

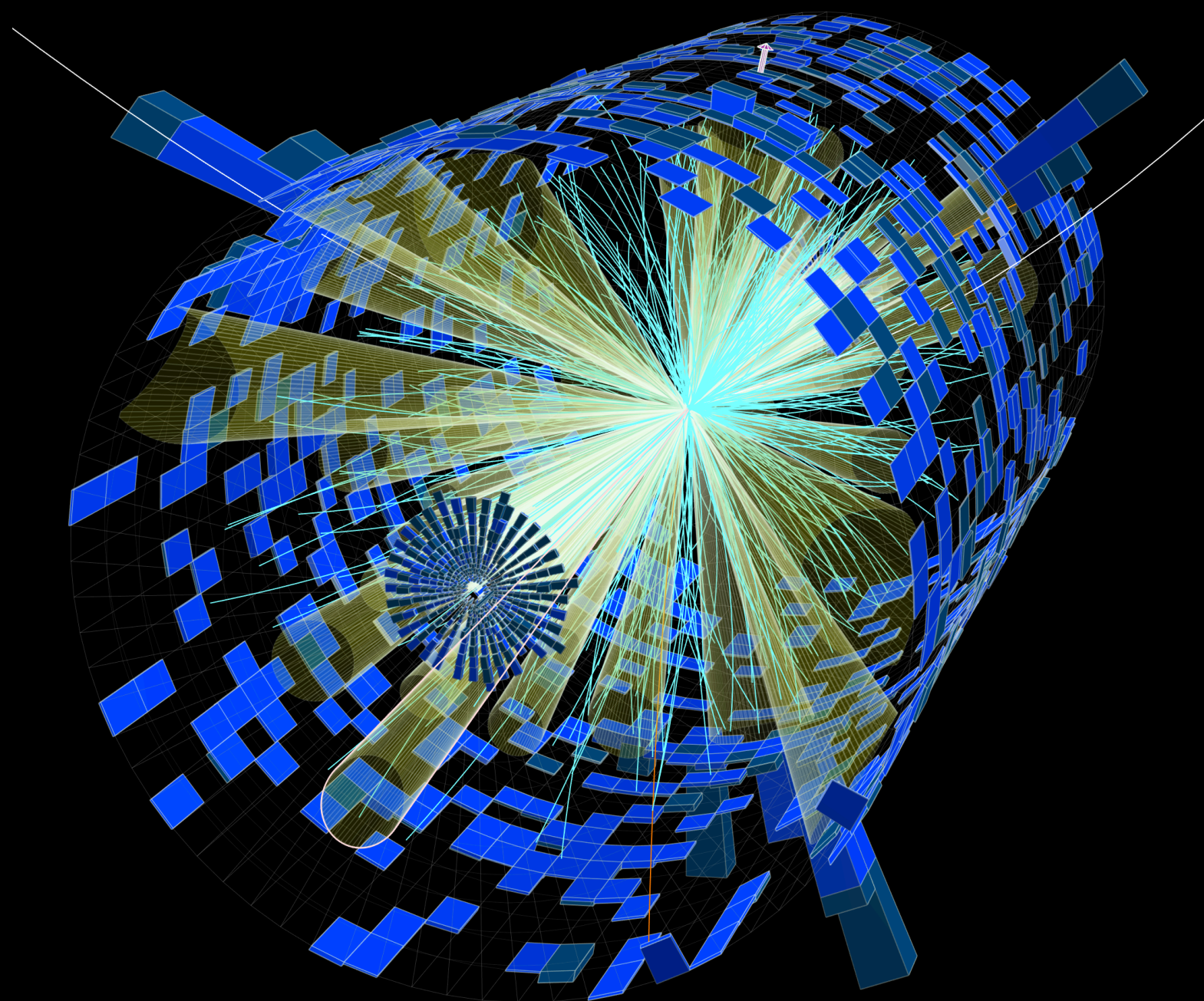


NYU CENTER
FOR DATA
SCIENCE

CENTER FOR
COSMOLOGY AND
PARTICLE PHYSICS



@KyleCranmer
New York University
Department of Physics
Center for Data Science
CILVR Lab



CURRENT STATUS

Lots of statistics tools popping up outside of the ROOT ecosystem (eg. pythonic)

- Would like to have interoperability between them and existing RooFit style
 - Common APIs to enable combinations between models implemented in different frameworks
 - Common specification to build same model in different frameworks
- Align with the APIs used in pythonic and data science frameworks
 - eg. tensorflow.distributions, pytorch.distributions, ...

Would like to integrate efforts like Coffea and downstream stats tools being developed — **next breakout**

STAT TOOLS IN ROOT ECOSYSTEM

RooFitCore

- defines abstract interfaces for PDFs, real-valued functions, datasets, etc.
- Implementations of operator type classes: simultaneous PDF, additions, products, function->Pdf converter
- Implementation of RooWorkspace (container for objects, also provides a factory language)

RooFit

- defines actual distributions: Gaussians, etc.

RooStats

- Interface and implementation of ModelConfig that specifies what is needed for stats tools to work. Eg. Pointer to top-level PDF, observables in the data, parameters of interest, nuisance parameters, “global observables” for systematic constraints, dictionary of parameter values snapshots
- Interfaces for statistical tools (eg. Confidence intervals) based around ModelConfig
- Implementation of statistical tools (eg. asymptotics, confidence intervals with toys and profile likelihood ratio, etc.)

HistFactory

- A few new functions, pdf types, operator classes
- Specification for HistFactory PDF and tool to parse and build them

HistFitter / TRexFitter / Combine etc.

- Tools that integrate tools above (eg. Select events, build model, do stats, make plots, etc.)

NEWISH EFFORTS (DISCUSSED IN #SATISFACTORY)

Common API - **pyfitcore**

- Inspiration / analog: pyhf has **tensor** library that defines a common interface used inside pyhf code. Different tensor backends (numpy, tensorflow, pytorch, mxnet, ...) implement thin adapters to satisfy the interface
- Do something similar for new tools in pythonic ecosystem. For now call it **pyfitcore**
 - Use RooFit's RooAbsPdf, RooAbsReal, RooDataSet, and ModelConfig as guides
 - Revisit a few choices
- Lukas prototyped as Simultaneous

Declarative specification - Satisfactory

- Specification like the RooWorkspace's factory mini-language (for Gaussians, etc.) that different frameworks can interpret and implement.
 - Some issues like normalization for PDF with custom restriction on domains
 - Considerations like reparametrization of observable vs. parameters
- Agree to satisfy pyfitcore API above
- Intermediate Function Representation? Examples that optimize based on

Nail down PDF and ModelConfig API

- PDF: evaluate likelihood, generate random variable.
- Align with tensorflow probability and pytorch.probability
- Maintain reparametrization

Nail down ModelConfig API

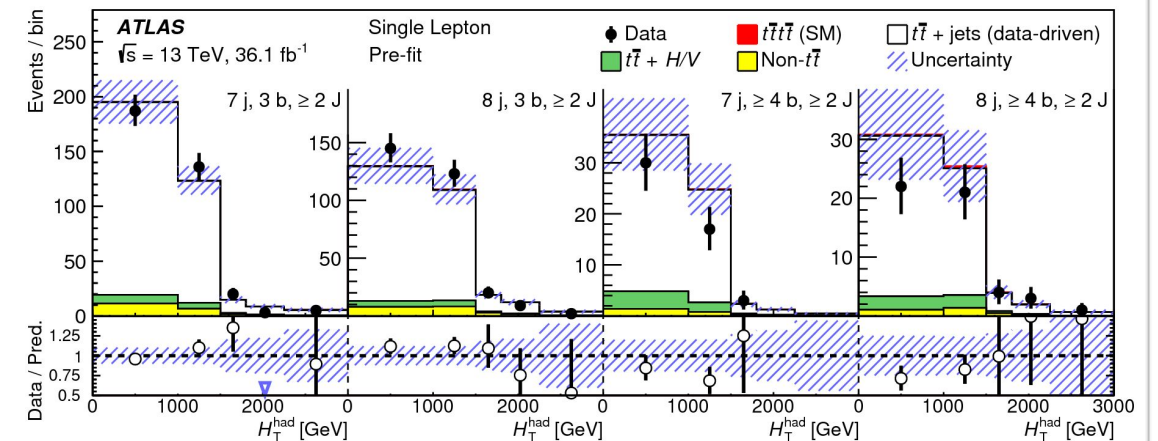
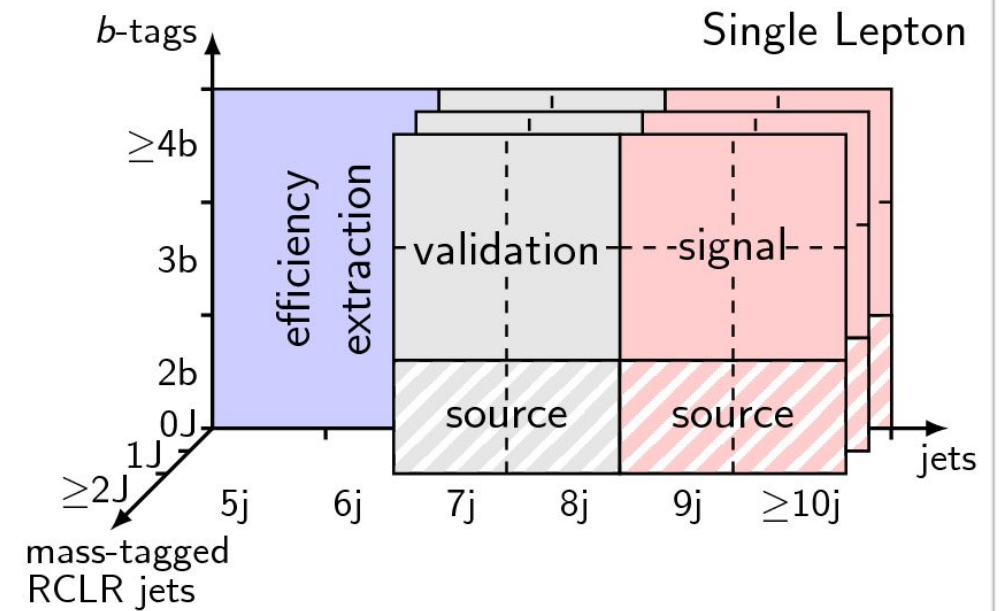
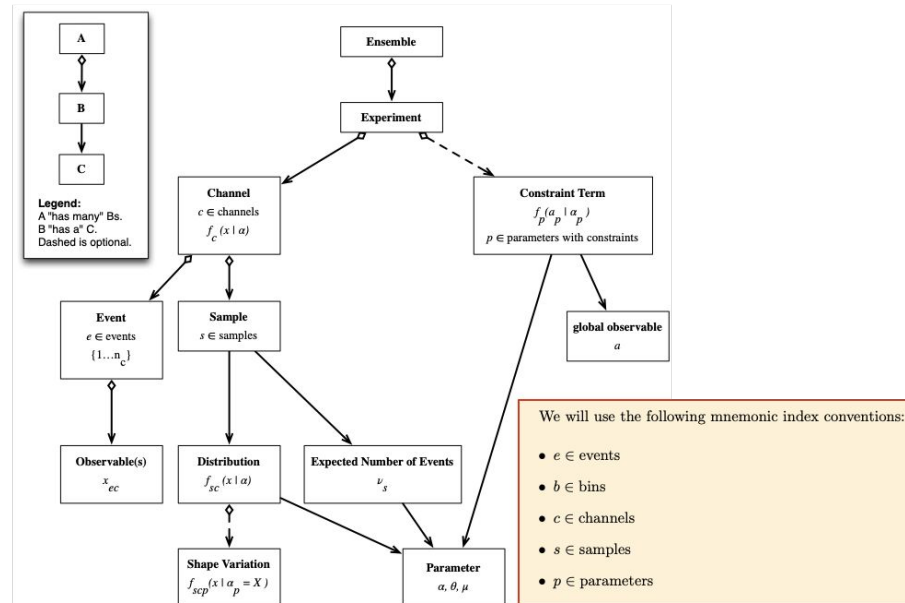
- PDF, POI, nuisance parameters, observables, global observables, parameter snapshots, mapping needed for simultaneous PDFs
- Revisit “global observables”

Part 2

A point of convergence

Several aspects of Analysis Systems converge in a typical physics plot:

- Specification of signal / validation / control regions
- Specification of variables to be used for stat analysis
- Reduction to that format running on data and MC
- Management of MC samples, data driven backgrounds, etc.
- Management of systematic variations
- Feed reduced data (eg. histograms) into specification for statistical model / likelihood function
- Fitting & statistical tools
- Publishing results & derived data products
- Analysis preservation & gateways targeting reinterpretation



FUTURE PLANS

Integrate Coffea -> pyhf a la HistFitter / TRexFitter

- Use the event / column languages to define
 - Variable definitions used to define the observable and variables used for event selection
 - Predicate used for event selection that defines the “channel”
- Alex Held will be working on this
- Some initial prototyping: <https://github.com/lukasheinrich/pyhfinput>

Going beyond HistFitter and TRexFitter

- Inferno-style
 - replace static observable with differentiable computational graph to compute it
- MadMiner / GPs etc.
 - Replace histogram representation of distribution with ML-based density estimation



Template fits: TRExFitter et al.

Fitter

Alexander Held¹²³

¹ University of British Columbia

² TRIUMF

³ starting at New York University soon!

June 19, 2019