



DEAP-3600 Analysis: looking for a needle in a haystack and some new capabilities at SNOLAB to make this easier

Chris Jillings

On Behalf of the DEAP Collaboration and SNOLAB

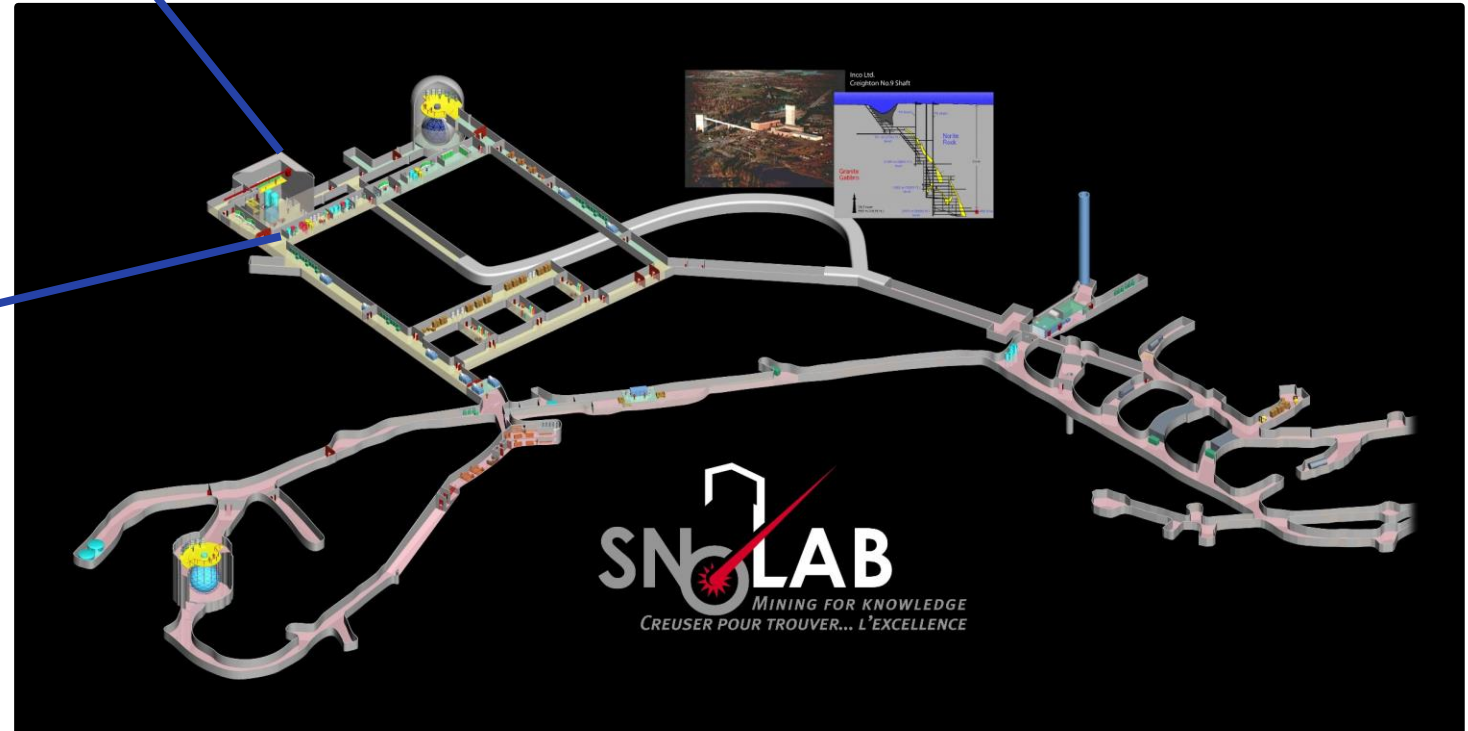




DEAP-3600 is in the SNOLAB Cube Hall



3279 kg of single phase liquid argon
for a sensitive high-mass WIMP search.





Stable Operation: Nov 2016 ...

Hardware:

Detector: Astroparticle Physics, Volume 108, Pages 1-23

PMT calibration: NIM A 922, April 2019, pp 373-384

First Fill (9.87 tonne day exposure, August 2016)

Phys. Rev. Lett. 121, 071801 (2018)

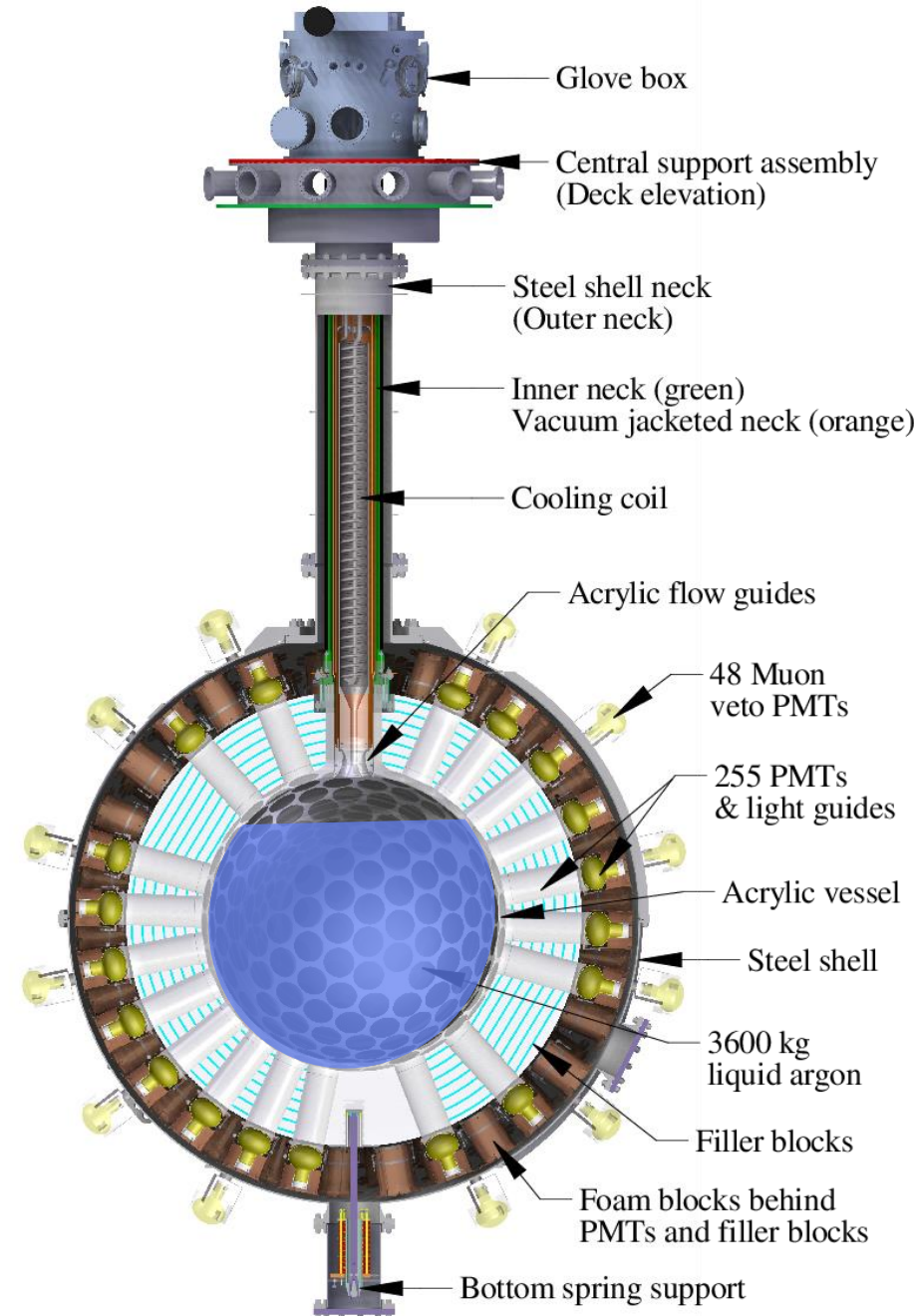
Second Fill (758 tonne day exposure, Nov 2016 - Oct 2017,
not blind)

Phys. Rev. D, 100, 022004 (2019)

EM Backgrounds, Ar-42, ... (Nov 2016 - Oct 2017, not blind)

arxiv.org: 1905.05811

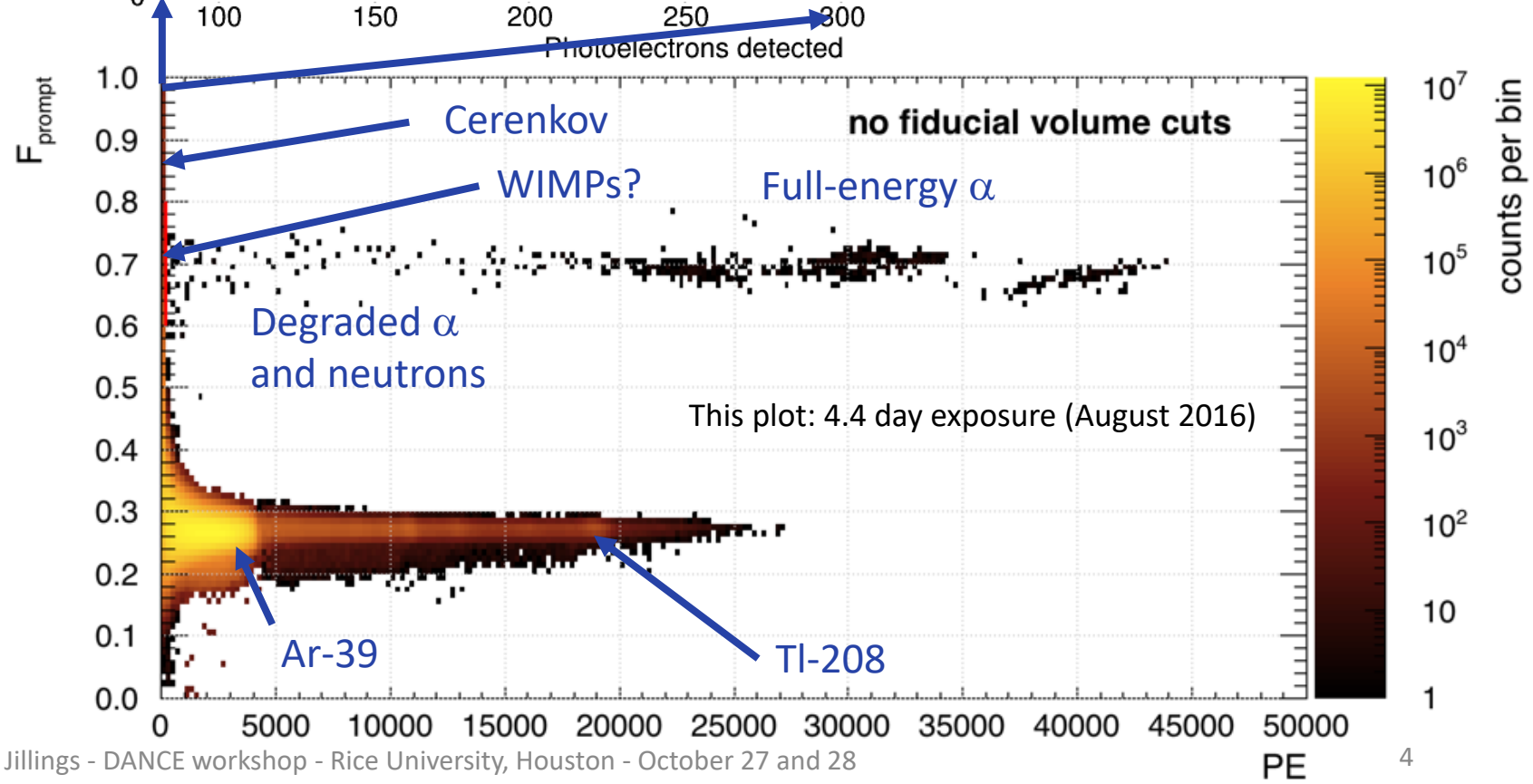
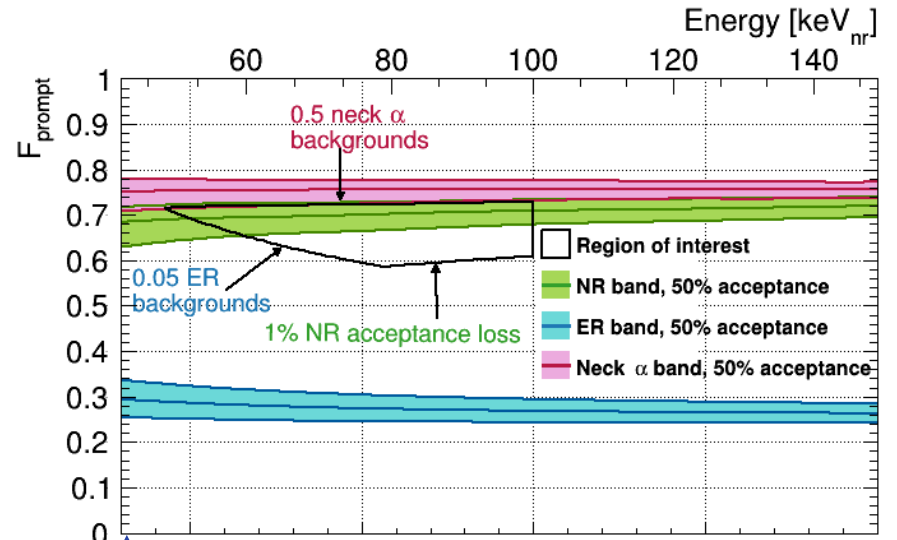
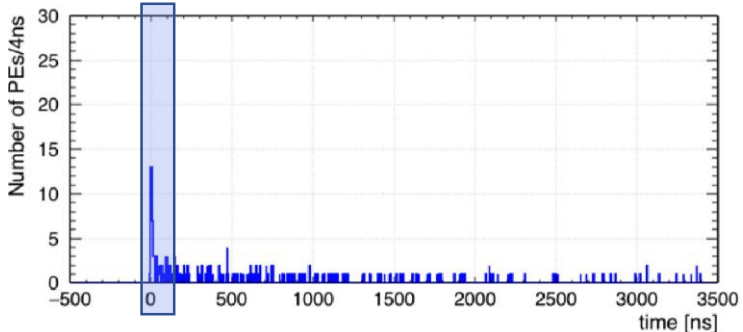
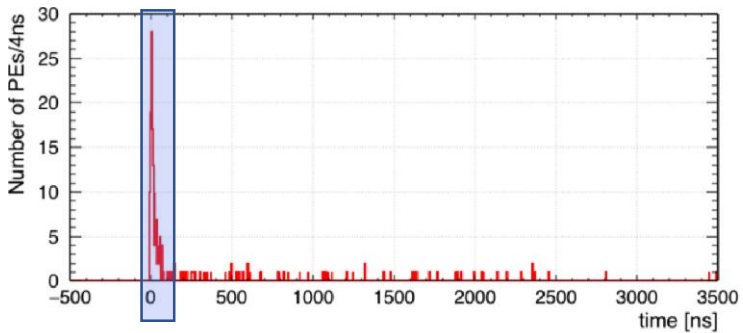
accepted by Phys. Rev. D



SOLID EDGE ACADEMIC COPY

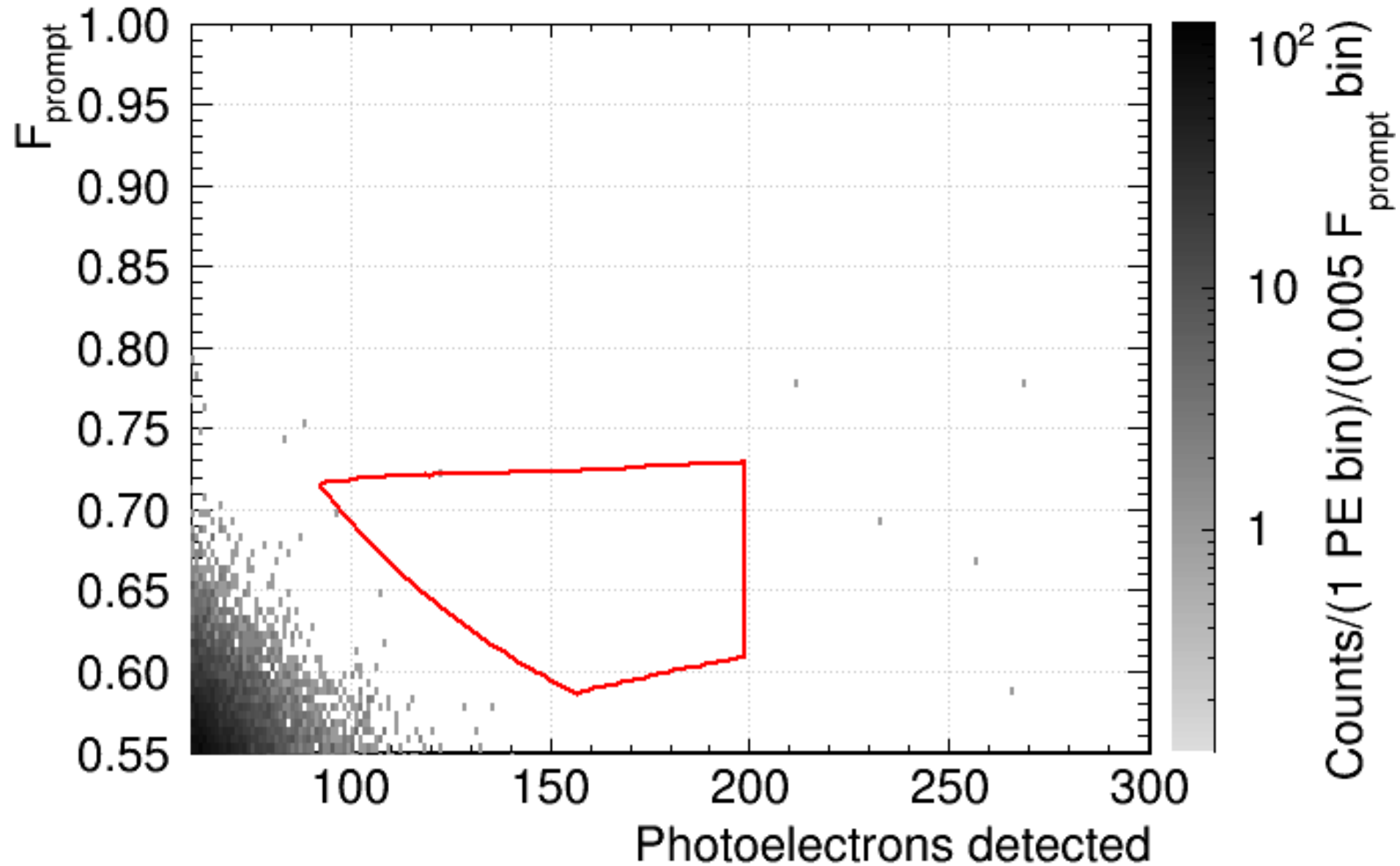


Overview of data



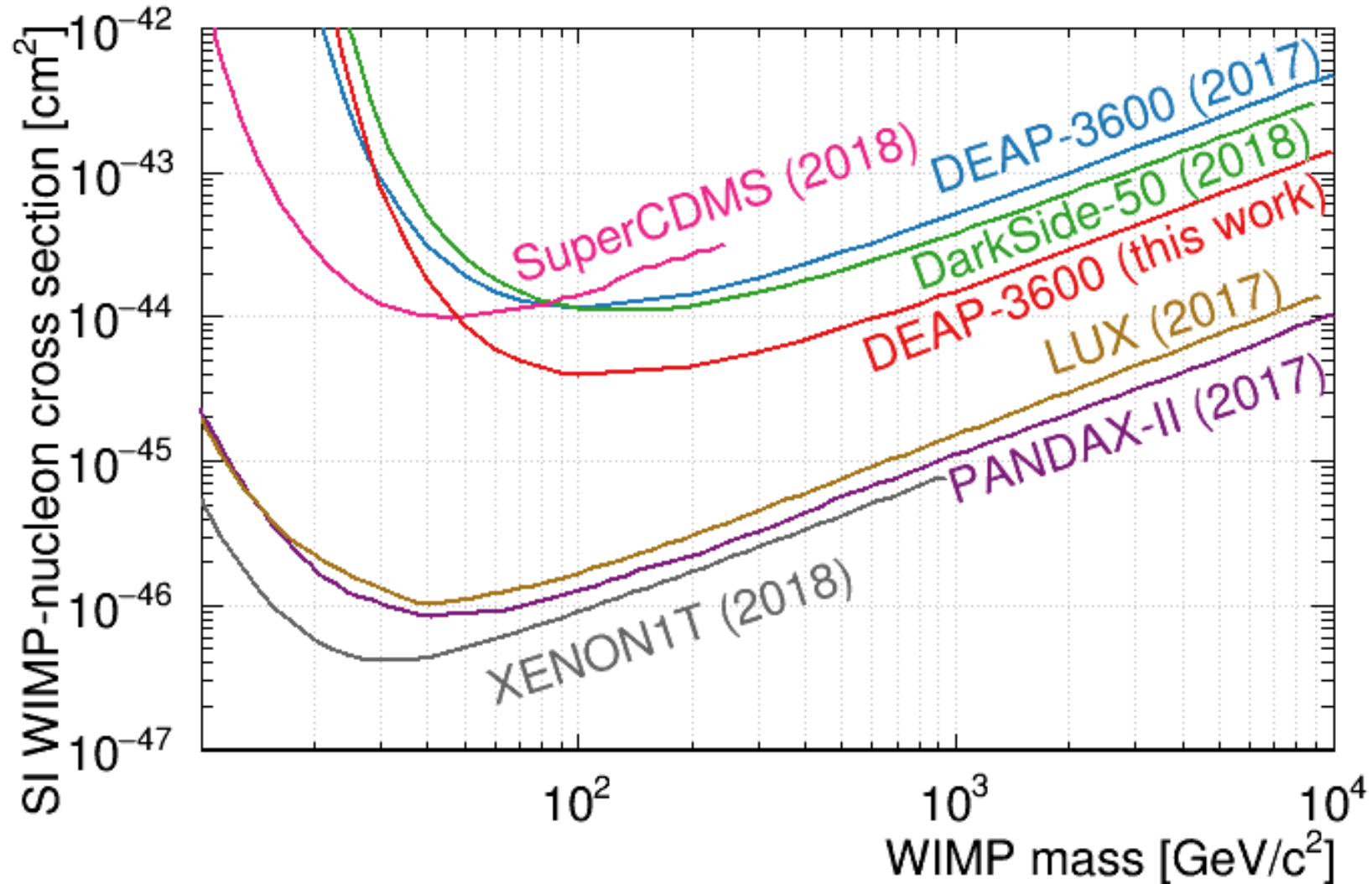


DEAP WIMP Search Region



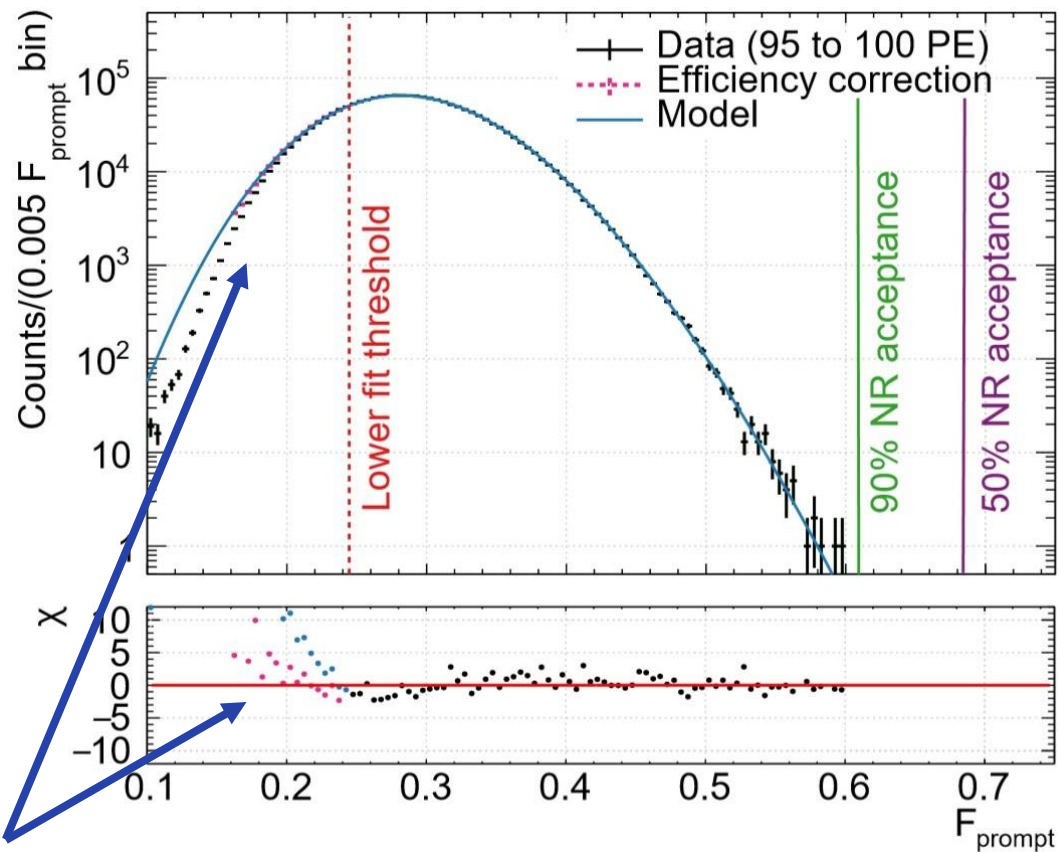


231-day exposure: first analysis





LEAr Pulse-Shape Discrimination: works as advertised. Lower Boundary of WIMP ROI tuned to 0.03 ± 0.01 bg event.



Mathematics of trigger correction:
EPJ C (2019) 79: 322

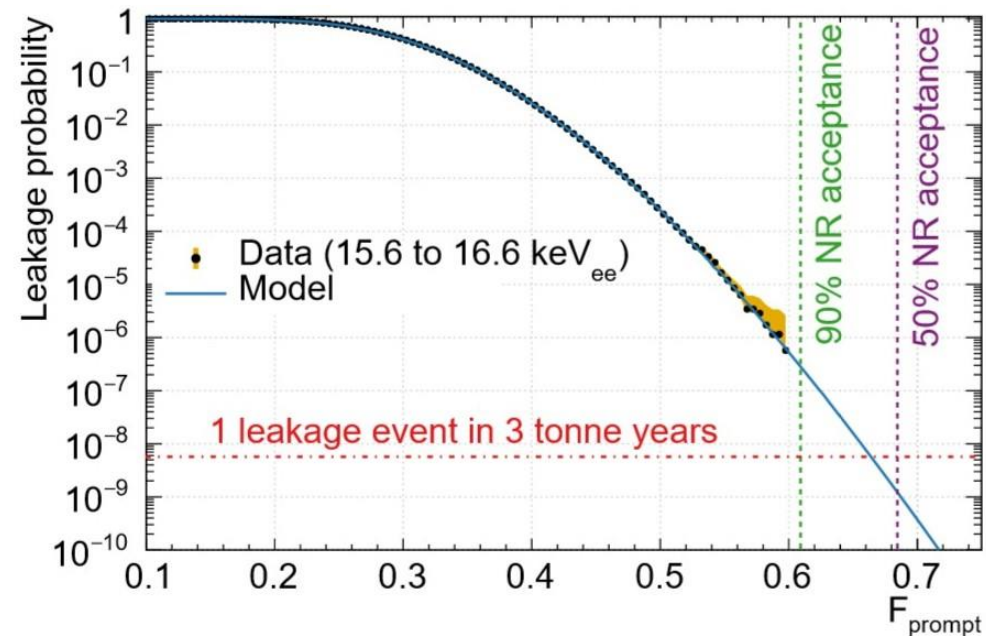


FIG. 13. Probability of an ER being detected above a given F_{prompt} value in the lowest 1 keV_{ee} bin in the WIMP-search region of interest. For comparison, vertical lines show the values above which 90% or 50% of nuclear recoils are expected to be found.



Hundreds of TBytes \rightarrow 0 events in ROI

Data processing effort

Organizing software

Efficient code

Database for calibrations and meta-data

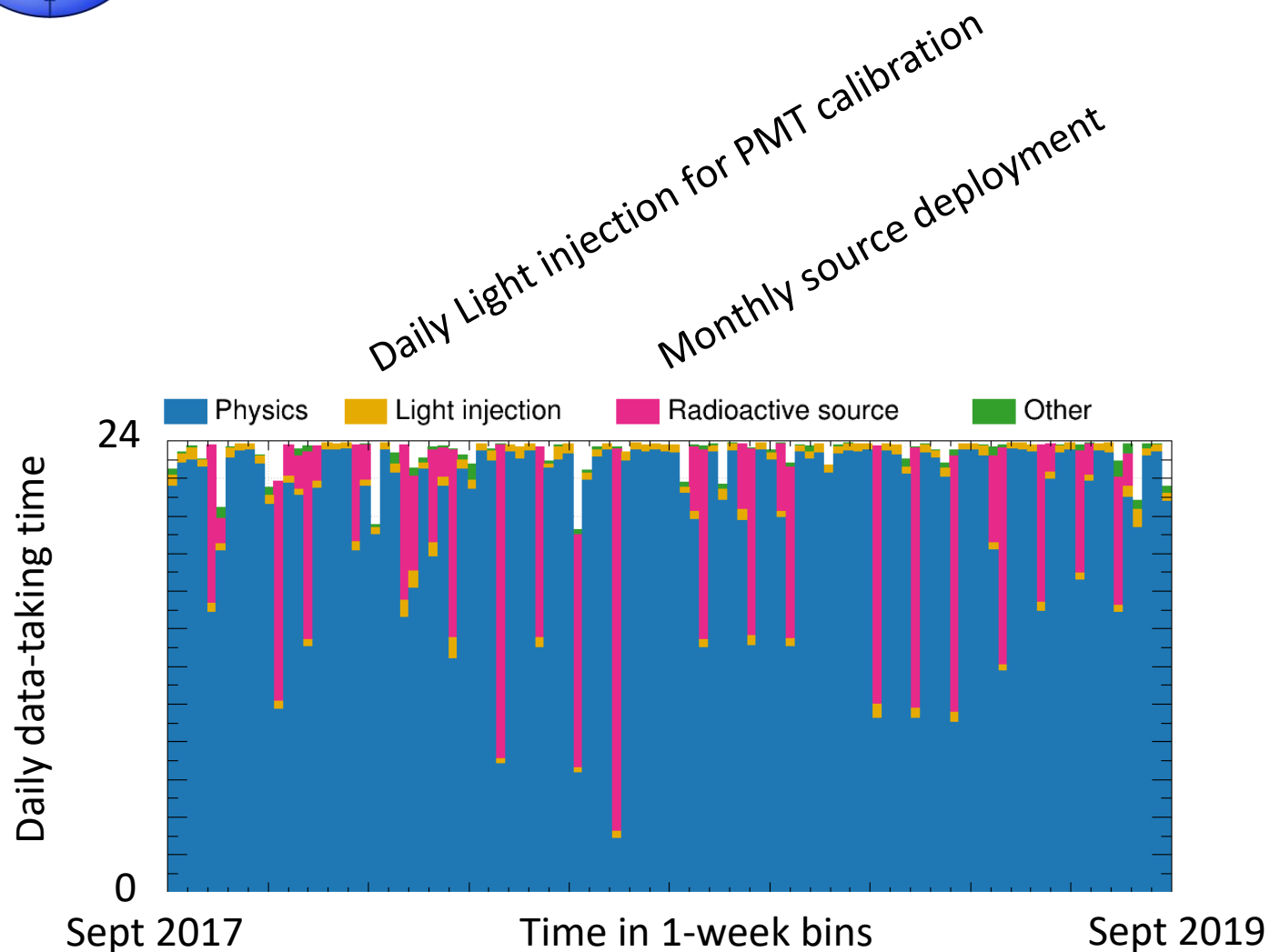
Database for detector parameters (ie thermodynamics of LAr)

Efficiently using large clusters, and managing when they are over-subscribed

Sharing and archiving



A data-taking routine helps analysis

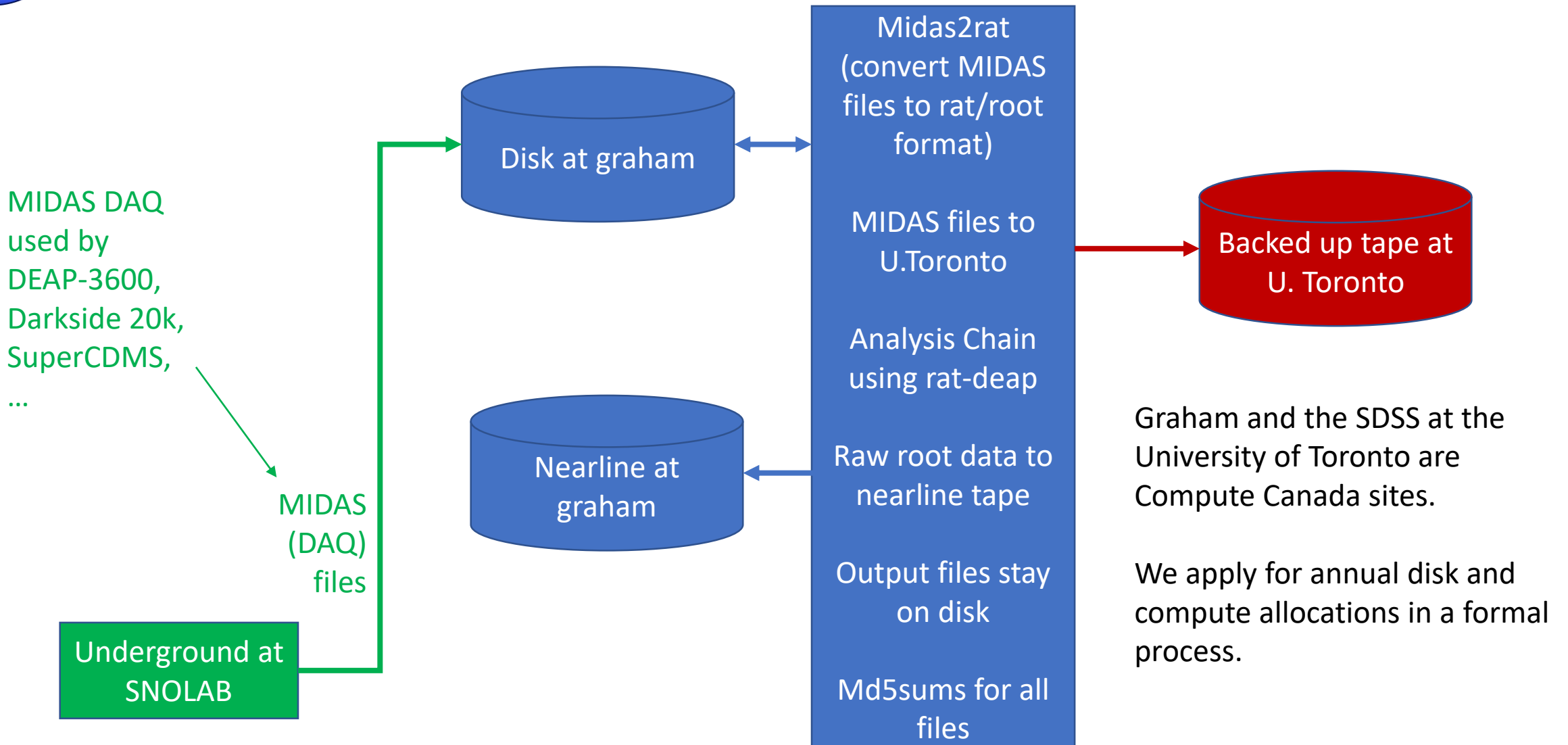


Ideal Chain

1. DAQ-level data-quality checks by shifters
2. Checksums of raw files
3. Nearline analysis of light injection followed by appropriate averaging of PMT calibrations over time.
4. Processing using PMT calibrations:
 1. Low-level
 2. Event reconstruction
 3. Data quality plots made
 4. Blindness
 5. Skimmed and complete output
5. Write raw data to backed-up tape
6. Data quality plots interpreted
7. High-level analysis

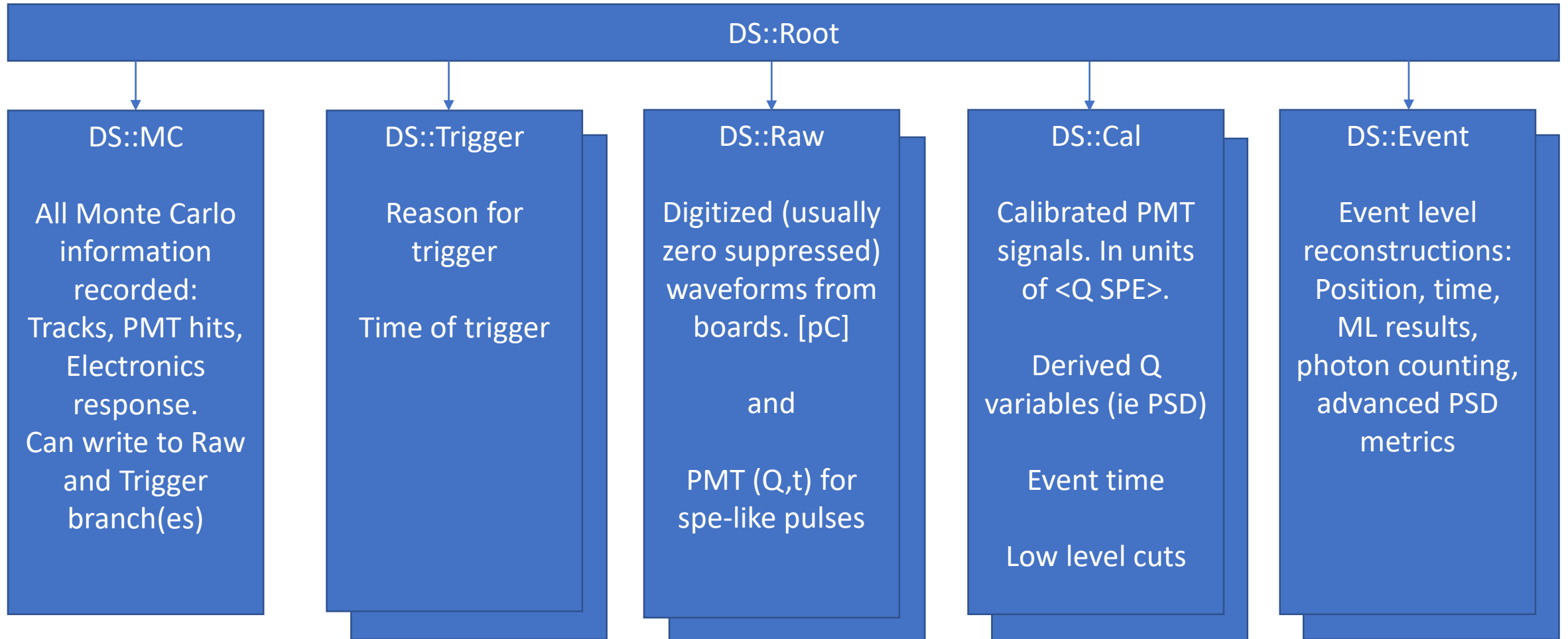


Our dataflow is run by a python program called autoDEAP





rat-deap starts with DAQ or MC info and through a series of processors builds up more sophisticated event information.





Preference for compiled code

- Use compiled code whenever possible. A great python utility that pulls out a couple of variables and writes them to an ntuple sounds great.
- We moved from such a python tool to C++ and sped up our code.
- (We use python to call some libraries example in machine learning, but not (much) in bulk data processing.)



Data available in contact forms so it can sit on small University clusters for high-level analysis

Note that MIDAS seamlessly closes and opens a new file once 2 GBytes is reached.

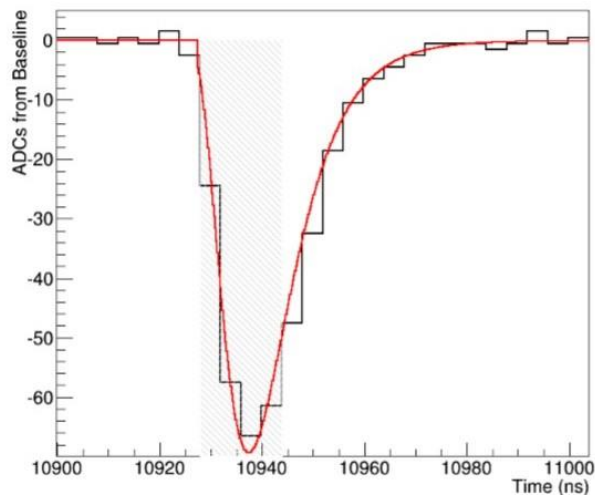
Type	Size	Final Location
MIDAS format raw data	2 GByte / subrun	U Toronto backed up tape
Root format raw data	1.7 Gbyte / subrun	Graham nearline (not backed up)
Root format processed data with raw information and standard reconstructions. All but Ar-39 events.	0.200 GBytes / subrun	Graham (backed up) + other sites
Root format summary ntuples (all events)	0.1 GBytes / subrun	Graham (backed up) + other sites
Root format ntuples (Ar-39 removed)	0.001 Gbytes / subrun	Graham (backed up)



What is the information in a PMT signal?

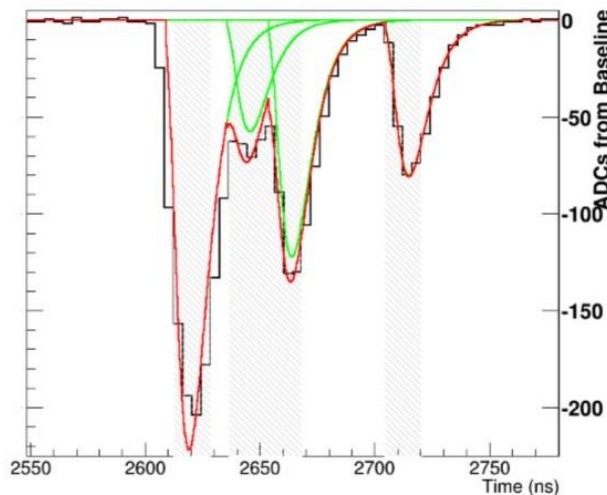
A single SPE pulse

2 numbers: (Q, t)



Multi-peaked pulse

8 numbers: 4 (Q, t)



Complete information can be stored with arrays of (Q, t) pairs.

Convert the easy pulses to (Q, t) in the DAQ.

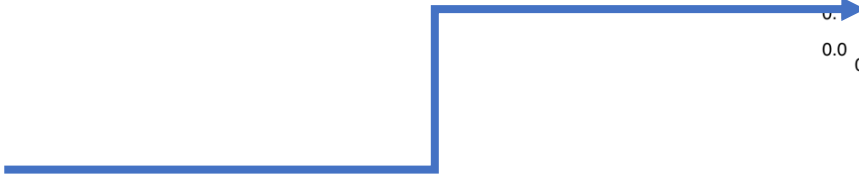
Write the hard ones to disk for offline analysis.

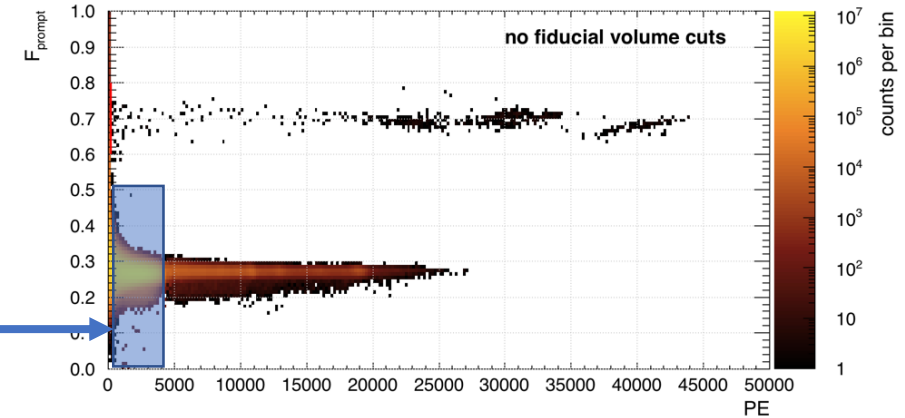
“Easy” means fast analysis with a template fit → About 50% of pulses in DEAP-3600.

Figure 5.3: Examples of pulse fits for a single SPE pulse (left) and a multi-peaked pulse (right). For the multi-peak pulse the individual (dotted) and summed reconstructed pulse shapes (solid) are shown. The shaded regions indicate the voltage samples used in the fits. From the doctoral thesis of T. McElroy, U. of Alberta



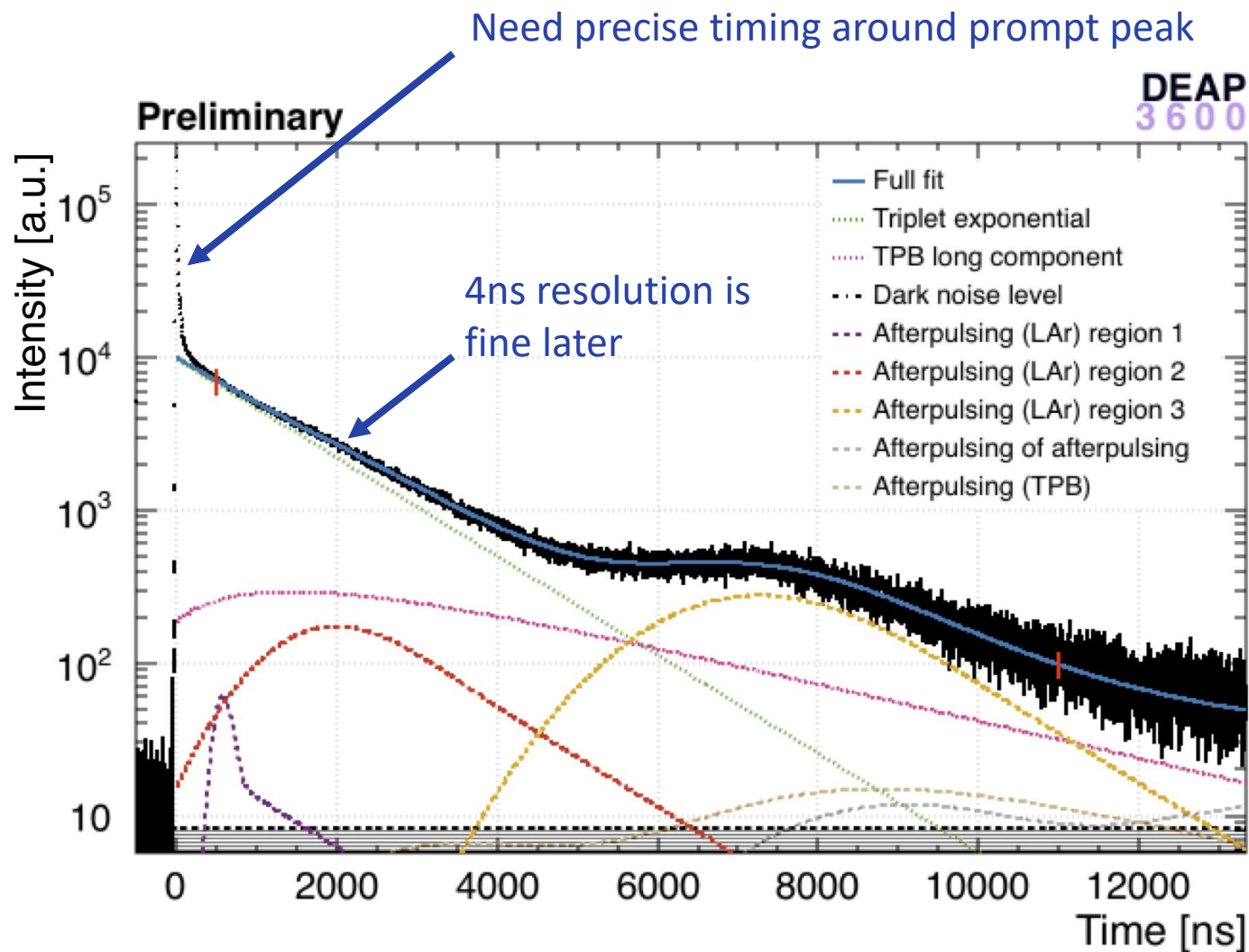
Minimal / Complete Data Structure will let us store more information with fewer bytes

- (Q, t) of each pulse
- Use only as many bits as strictly necessary
- Charge stored in hardware units, pC not nPE.
- Time since last event stored for all events.
- Pre-scale Argon-39. 
- Factor of ~ 20 reduction compared to raw formats with waveforms.
- Small university storage systems can hold all relevant data.





What is the information in the time of PMT signal?



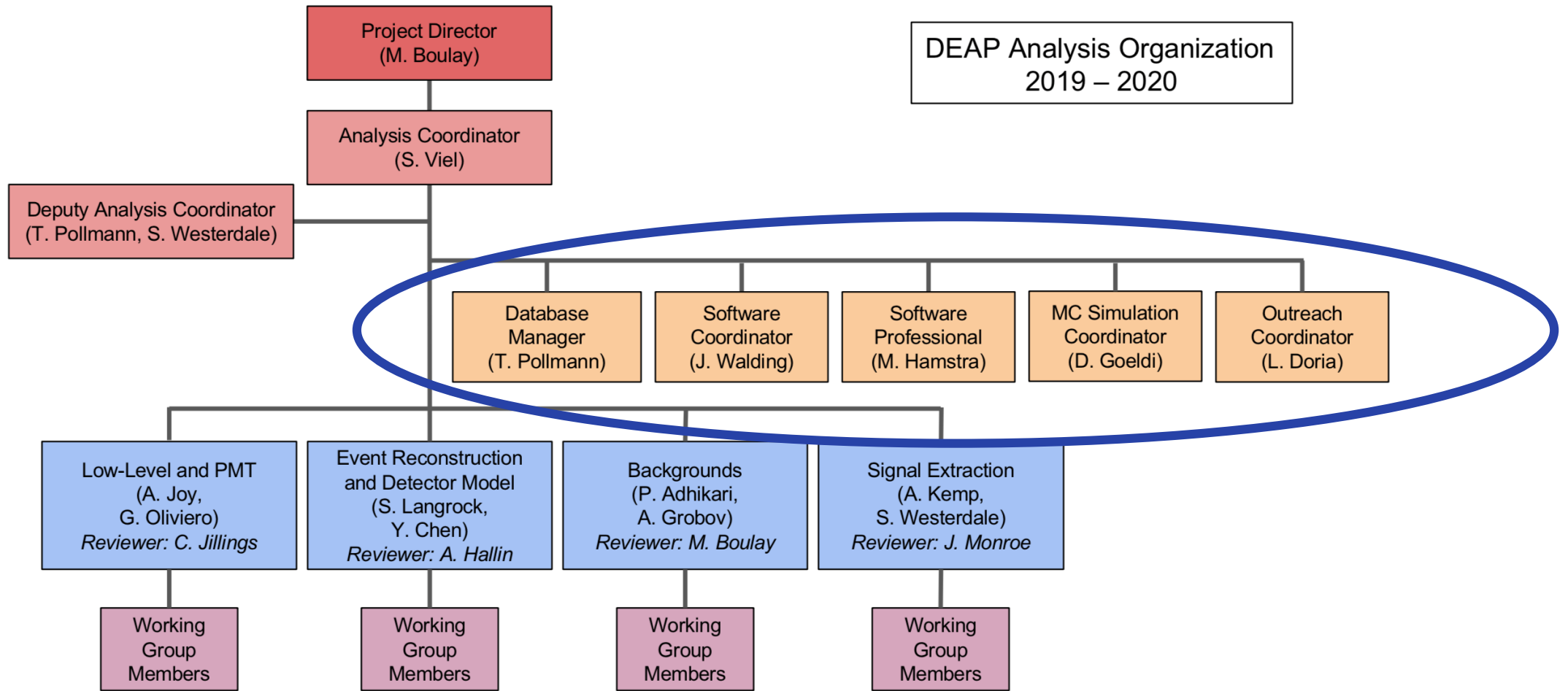
Early in event signal sub nano-second timing is important for reconstruction and PMT response.

Later in the pulse we are washed out by ~ 1400 ns triplet lifetime.

Can code information about pulse times carefully, using fewer bits.



Important parts of software package have technical leads





We use CouchDB with web, C++, python interfaces

Run information Lookup.

Quick links

- Home page
- PMT/DAQ
 - PMT information
 - Veto PMT information
 - SCB channel map
 - PMT docs by RUNID
- Runs
 - Run information
 - Run lists
 - Data quality summary
 - Data quality questions
 - PPG calibration status
 - Trend plots
- Misc
 - AARF information
 - Slow control

• Search by run number

- Specific run:
- Run range: to
- Date range (yyyy-mm-dd): to

This will list all runs between the first and the last 'good' run started between the dates given. Good means data written and HV on and dates are in UTC

• Search by run type ([click here for run type information](#))

- Specific run type:
- Show all runs by runtype:

• List last 50 runs:

Greyed out run numbers mean that no data was written to disk. These can be hidden by clicking the "Hide empty runs" button.
 Orange background means the HV was NOT ON.
 An entry of 'undefined' in the table means that this data is not available, either because the field was added later, or because the run crashed.

[Data quality plots for this run may be found here](#)

[You can edit the data quality information for this run here](#)

```
void PMTInfoUtil::LoadSPECharges(int run_num, bool skipVeto)
{
    if (run_num == -1 || run_num == 100000) {
        LoadSPECharges(2, skipVeto);
        return;
    }
    SPEChargeMap.clear();

    map<std::string, float> SPEMap = DB::Get()->GetFArray("PMTSPE", "avgCharge", run_num);
    debug << "PMTInfoUtil: Reading veto spe data.." << endl;
    map<std::string, float> VetoSPEMap = DB::Get()->GetFArray("VETOPMTSPE", "avgCharge", run_num);

    vector<int> pmtIDs = dc->GetPMTIDs();

    for (unsigned int i = 0; i < pmtIDs.size(); i++) {
        int pmtID = pmtIDs[i];
        std::string pmtIDStr = ::to_string(pmtID);

        try { // see if we read a value for this PMTID from the database
            SPEChargeMap[pmtID] = SPEMap.at(pmtIDStr);
            if (i==0) {
                debug << "SPE Q: Run " << run_num << " PMT " << pmtID << " " << SPEChargeMap[pmtID] << endl;
            }
        } catch (...) {
            Log::Die(Form("PMTInfoUtil: Database does not contain necessary data for SPE calibration of PMT %d at run %d.", pmtID, run_num));
        }
    }
}
```

```
results = db.view('WebView/goodRunsByTimestamp', startkey=UnixBeginWeek, endkey=UnixEndWeek, include_docs=False)

for row in results:

    if row["value"][0] == 123:

        runNumber = row["value"][1]

        #get the RunConfig document for the run in the run range provided
        docID = row["id"]

        if docID in db:
            doc = db[docID]

        else:
            print "Missing runinfo document for run " + str(runNumber) + ". Please contact database admin."

            continue

    #Set the restrains on what runs are picked to be analyzed
    if 'sourceIntensity' in doc and 'subrunCount' in doc and 'runType' in doc and 'sourceID' in doc:
        sourceIntensity = doc['sourceIntensity']
        subrunCount = doc['subrunCount']
        runType = doc['runType']
        sourceID = doc['sourceID']
        dateTimeStart = doc['dateTimeStart']
        eventCount = doc['eventCount']
```

https access

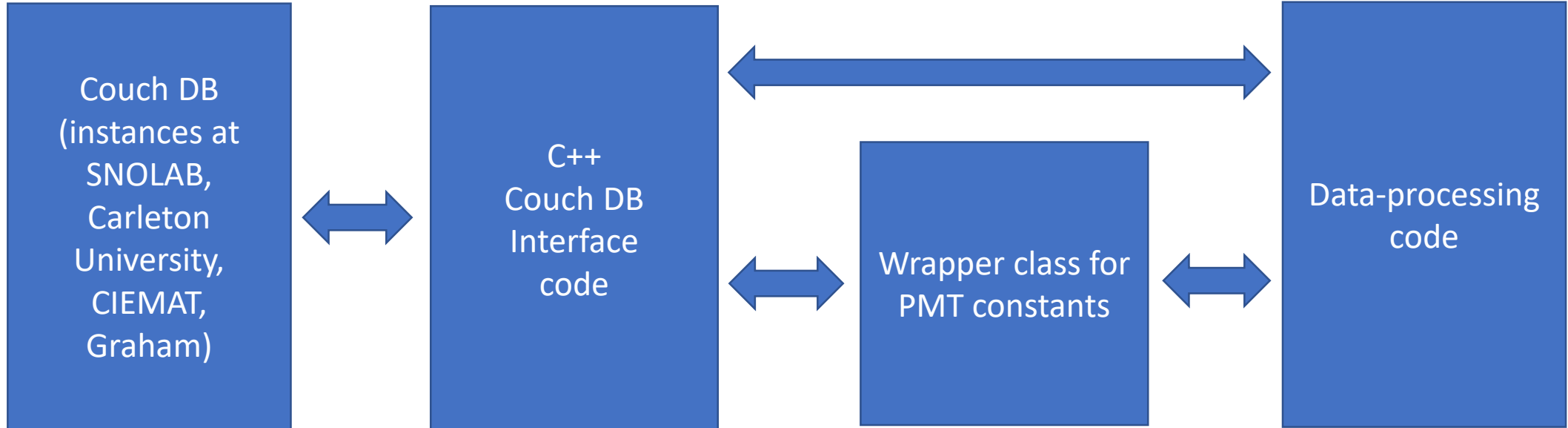


Database skills are different from analysis skills. Maintain and spread DB expertise.

1: Database code has efficient well-defined read tools with local caching.

2: Wrapper classes for DB make it easy for physicists to get parameters for particular purposes.

3: Analyzer high-level calls to get PMT parameters for given runs.



4: Take care with wrapper classes: they can ask the DB to download all the information at once. If a reprocessing or Monte Carlo script starts many jobs at the same time, you can have a huge demand spike causing failures: Retries should come at random times after the first try.



Blindness implementation

We use boxes which will be opened in a defined way.

We use a processor which determines the blindness level (ie open or which box).

The output streams all use this one processor.

Blinded events are written to different files while preserving time-since-last-event information. The files not readable to the general collaboration. Only our service account can read them.

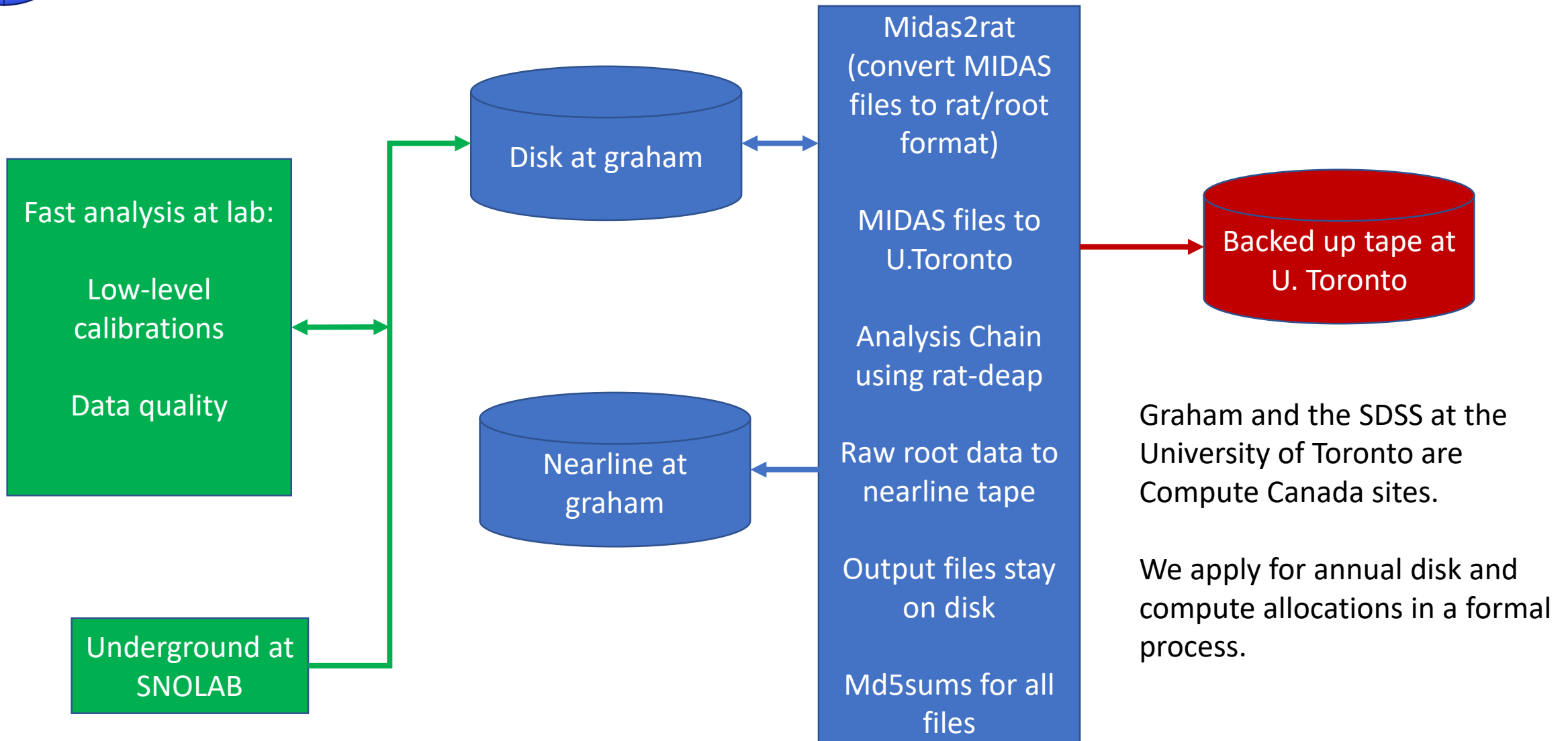


A Local Cluster at the Lab

- A cluster at the lab for low-level calibration and analysis is necessary for efficient analysis.
- Cluster upgrades/outages/usage demands can cause analysis to fall behind. If so, data-quality can fall behind causing loss of live time as un-diagnosed problems continue.
- SNOLAB is building such a nearline cluster now. It will have the Compute Canada software pack but will be administered locally. The goal is to allow experiments guaranteed access to enough process to make nearline checks.



A fast analysis in a cluster at the lab is a help





A fast analysis in a cluster at the lab helps two ways

Fast analysis at lab:
Low-level calibrations
Data quality

Underground at
SNOLAB

1. Large system upgrades/outages/usage demands can cause analysis to fall behind. If so, data-quality can fall behind causing loss of live time as un-diagnosed problems continue.
2. Doing low-level calibrations immediately means a full analysis with proper constants can be done. (Automate with human checks)

SNOLAB is building such a nearline cluster now. It will have the Compute Canada software pack but will be administered locally. The goal is to allow experiments guaranteed access to enough process to make nearline checks.



What if someone outside the collaboration wants to analyze data?

We are having this meeting because analyzing data is hard.

The approach in SNO and DEAP-3600 (and others) is to invite those outsiders to propose formally joining the collaboration.

They would join analysis working groups and thus work with experienced collaborators to avoid rookie mistakes.

They would also do agreed-upon service work.



Long term archiving

- What information matters?
- What information will be meaningful in 20 years?
- Even if you have a host that can run a RedHat 6 VM with the right version of root, will the new analyzer understand what they are analyzing?
- How do you keep your constants and meta data?
- Maybe you keep summary ntuples – with variables in physics units - for data and Monte Carlo?



Needs and Challenges

- Computing resources in Canada are oversubscribed. Re-org and cash infusion should help on time scale of ~3 years.
- Falling behind – automated tools with problem checking
- Understanding the information theory behind you signals: what do you need to keep? Leading to the question:
- How do you package your data setup up as small as possible, but still completely so it can be analyzed?
- Long-term archiving?



Canadian Nuclear Laboratories

Laboratoires Nucléaires Canadiens



Jillings - DANCE workshop - Rice University, Houston - October 27 and 28

