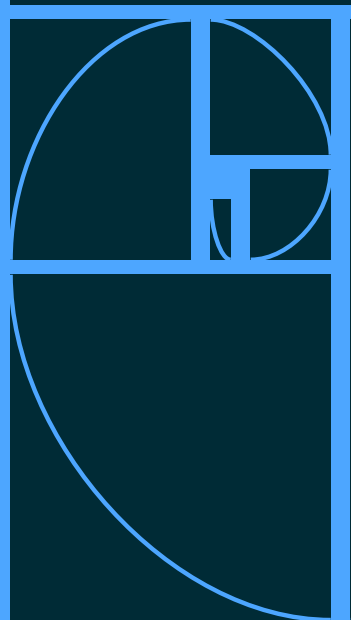
TELLIE PCA: Processing Automation

Report
November 7, 2022

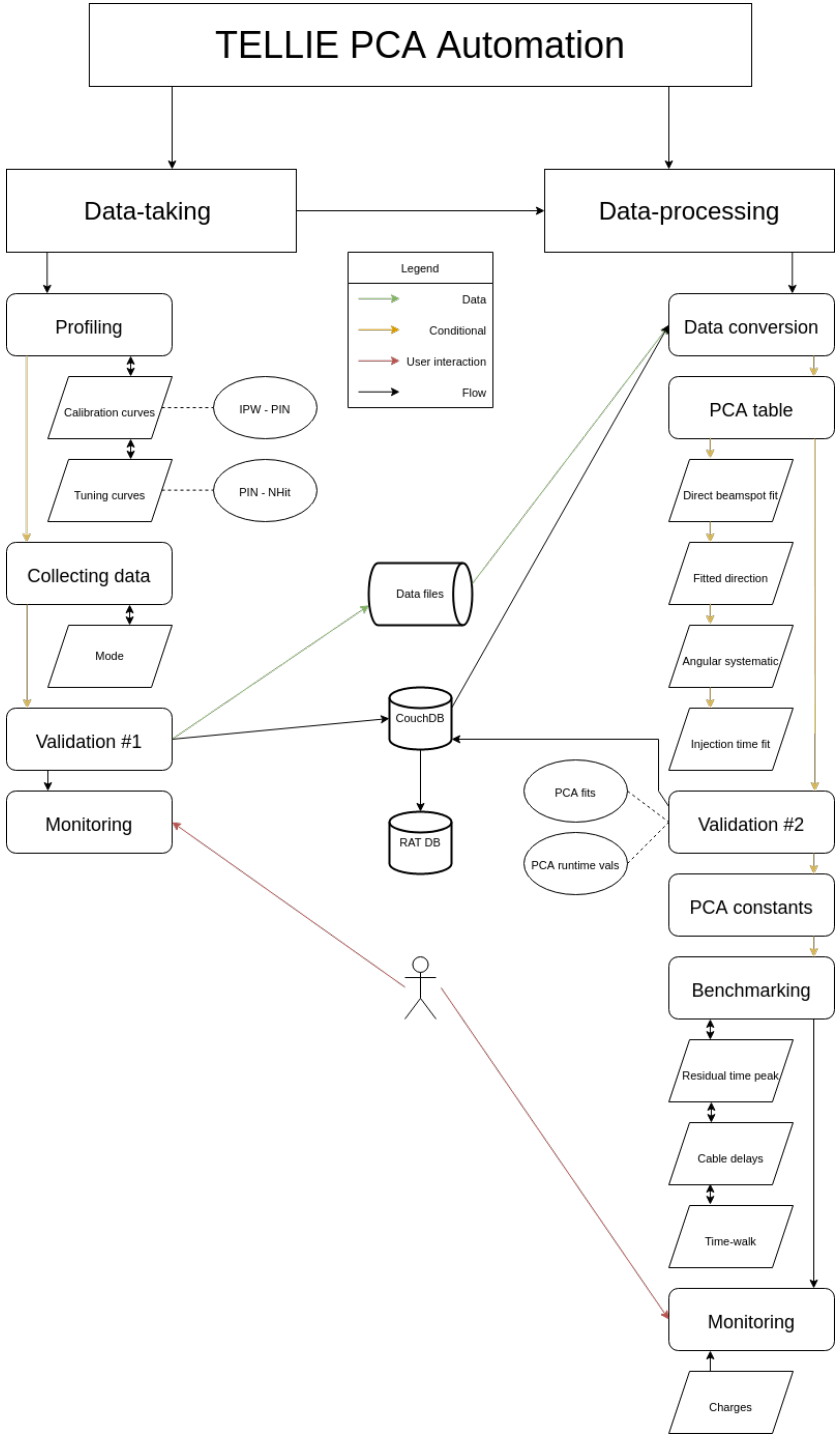


Michal Rigan
mrigan@snolab.ca

University of Sussex

Processing automation - Why

- extracting and validating PCA constants from data is complex...
- **Goal**: streamline (possibly speed up) the process of obtaining the PCA constants from data
  - regardless of the method to obtain the data
  - modular
  - require minimum human input
  - provide monitoring


TELLIE PCA Automation
Data-taking
Data-processing
Legend
Data
Conditional
User interaction
Flow
Profiling
Calibration curves
IPW - PIN
Tuning curves
PIN - NHit
Collecting data
Mode
Validation #1
Monitoring
Data files
CouchDB
RAT DB
Data conversion
PCA table
Direct beamspot fit
Fitted direction
Angular systematic
Injection time fit
PCA fits
PCA runtime vals
Validation #2
PCA constants
Benchmarking
Residual time peak
Cable delays
Time-walk
Monitoring
Charges

Processing automation - What

- *validate #1* $\rightarrow$ validates data is 'good enough' for PCA $\rightarrow$
- *PCA table* $\rightarrow$ fits for required corrections: beamspot fit, fibre direction, angular systematic, injection time
- *PCA table* $\rightarrow$ compare these fits and runtime values to previous set (stability)
- *validate #2* $\rightarrow$ validates fits are 'sensible'
- *PCA constants* $\rightarrow$ extracts PCA constants (PCA processor)
- *Benchmarking* $\rightarrow$ benchmarks the constants against previous set
- *Monitoring* $\rightarrow$ provides monitoring of each step, and between datasets (!)

Processing automation - Validations

Run series of checks:

- Validation #1:
  - correct fibre, number of events (EXTA), passed hits, cuts on PMTs, checks on LPC, run length, frequency
  - NHit distribution, NHit over time, delays
  - time of hits over time, # peaks, PMTs in beamspot, PMT occupancy
  - PIN, PIN vs NHit, events over subruns, ... (21 total)
- Validation #2:
  - for each correction: check mean, rms, min, max
  - residual times: distribution, # peaks, function of angle
  - evaluate trends (12 total)
- this is available on monitoring page (flags $\rightarrow$ bitword)

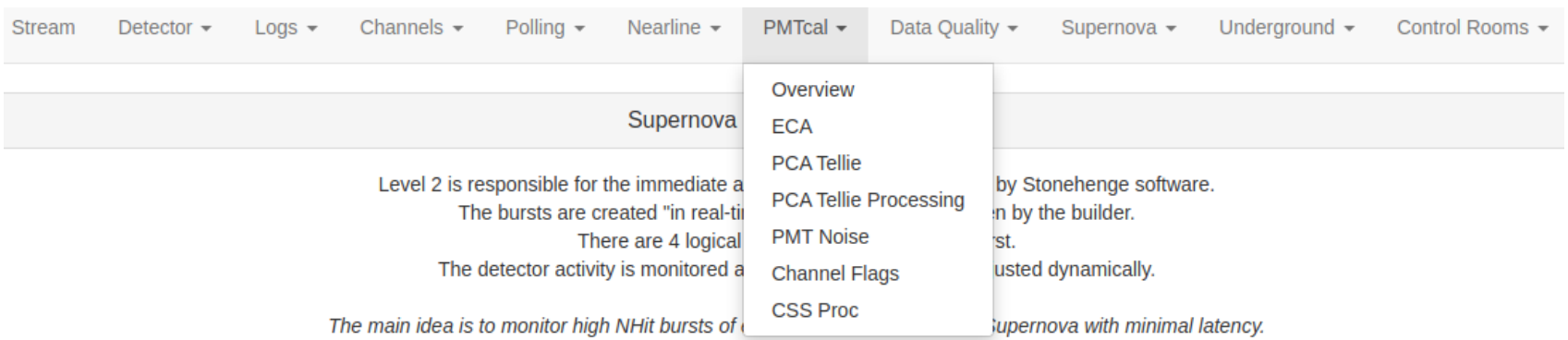
Processing automation - Benchmarking

- compare PCA values (cable delays, TW fit) to previous set
- apply these constants to a well understood run
- extract the residual hit times distribution
- monitor charges: threshold, peak, hhp for QHS & QHL

Processing automation - How

- *simple* → only requires to provide a runlist
- *modular* → master script that spawns subprocesses, individual steps can be (re)run. Also allows for easier changes to modules
- *submission platform* → can queue processes, submit (up to a limit), monitor their status
- *customizable* → thresholds (other settings) are loaded from environment (tuning)
- *linked* → stores data in couchdb, ratdb, redis, provides plots to minard
- *regulation* → unifies cuts, data checks, event selection, ranges, ...
- *evaluative* → provides bitwords (flags) for fits / checks

Processing automation - Minard



Open questions

- tagging events outside Orca
- when (from) to apply new constants
- ensure recent ECA
- where to deploy
- data-taking (modes) implementation (#7612)

Processing automation - Next steps

- documentation
- minard PR
- help to deploy
- ...
- tuning of the threshold values
- PR for PCA Proc