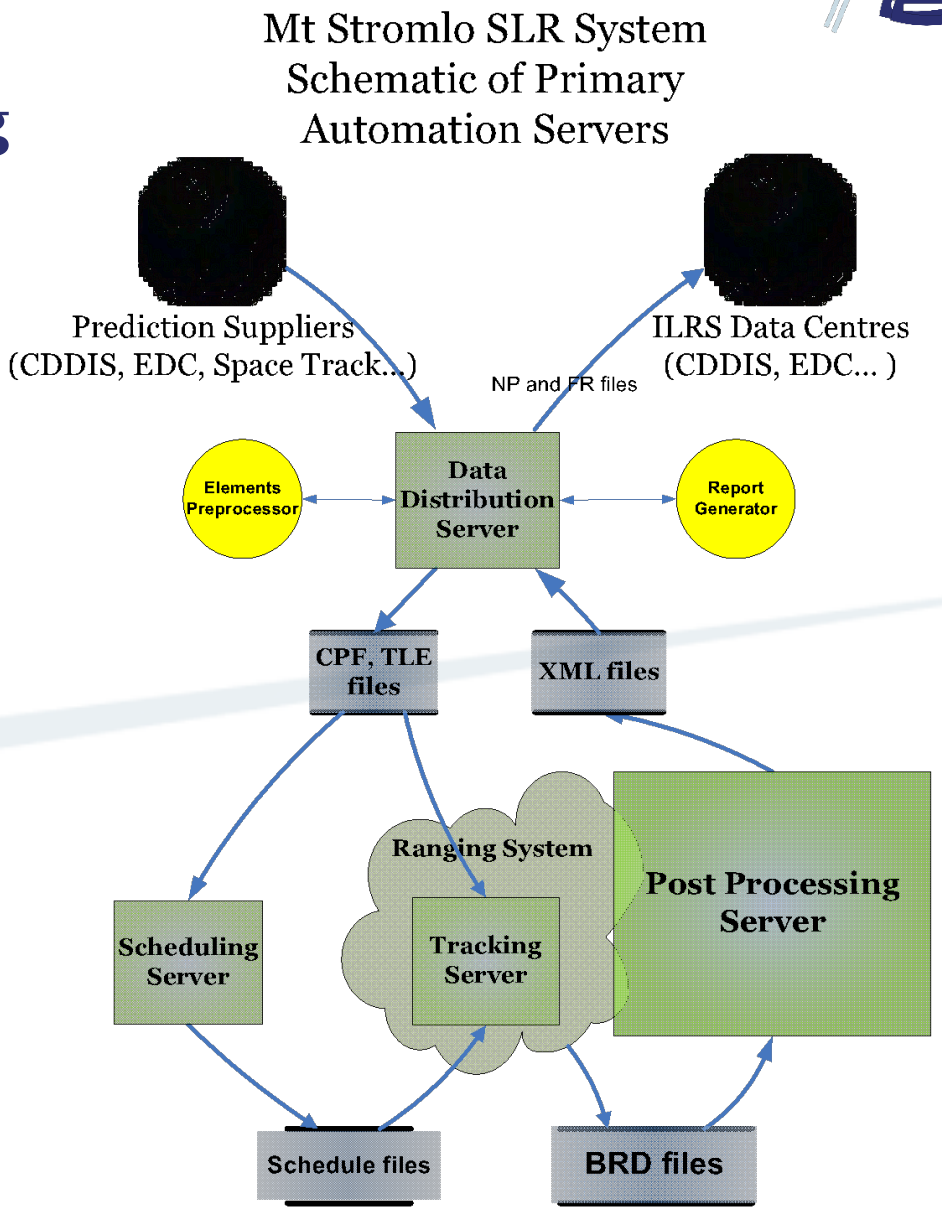


Autonomous Post Processing

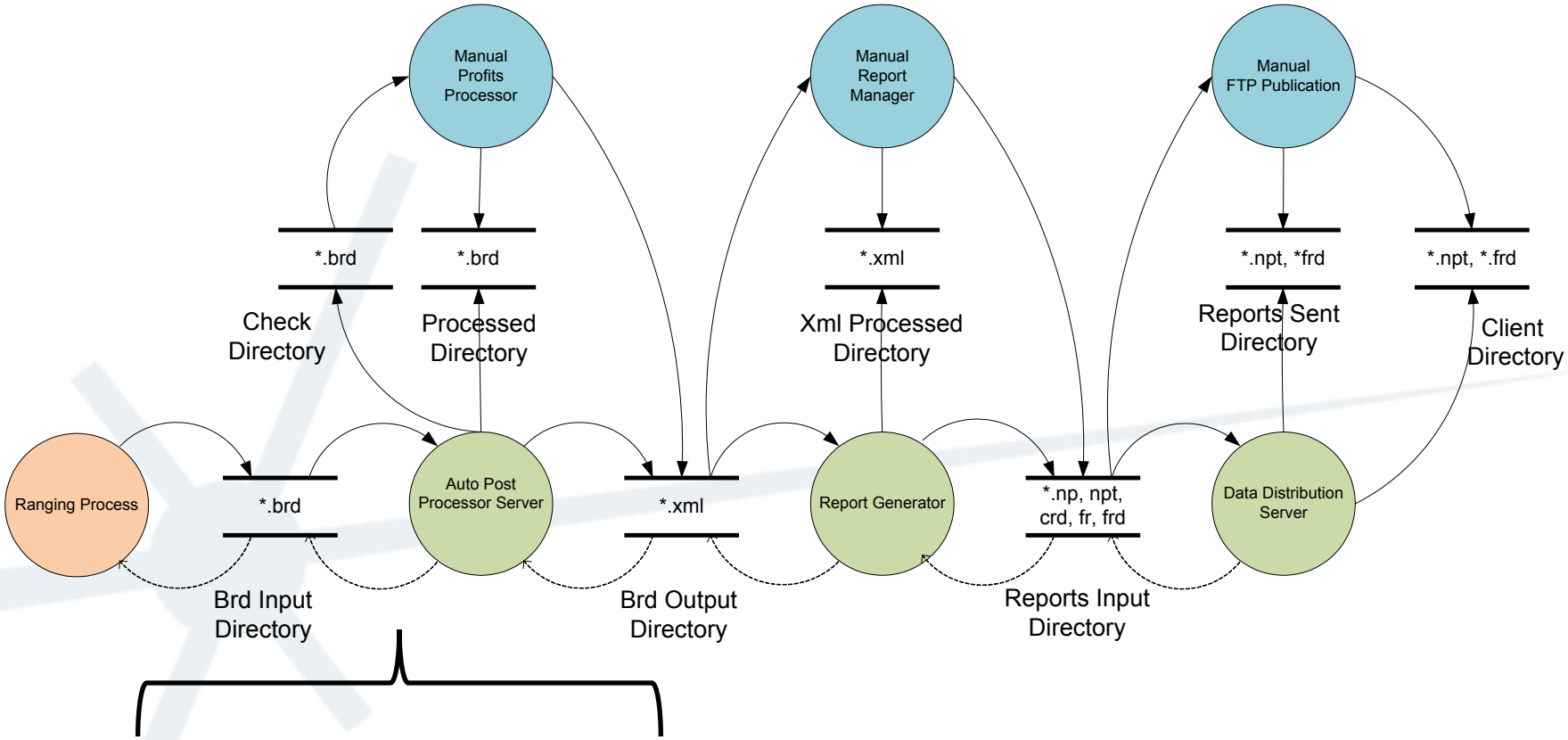
Examples of a weak signal in cloud noise. Very difficult to identify and extract the required signal.



Autonomous Post Processing



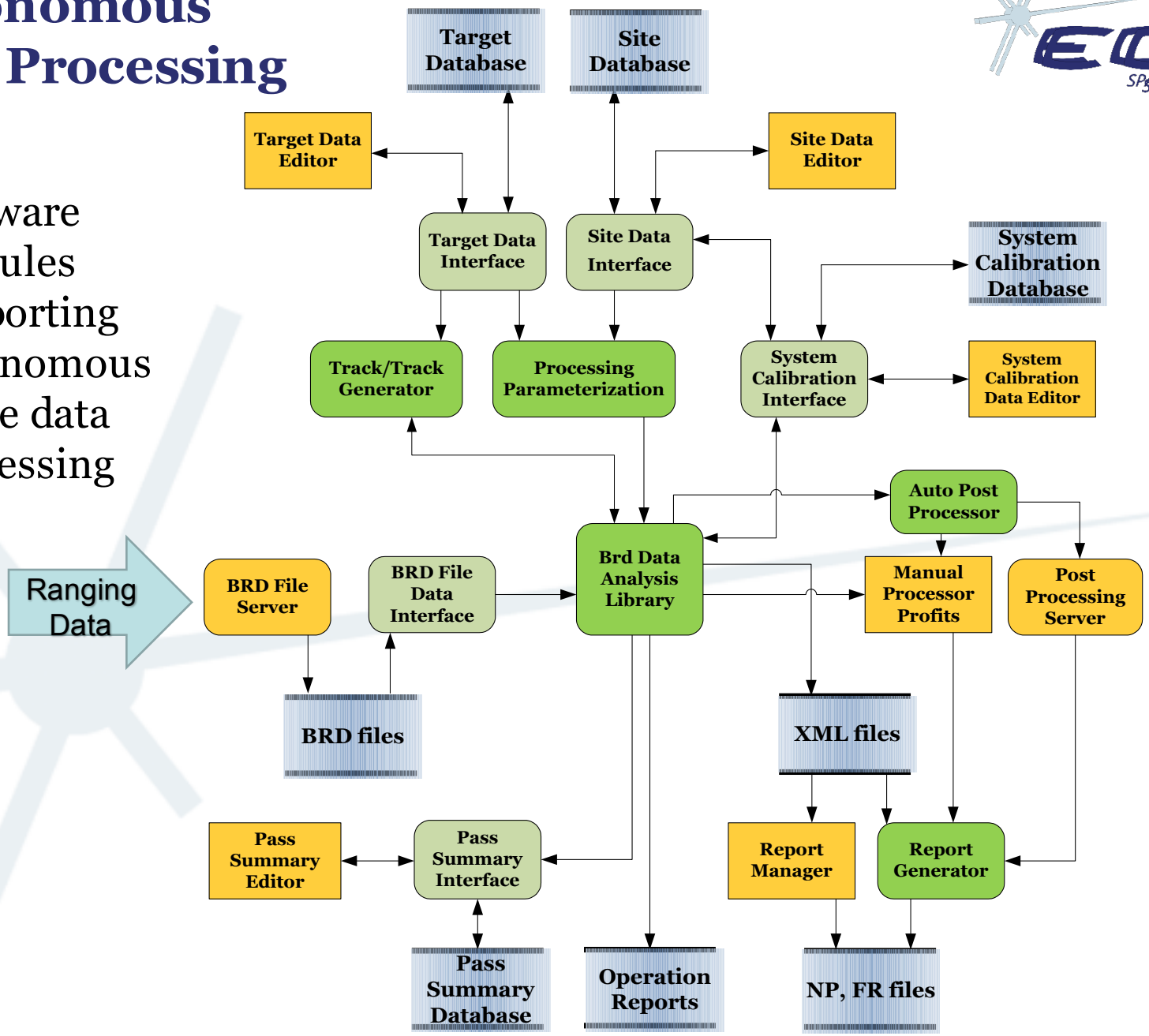
Autonomous Post Processing



BRD file post processing stream

Autonomous Post Processing

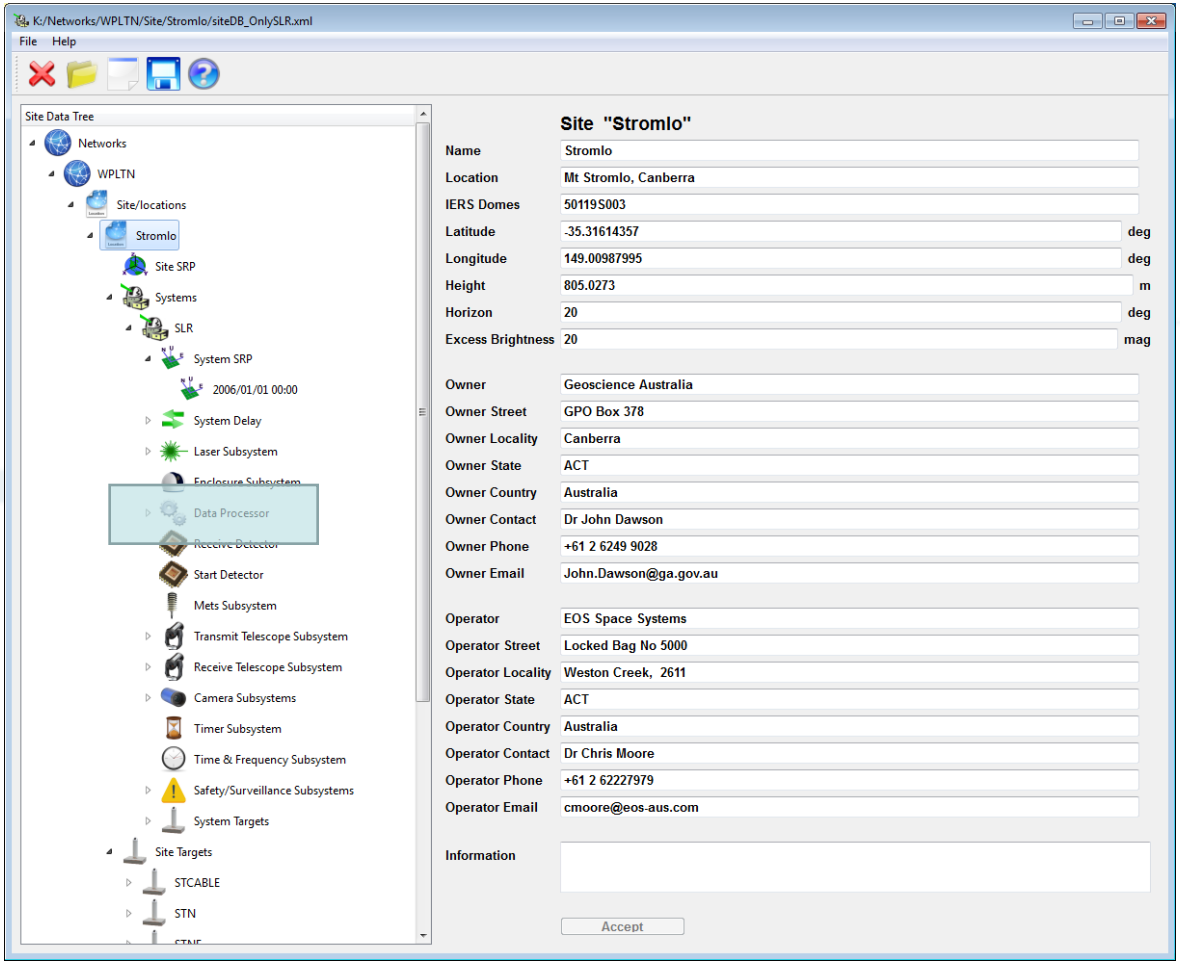
Software modules supporting autonomous range data processing



Autonomous Post Processing

Site Database

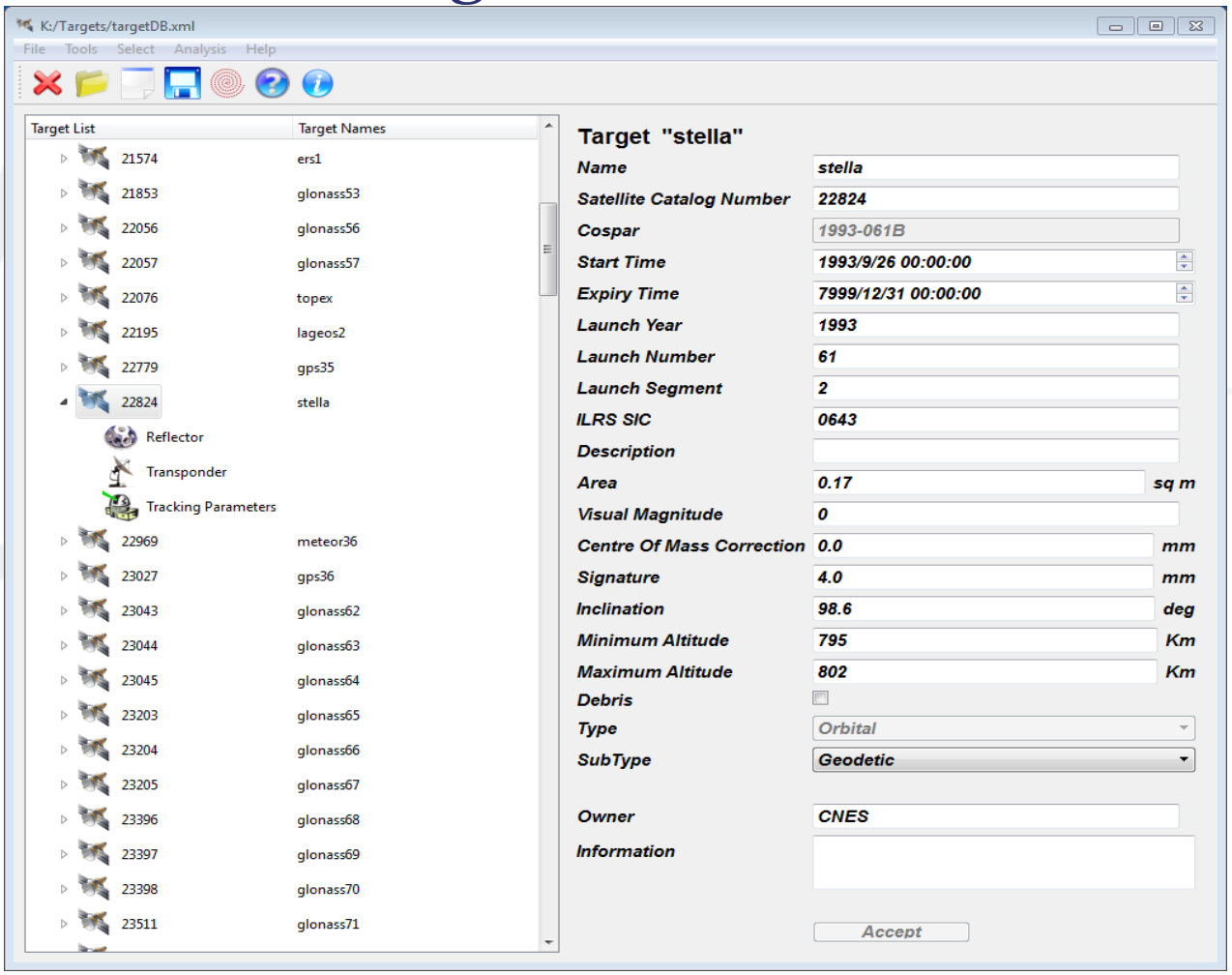
Site based parameterization for post processor



Autonomous Post Processing

Target Database

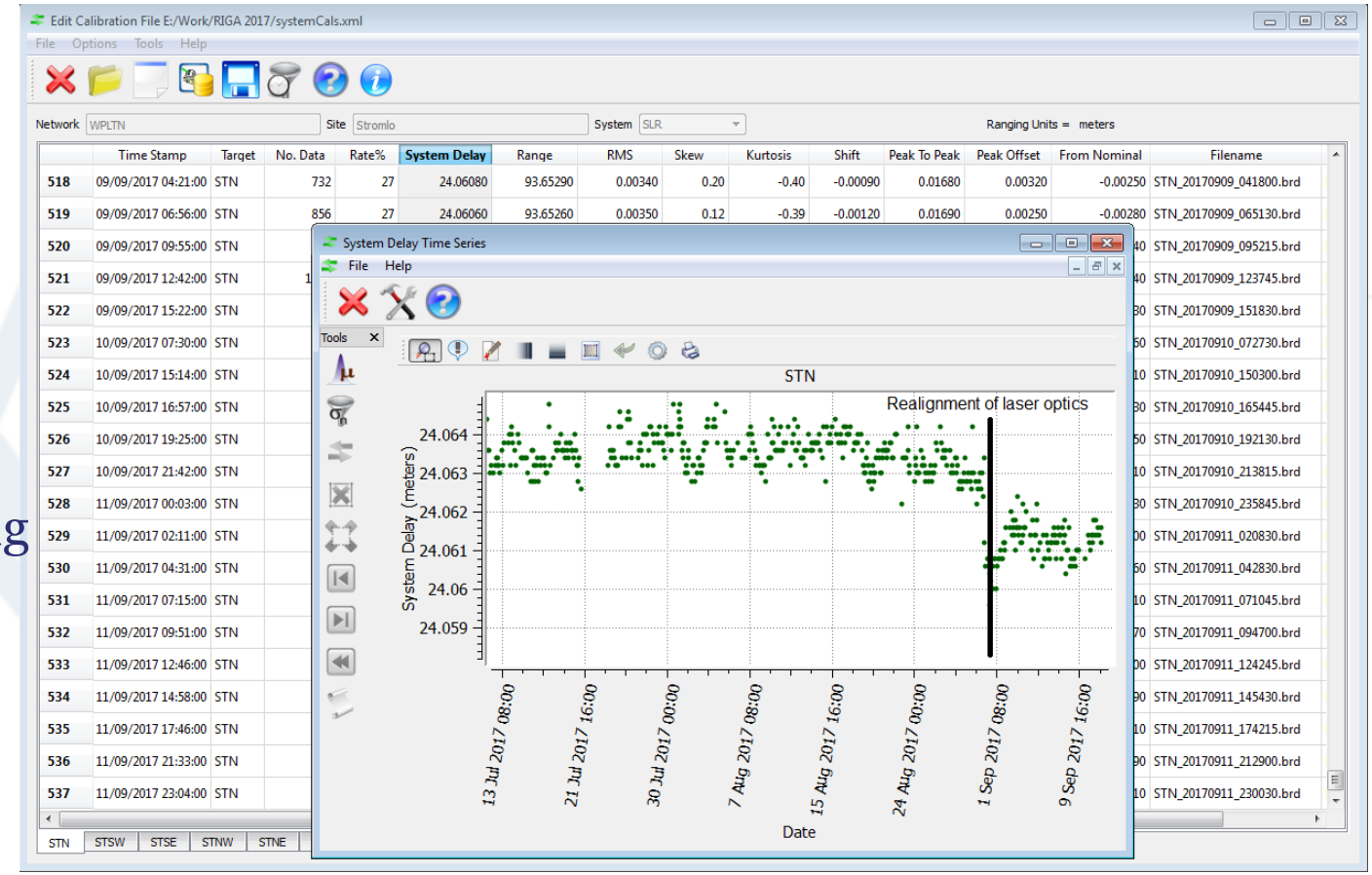
Target parameterization also used for post processing



Autonomous Post Processing

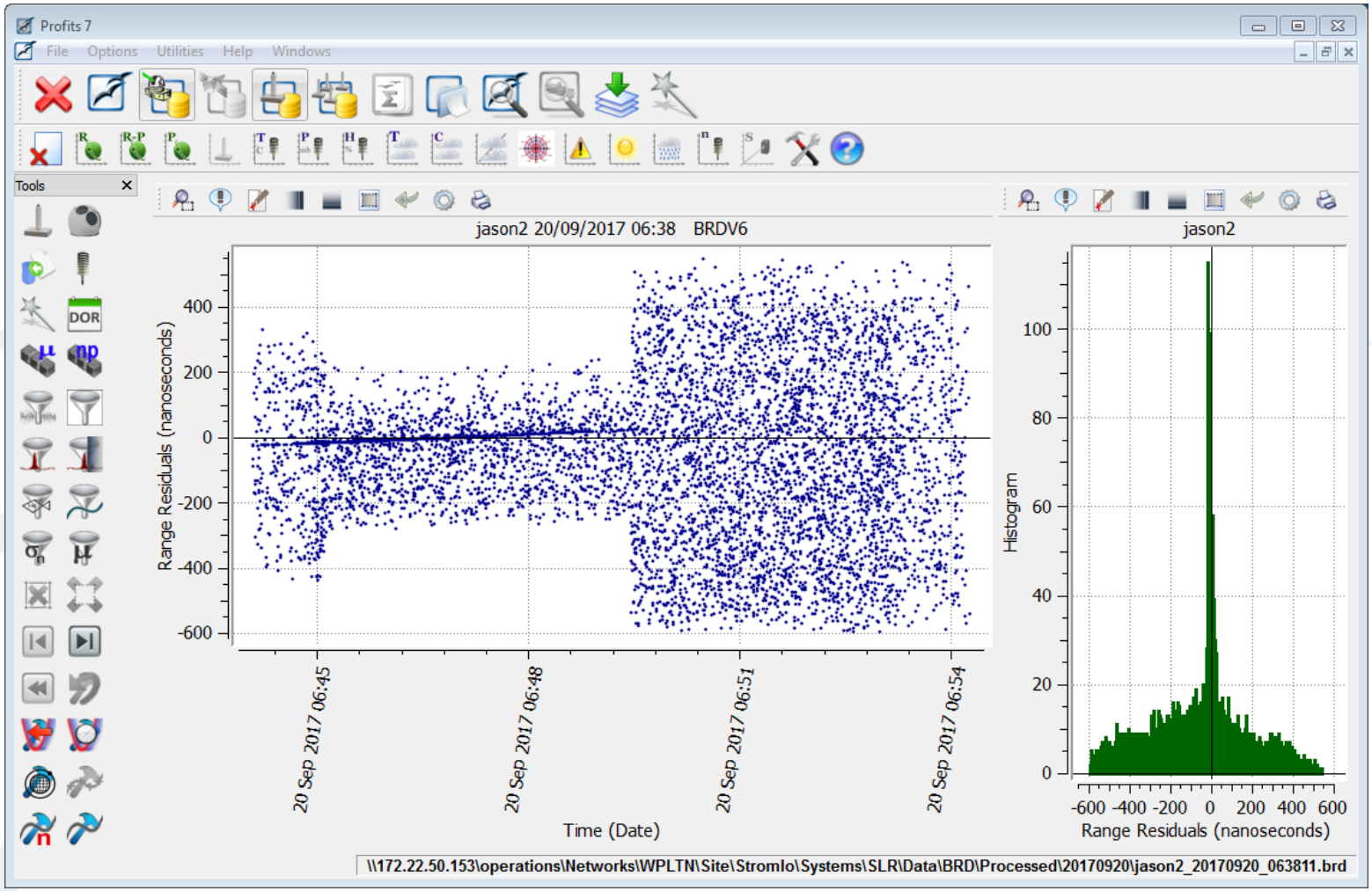
System Calibration Database

Pre and post system calibrations required for post processing



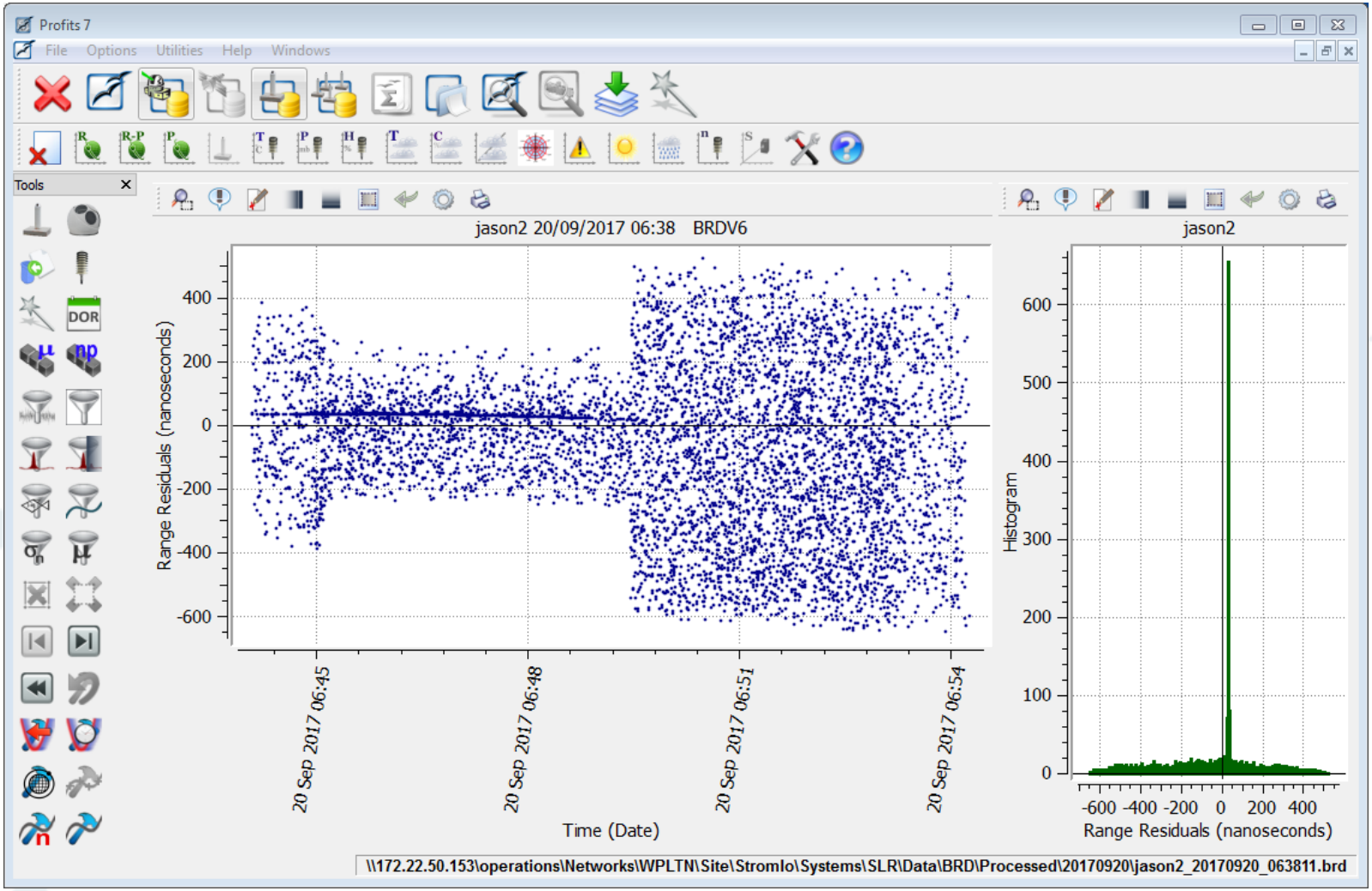
Autonomous Post Processing

Post Processing Steps – input range data



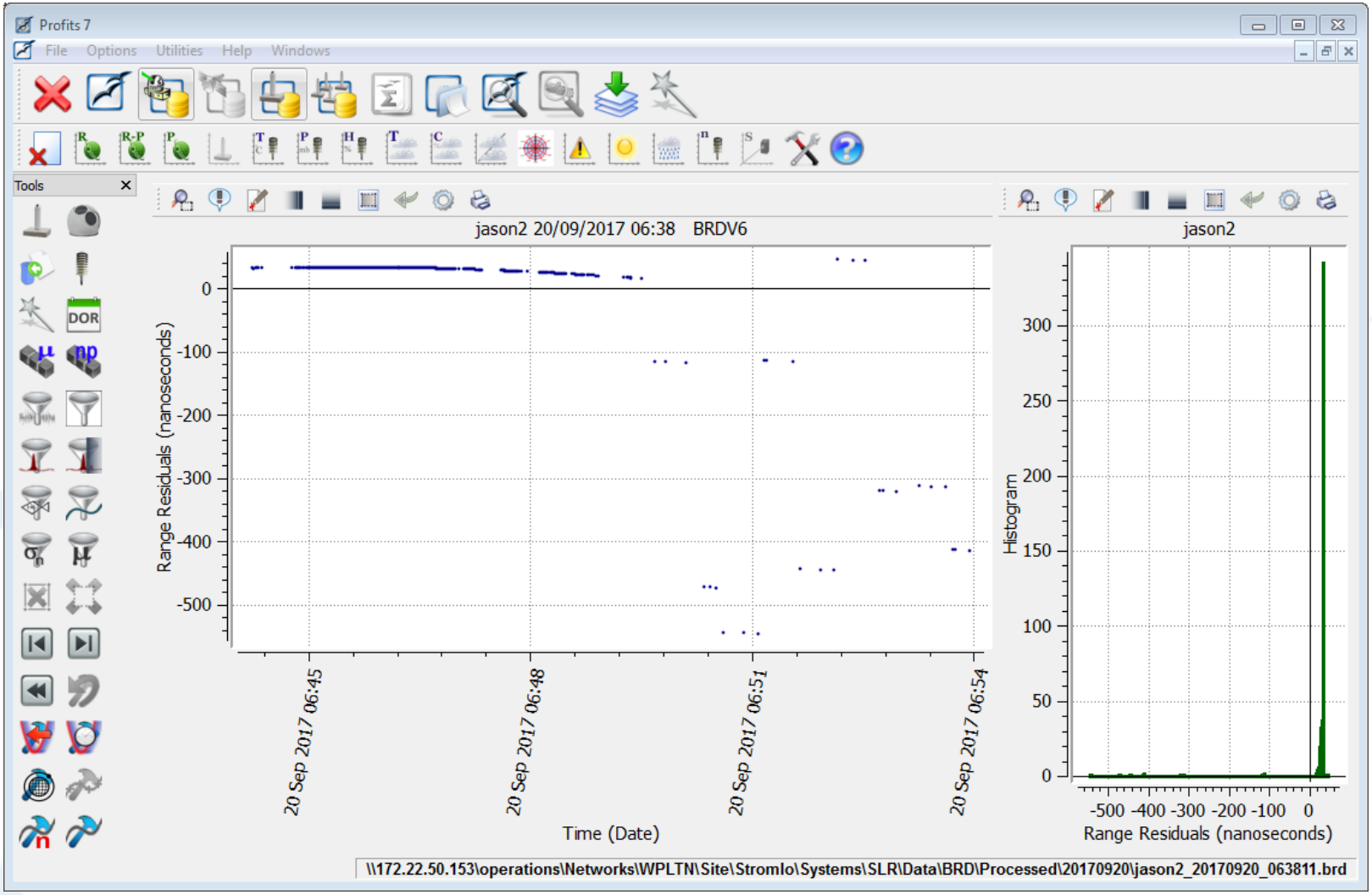
Autonomous Post Processing

Post Processing Steps – after time bias sweep



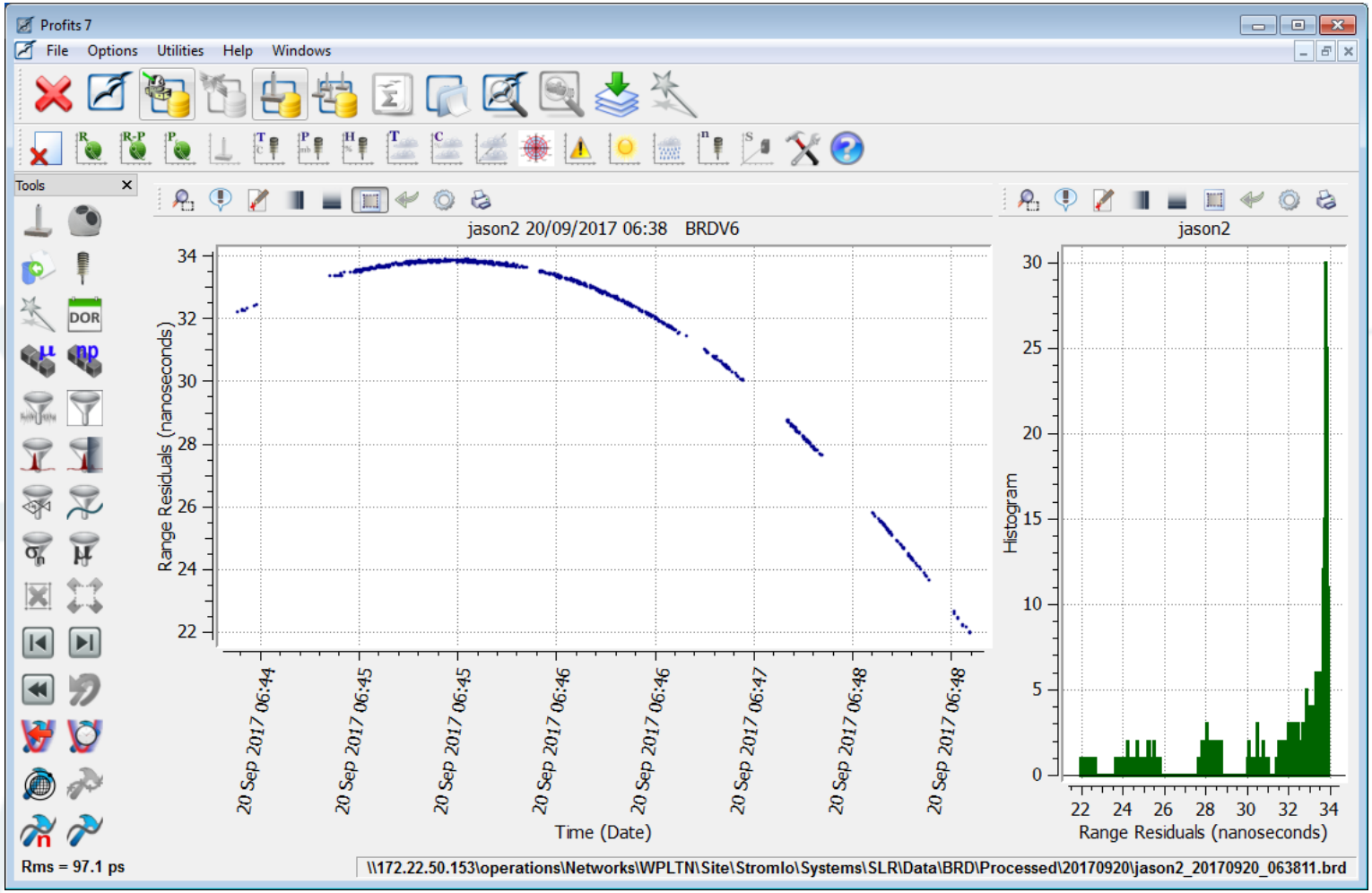
Autonomous Post Processing

Post Processing Steps – after Poisson filtering



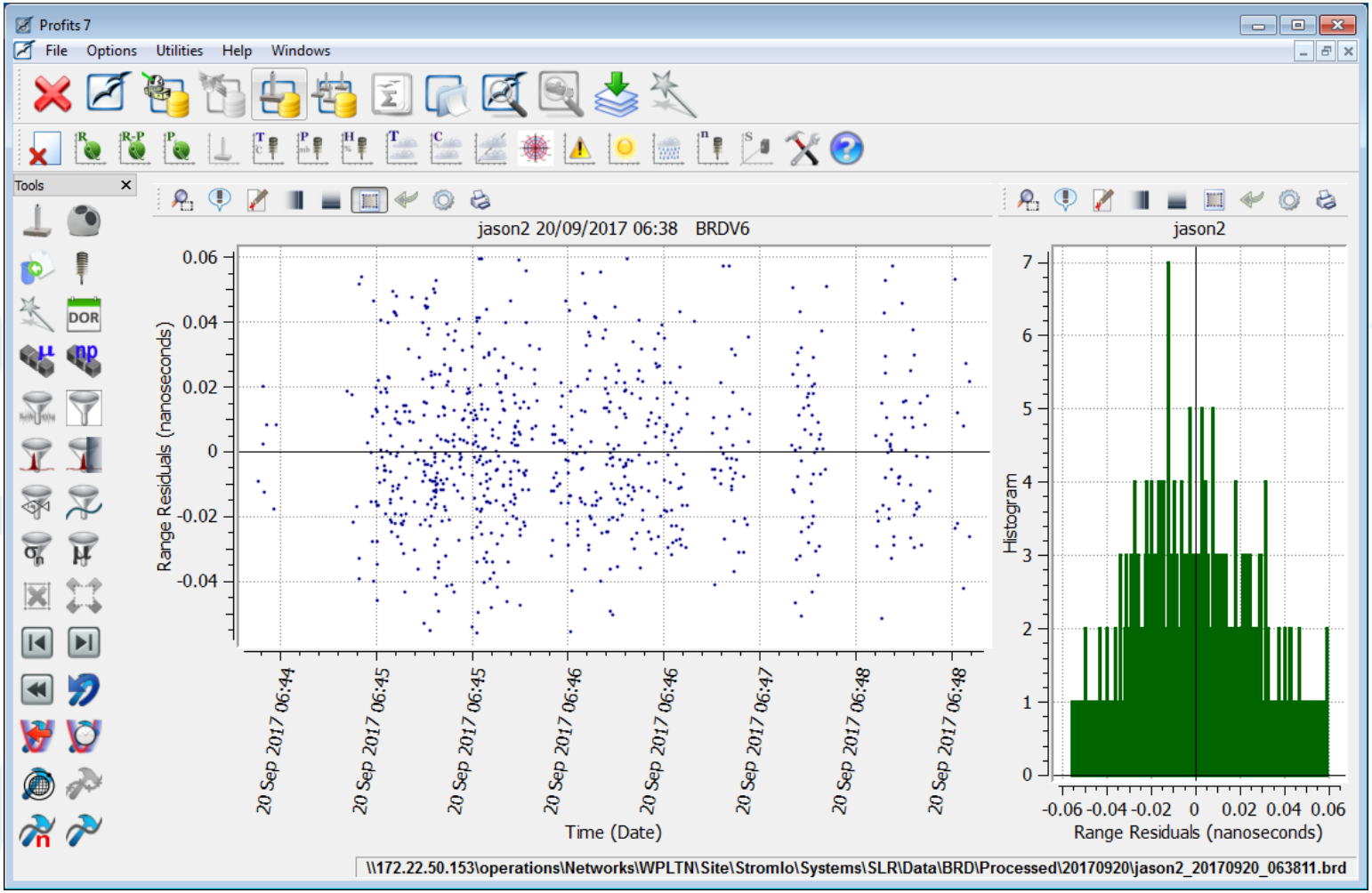
Autonomous Post Processing

Post Processing Steps – after polynomial filtering



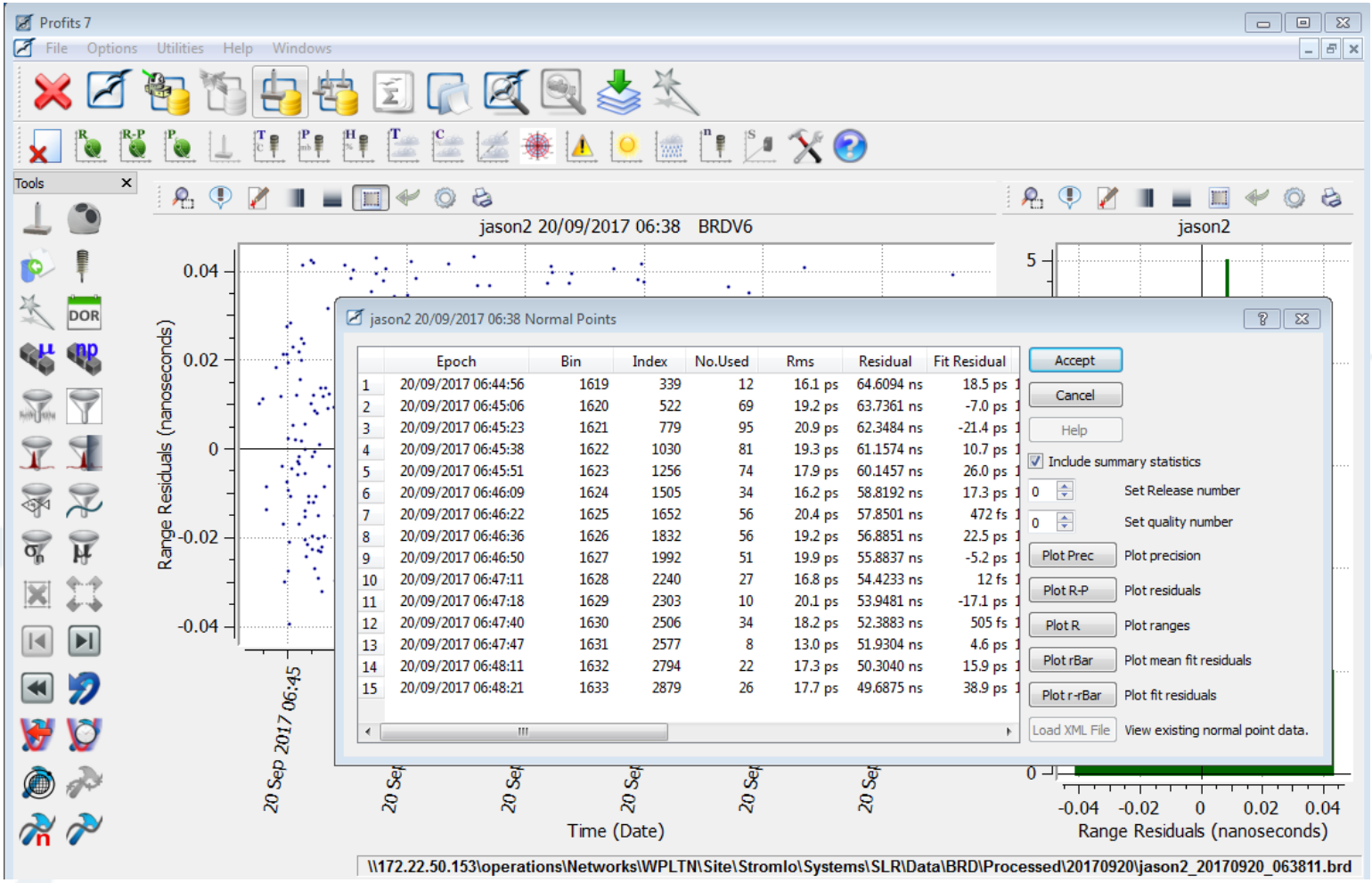
Autonomous Post Processing

Post Processing Steps – after polynomial fitting



Autonomous Post Processing

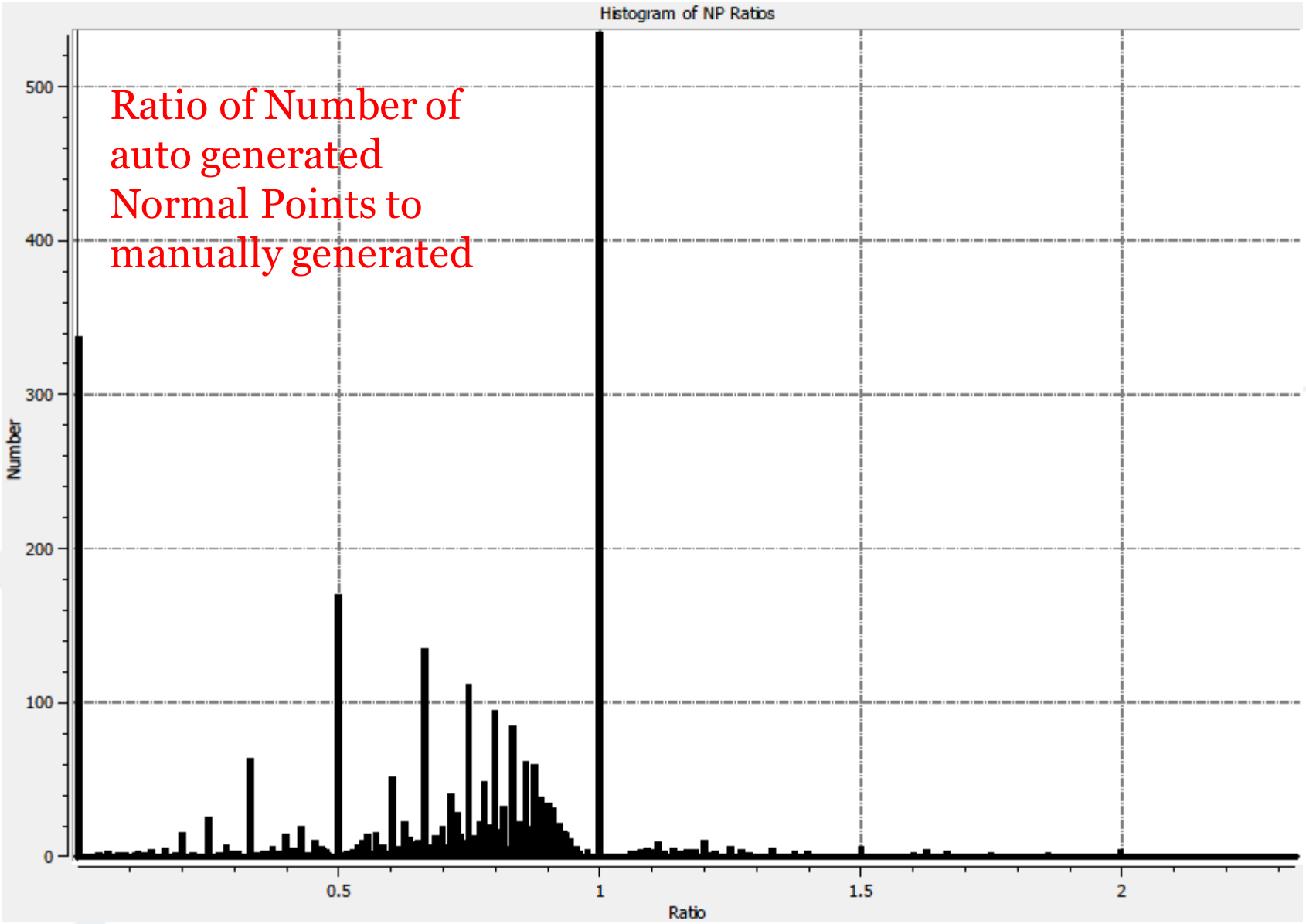
Post Processing Steps – Normal Point Generation



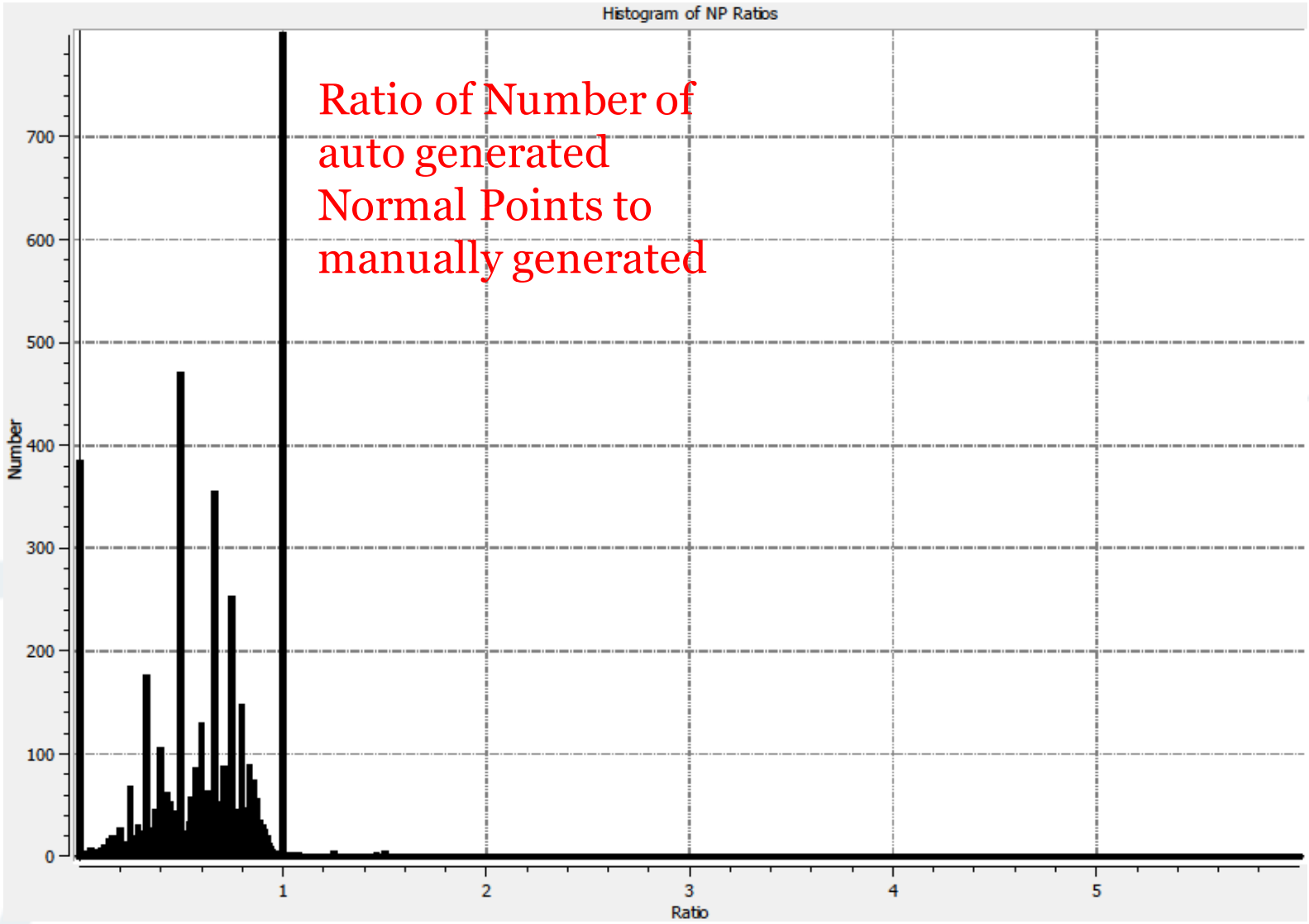
Autonomous Post Processing Analysis

- Analysis of differences between manually and automatically processed BRD files.
- Post processor server supports repeated re-analysis of BRD files.
- Allows comparison of results from ~20000 BRD files (obtained in 2017).
- Able to identify and focus on outliers.
- Provides average statistics.

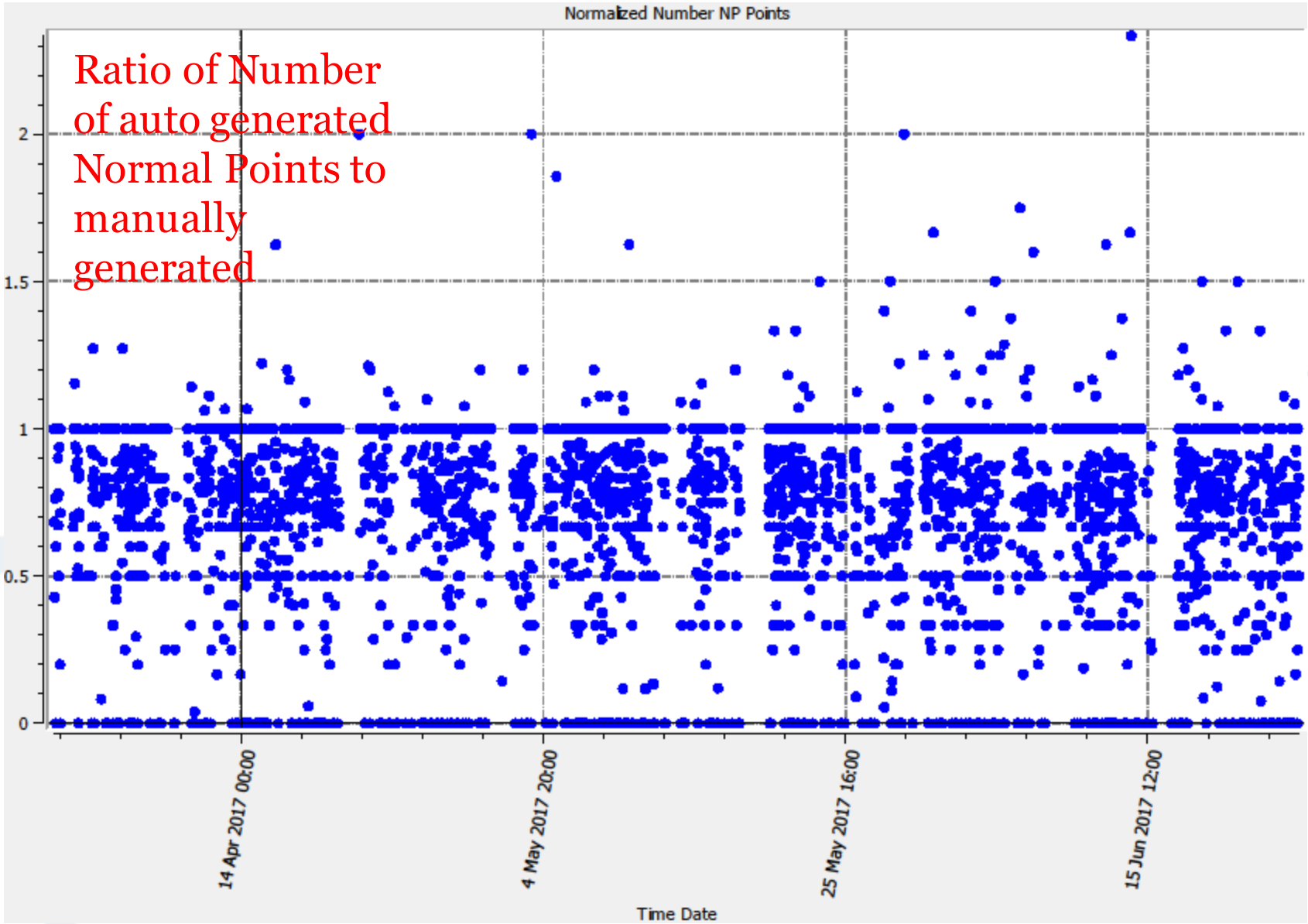
Autonomous Post Processing Analysis



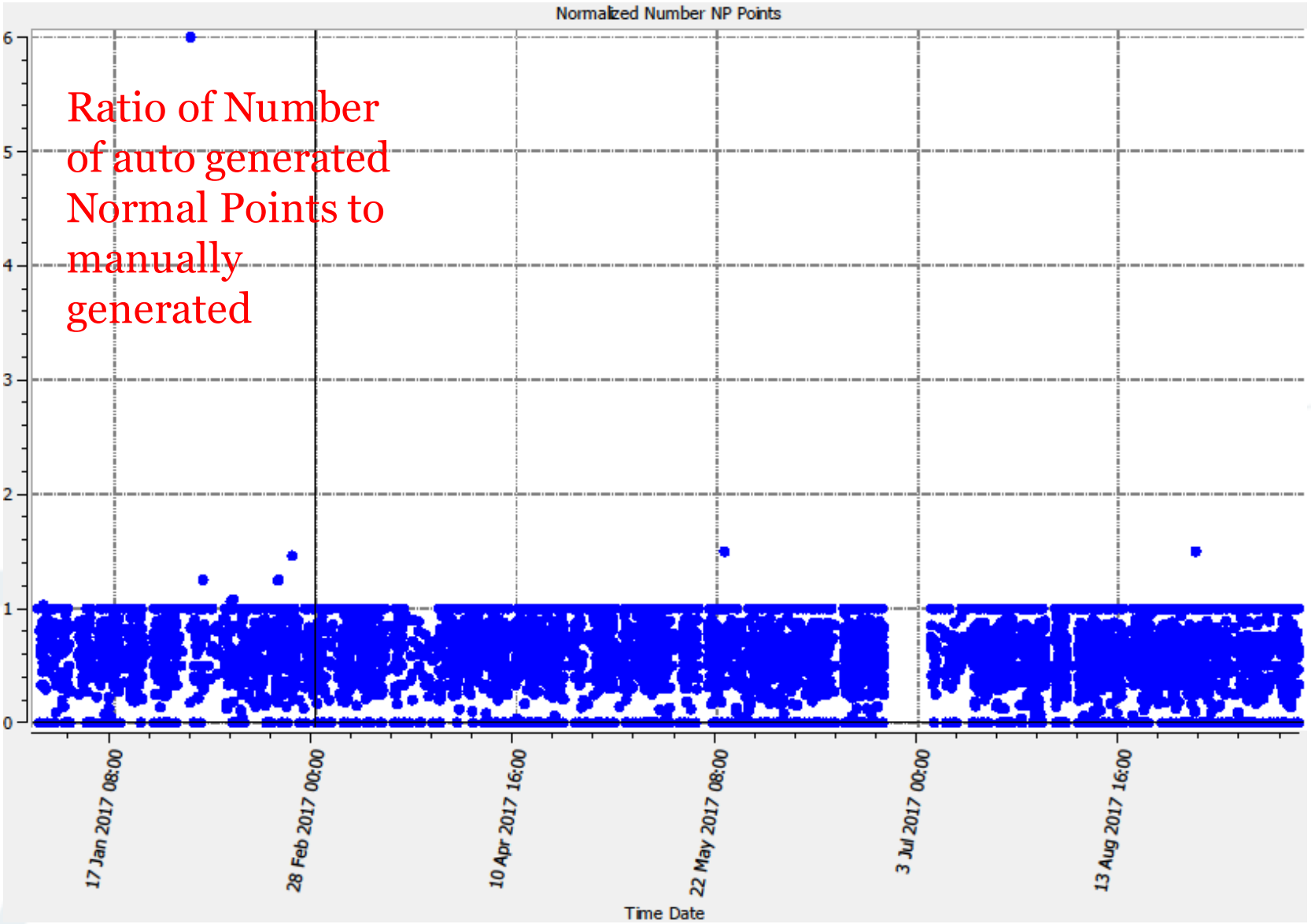
Autonomous Post Processing Analysis



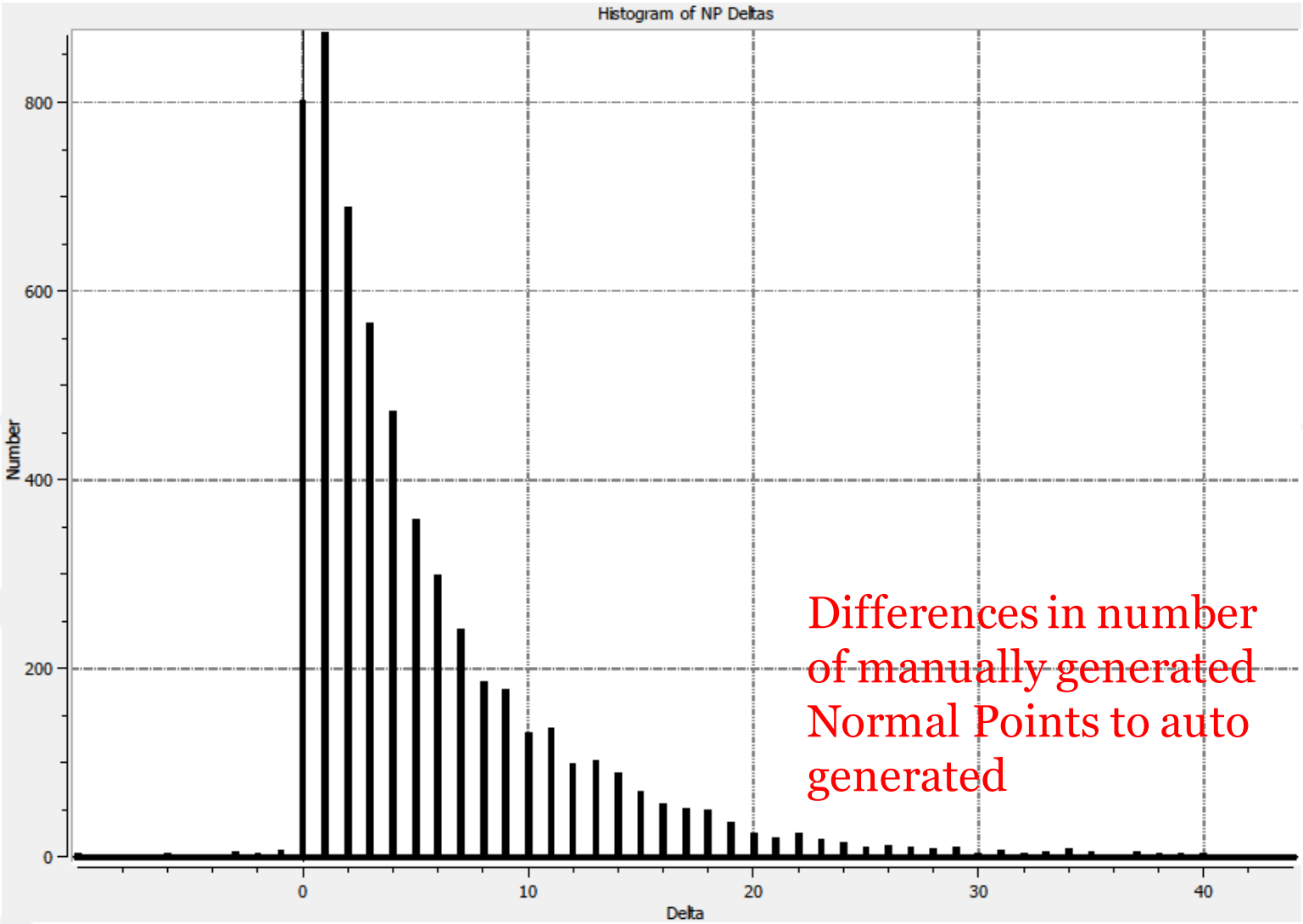
Autonomous Post Processing Analysis



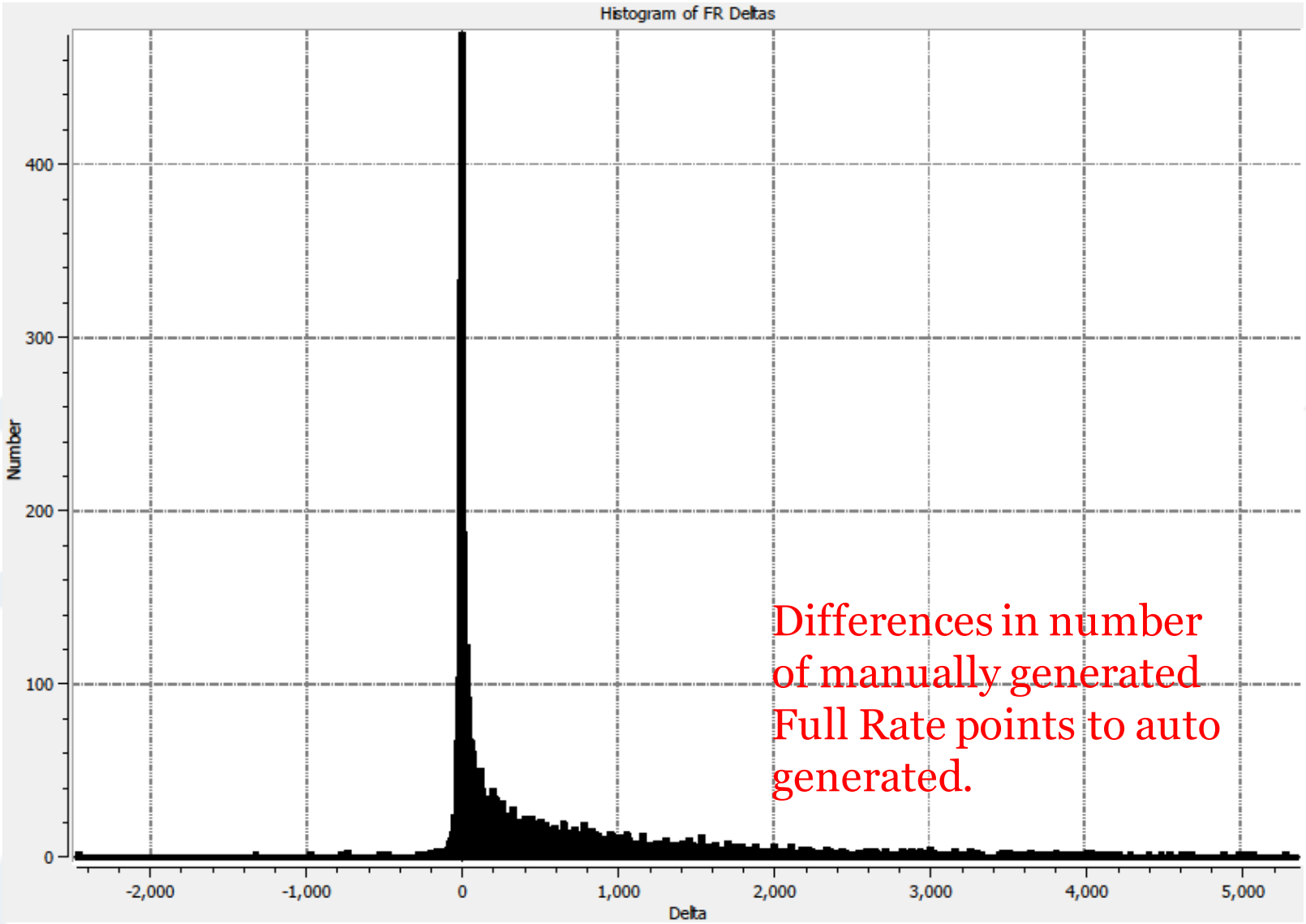
Autonomous Post Processing Analysis



Autonomous Post Processing Analysis



Autonomous Post Processing Analysis



Summary & Plans

- Automated processing well developed.
- Refinements to remove spurious points.
- Switch publication of results from manually processed to automated.
- Continuous improvement to reduce lost points.

Binary Range Data files (*.BRD)

- Captures raw data from the ranging system, including;
  - Pass metadata
  - Shot Events
  - Mets, Cloud data
  - Telescope Pointing
  - Prediction Element(s)
  - System State/Interlocks
  - Current site database
  - Current Target characteristics
- Stored as serialized files using Google's Protocol Buffers.
- Input to post-processing stream.

Binary Range Data files (*.BRD)

Protocol Buffers is used to serialize Ranging data into *.BRD files.

- BRD files <50% size of binary files and much smaller than XML etc.
- Support fast processing.
- Supports backward compatibility.
- Schema based. Maybe support sharing data.

Tip: Consider using Google's Protocol Buffers

Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data. A good solution for capturing SLR shot data for subsequent processing. Protobuf files are compact and support fast processing. Content can be easily upgraded while maintaining backward compatibility. Experience has shown that protobuf files of ranging data are 50% (small files) to 80% (large files) smaller than even binary files with fewer fields, yet have the advantages of self-describing formats like JSON or XML.

See: https://developers.google.com/protocol-buffers/docs/reference/overview

"Protocol buffers are a flexible, efficient, automated mechanism for serializing structured data – think XML, but smaller, faster, and simpler. You define how you want your data to be structured once, then you can use special generated source code to easily write and read your structured data to and from a variety of data streams and using a variety of languages. You can even update your data structure without breaking deployed programs that are compiled against the "old" format."

Thought:

Protocol buffers require a *schema* for the definition of the data fields to be stored in protobuf files. If adopted and the schema was common to all SLR stations, then it may be possible for SLR stations to read (and process) each other's ranging data files.

Could this be a good thing???