Collider Data Analysis Strategies
Tuesday, October 6 from 2:30pm - 4pm Fermilab time
https://indico.fnal.gov/event/44870/sessions/16380/

Inspiration:
(1) Josh Bendavid: Systematics-limited targets for measurements/searches
(2) Liza Brost: Statistics-limited targets for measurements/searches

Data Acquisition:
(3) Jennifer Ngadiuba: Advances in triggering and data acquisition
(4) Antonio Boveia: Challenges and opportunities for trigger-level analyses

From Reconstruction to Analysis:
(5) Loukas Gouskos: Object identification and event classification
(6) Matthew Feickert: Differentiable programming
(7) David Shih: Challenges and opportunities for anomaly detection

Interfacing beyond Experimental Collaborations:
(8) Simone Marzani: Interface of theory calculations with experimental methods
(9) Ana Trisovic: Data preservation and reinterpretation

=====

Ideas/Comments:

- Enter your thoughts here!
- @Josh: there's a famous problem with PDFs of eating your own tail, i.e. when you use your own data to constrain them you start correlating different analyses that then take them as input. Is attention being paid to how to make sure these effects can be understood?
- @Jennifer, @Antonio: are cases where it would make sense to do analysis in L1 hardware, or is the bandwidth to the software level high enough?
    - There is ongoing work in CMS to do analysis both at HLT (several already published) and at L1 in Run 3 and Run 4 - Javier
    - Reference for ongoing work in the L1 scouting @ CMS: https://indico.cern.ch/event/773049/contributions/3474327/ - Jennifer
    - What sort of analysis benefits from L1?
      Antonio: This can also be found e.g. at the above link, which is also at the end of my talk (soft bombs, RPV SUSY, dark photons…). At ATLAS, even for the prototype 'dijet TLA', L1 is still the main limitation on the dijet resonance searches (and is the reason why the dijet TLA doesn't cover the higgs mass region). Strong production (multi-jet and flavor physics triggers) would quickly saturate the full L1 bandwidth during Run 2&3, and any signals with primarily soft hadrons/photons would be prime candidates (e.g. <<100 GeV scalars decaying to photons or b jets).

- @Jennifer Another exciting aspect of TDAQ development deserving to be highlighted is the dramatic improvement in the quality of 'trigger primitives' for HL-LHC. In particular, CMS's L1 Tracking may end up as the biggest game-changing technology as far as HL-LHC trigger goes -Christian H.
    - Indeed I highlighted/mentioned that in slide 4 and provided a link to the CMS Phase-2 L1 Trigger TDR Interim Document (the final TDR to be public soon) - Jennifer
- @David: what would it mean to have an analysis looking for new physics that is *not* comparing against SM predictions?
    - I agree one must of course compare against SM prediction (the null hypothesis). But I would distinguish between simulation and data here. --David
    - To be a bit clearer, perhaps - we can't switch off BSM in time (so any training on data inherently includes BSM effects). So anomalies have to be determined against some expectation - how do we get this purely from data?
    - That's what I alluded to in my talk, there's no free lunch in the bg estimation. If you want to get it from data you have to assume something about your signal (eg it's localized in some feature such as invariant mass). --David
    - @David: does one need to assume something about the signal or could just be assumptions on the background? For example, as in this work: https://arxiv.org/pdf/1912.12155.pdf
    - In the bump hunt example there are assumptions made on both the signal (it's localized) and the background (it's smooth) --David
    -
- @Simone: could we imagine "differentiable MC" where tuning is fully carried through? (This already exists at some level with PDF, PS weights, can this be extended to more parameters?)

From Ben Nachman:

Q for Josh: why do you say that MC stat uncertainty is not a systematic uncertainty per se? I think it is definitely a systematic uncertainty and is quite important for the reasons you highlight!

- Comment for Matthew: another advantage is that you can exploit accelerators (e.g. for 1910.08606, we had to encode Tensorflow versions of some standard jet substructure observables so now they can be readily computed on GPUs)
    - 100%! We didn't include this in the LOI as we were trying to focus on some of the less explored advantages, but this will for sure be in the Community White Paper. - Matthew
- Another comment for Matthew: Have you thought about other languages? There's quite a bit of effort in TF+Swift and in Julia on differentiable programming.
    - We haven't looked specifically at Swift (as far as I know) and I think people in gradHEP have only poked a bit at Julia. Julia is obviously highly impressive and

probably has a bright future in scientific computing, but it is missing the rich ecosystem of flexible scientific libraries that the extended Python ecosystem has now --- and we want to do things _now_ and not engineer a whole lot of stuff and maybe have things in 3 years. We'd love to be proved wrong on this, so please share your thoughts in gradHEP if you have them. - Matthew

        - I will say that Miles Cranmer (Astrophyics, ML) has been doing some very cool stuff with Python AND Julia in PySR: "Symbolic regression built on Julia, and interfaced by Python. Uses regularized evolution, simulated annealing, and gradient-free optimization." - Matthew
    - Michael Kagan notes that there is a Snowmass LOI on "Differentiable Simulators for HEP"
- Follow up on the previous answer by Mattew: what is missing in Julia and what quality we're looking for when we want to deploy diff. Programming in context of HEP?
    - We might just not be as familiar with that community as the SciPy stack is so mature in giving a full stack to work in. Can you join gradHEP and direct people to the equivalent ecosystem/stack of NumPy, SciPy, JAX/PyTorch/TensorFlow (at a minimum) that gives the same range of support (as well as fully differentiable statistical distributions, but that's more a personal interest)? We're looking for fully mature ecosystems that have stable APIs --- JAX doesn't really meet this of course, but I'm making an exception for it given how flexible and performant it is.
    - I think that's all there. I don't think you'll find a truly stable API when you look for differentiable programming, though. I'm following this up on the gradHEP MatterMost -- Jan

====

Some themes from the discussions (from random scattered notes, not exhaustive, perhaps not correct even)

- Important to have a diversity of analysis approaches, and worth thinking about mechanisms to encourage that. Do experimental collaboration policies get in the way of creativity? Do experimental collaboration policies motivate openness and verifiability?
- The feedback loop to improve simulations is very important but very slow. Should enumerate the things we need MC to get right and then get it right. Still have to have mechanisms to deal with mismodeling.
- How do we ensure cross pollination, e.g. in theory studies when techniques from one area can influence other area.
- Move to open data enforces discipline to release information
- Worth looping to other fields to understand their evolution in MC generators
- Worth considering probabilistic approach to object assignment
- Our belief systems and mindsets are encoded in our trigger menus, worth revisiting
- Creating policies within small teams early in a collaboration is easier than retrofitting later.

- Data driven versus simulation drive background estimation?
- Computing can be thought of as a constraint on future analysis.
- Worry about self correlation in tuning/fitting
- Benefit of improved triggers is for purity of events, still constrained by bandwidth