Enough `RooFit` to be dangerous

Sam Cunliffe

Pacific Northwest National Laboratory

26th B2GM. Hands-on analysis and software tutorial. Afternoon session.

11 February 2017

PNNL-SA-123596

Prerequisites

- **Required**:
  - Local install of ROOT with `cmake -Droofit=ON` (probably default these days)
- Optional:
  - pyROOT: `-Dpython=ON` (is default these days)
  - `jupyter-notebook` with ipython kernel
    - exactly same thing as `ipython notebook`
    - only needed if you want to follow along exactly with me

OR

- Just work at KEKCC

OR

- Work somewhere with `/cvmfs/belle.cern.ch` mounted.

Preamble

- I'm fairly agnostic to setup / programming languages / tools.
  - …should work in both ipython and with "vanilla" pyROOT python script
  - …or you can write a macro for interactive root session (CLING)
  - …or a compiled C++ executable (`-lRooFit` and `-lRooFitCore`)

- I try to be pedagogical and start from the total basics.
- Apologies in advance if patronising.
  - Intended for PhD students and new postdocs who don't come from LHC experiments. And/or new, eager `RooFit` users.
  - If you find this a little slow-going. Feel free to skip ahead.

- Rigorous statistics outside scope of this tutorial – but ask if unclear.

Answers

- … I will post answers to the confluence page after this session [https://confluence.desy.de/display/BI/Physics+HandsOnAnalysisTutorialFebruary2017]

- And push my notebooks (with answers) to [https://stash.desy.de/users/scunliff/repos/b2gm-roofit-tutorial-feb2017/browse] it currently contains a copy of these slides and a very small `release-00-08` data file

```
git clone ssh://git@stash.desy.de:7999/~scunliff/b2gm-roofit-tutorial-feb2017.git
```

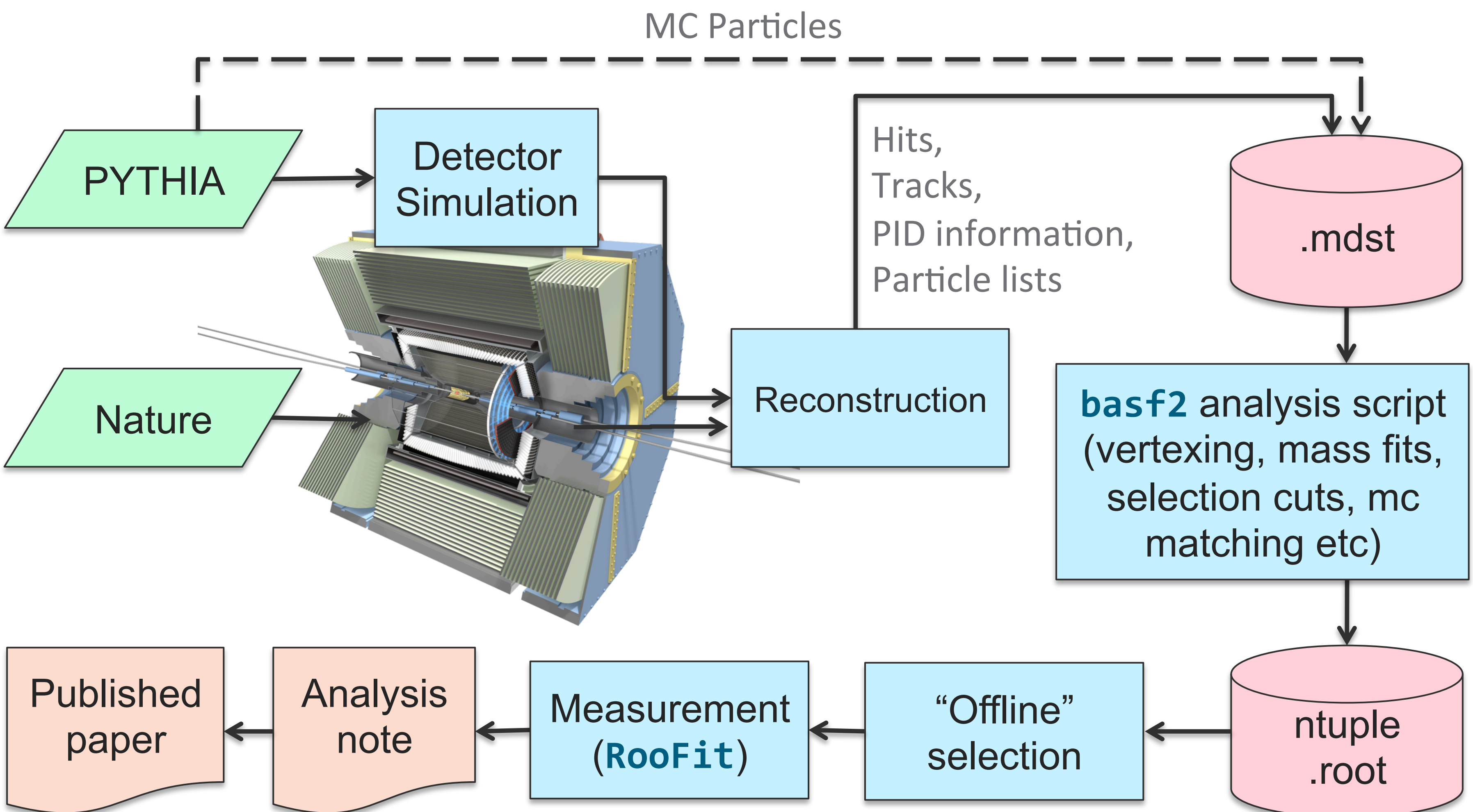
Introduction

- Thanks to previous tutorials we have some reconstructed data (actually it's simulation) in a flat ntuple…
  - Specifically a `TTree` stored in a `TFile`
  - Even if you don't have that you know how to use `basf2` to go and get some.

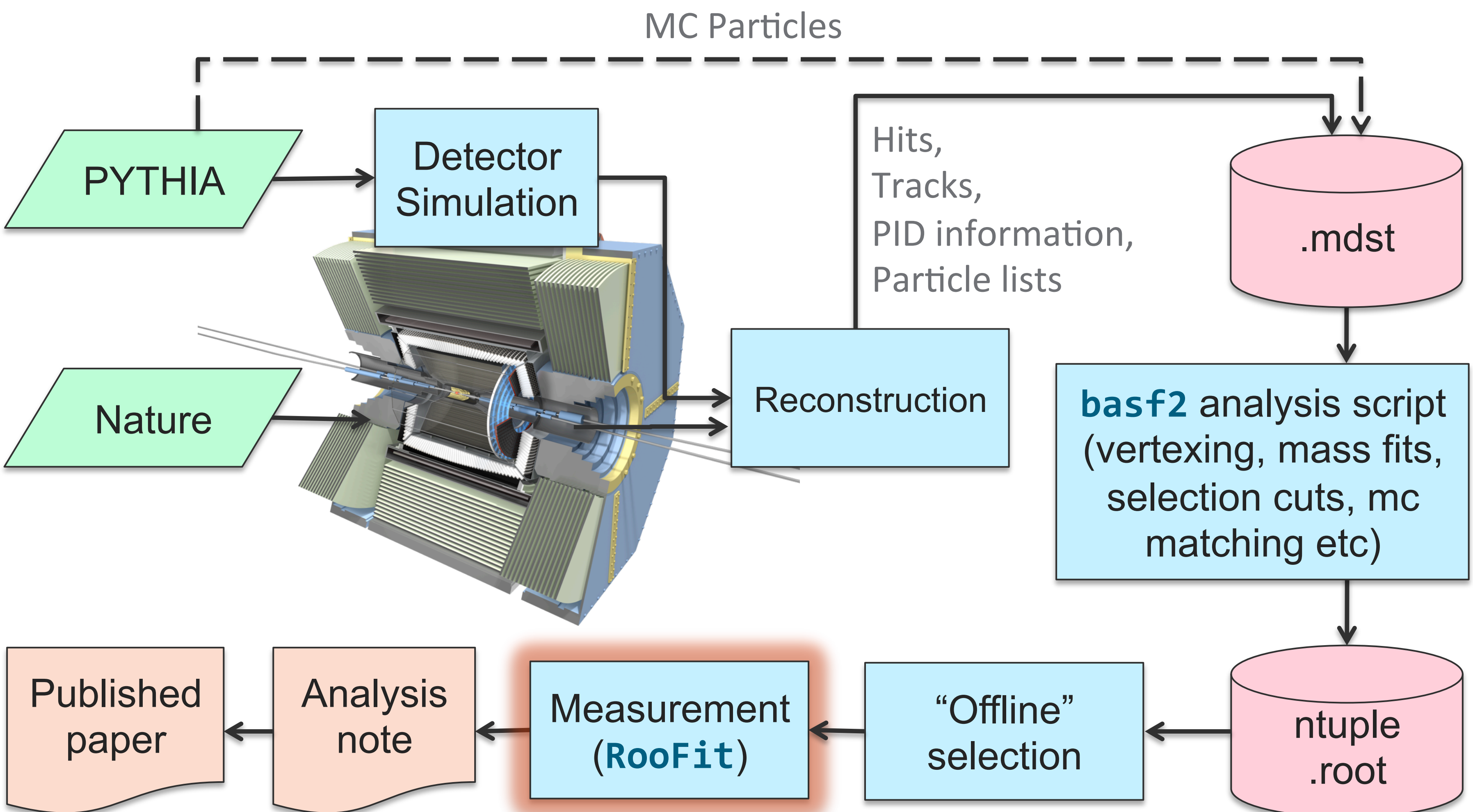
- I assume you've figured out a really cool selection.
  - "Just" a signal clean-up problem.

- Now let's make a measurement of something… the fun bit.

Belle II flow chart



Belle II flow chart



Measure what?

- (Most of the time) want to know **how many events**..

$$\mathcal{B}(B \to f) \propto N(B \to f) \qquad A_{\mathrm{CP}} = \frac{\mathcal{B}(B \to f) - \mathcal{B}(\overline{B} \to \overline{f})}{\mathcal{B}(B \to f) + \mathcal{B}(\overline{B} \to \overline{f})}$$

  - OK sometimes it's "as a function of time" or "as a function of angle"

- Two main approaches:
  1. Define a signal "box" or "window" in one or more variable
     a.k.a "cut-and-count" "look in the box" methods
  2. Perform a fit and extract a yield / parameters of a model…

- Advantage of method #2 is that we can also measure other things…

Pre model-building: some definitions

- Observable quantities / **independent variables** / data in an ntuple

$$m_{\mathrm{BC}},\ E,\ \Delta E,\ x,\ y,\ z,\ \theta,\ t \longrightarrow \vec{x}$$

  - One 'coordinate' provided by each datum (each event, or photon or whatever)
  - Typically have an associated physical range (or can be assigned one)

- **Parameters** / measurable quantities (sometimes known sometimes not)

$$m_{B^0},\ \Gamma,\ n_{\mathrm{events}},\ \mathcal{B},\ \Delta m^2 \longrightarrow \vec{p}$$

  - Given by nature but not measured directly by our detector for each event

Pre model-building: more definitions

- Probability distribution function: ‘is’ the model

$$g(\vec{x};\ \vec{p}\,)$$

  - Everyone in the universe calls it a “**pdf**”.
  - (If the model builder has done a good job) It describes the probability of observing a datum with ‘coordinates’ $\vec{x}$ given the set of parameters $\vec{p}$
  - You get to choose the functional form based on previous experimental work / your supervisor’s advice / some physics reason.
  - It is normalised:

$$\int_{\vec{x}_{\mathrm{min}}}^{\vec{x}_{\mathrm{max}}} g(\vec{x};\ \vec{p}\,)\mathrm{d}\vec{x} \equiv 1$$

Likelihood function

- A single datum is a set of coordinates $\vec{x}$
  (e.g. a point in some 2D plane, a single point in an energy spectrum…)
- A dataset is a set of these coordinates $\{\vec{x}_i\}$ (let's label them $i \in [0..N]$)

- The likelihood is the **joint pdf**. I.e. the "pdf" for $\vec{p}$ given the data and g.
  - Don't call it a pdf, that's confusing. Call it the likelihood.
  - Construct the likelihood by evaluating the pdf for each data point and multiplying those numbers:

$$\mathcal{L}(\{\vec{x}_i\}, \vec{p}) = \prod_{i=0}^{N} g(\vec{x}_i;\, \vec{p})$$

  - At a maximum for $\vec{p}$ most consistent with the data

Likelihood function

- A single datum is a set of coordinates $\vec{x}$
- A dataset is a set of these coordinates $\{\vec{x}_i\}$

- The likelihood is the **joint pdf**.
  - Construct the likelihood by evaluating the pdf for each data point and multiplying those numbers:

$$\mathcal{L}(\{\vec{x}_i\}, \vec{p}) = \prod_{i=0}^{N} g(\vec{x}_i;\, \vec{p})$$

  - At a maximum for $\vec{p}$ most consistent with the data

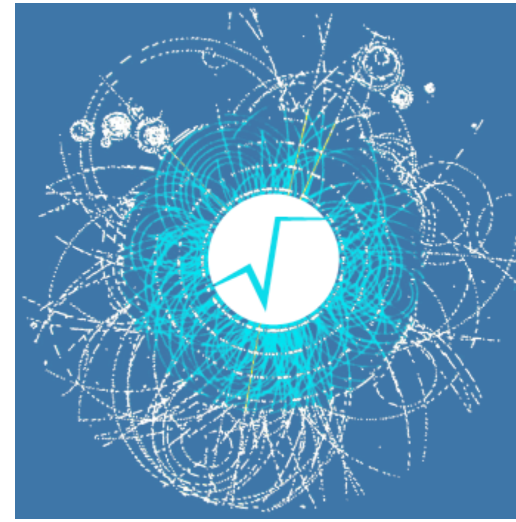
- Negative log likelihood:

$$\boxed{\mathrm{NLL} \equiv -\log\left(\mathcal{L}(\vec{p})\right) = -\sum_{i=0}^{N} \log\left(g(\vec{x}_i;\, \vec{p})\right)}$$

Fun questions

- Why work with NLL rather than likelihood itself?

What is RooFit ?



- A library of C++ objects and data structures.
- A nice* way to interface with MINUIT
  - specifically targeted at minimising NLLs, perform fits and estimate/measure parameters.
- Part of ROOT. “Just works” with ROOT files.

- It’s not the only way to do fitting…

- …**<u>but</u>** it is the way that your supervisor / senior postdoc is probably going to be most familiar with.
- And it is very powerful.

`RooFit` land

- Everything is a `RooFit` object starts with “Roo” instead of “T”
- No distinction between independent variable and parameter (you make the distinction in what data you provide)
- Bazillions of functions are already implemented as pdfs
  - If your favourite function is not implemented it’s very easy to make your own

RooFit land

independent variable	x	`RooRealVar`
parameter	p	`RooRealVar`
function	$f(x)$	`RooAbsReal`
pdf	$g(\vec{x};\vec{p})$	`RooAbsPDF`
integral	$\int f(x)\mathrm{d}x$	`RooRealIntegral`
single datum	$\vec{x}_i$	`RooArgSet`
unbinned data set	$\{\vec{x}_i\}$	`RooDataSet`
histogram of data		`RooDataHist`
NLL	$-\log(\mathcal{L})$	`RooNLL`
plot		`RooPlot`
range, signal window		`RooFit::Range( , )`
fit result		`RooFitResult`
set of parameters	$\vec{p}$	`RooArgSet`
ordered list		`RooArgList`

Fun questions

- Why work with NLL rather than likelihood itself?
- When was `MINUIT` written?
  - In what language?
  - What is `Minuit2`?
- When was ROOT’s original release?
- When was `RooFit`’s original release?

Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

1a-Fit-mBC.ipynb
1b-Fit-mBC.ipynb

Interlude - `MINUIT`

For our purposes MINUIT has 3 algorithms… (read the stdout!)

- **MIGRAD** – a minimiser.
  Performs an approximate error calculation and minimises assuming a smooth and well behaved (parabolic) likelihood.

- **HESSE** – an error matrix calculator.
  Calculates a 'better' error matrix using second derivatives. You might want to call it before minimising, and definitely afterwards.

- **MINOS** – calculates the correct errors in all cases.
  Even badly behaved (non-parabolic) likelihood. Can produce asymmetric errors. Computationally more expensive.

Another interlude Extended likelihood function

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by **Battelle** *Since 1965*

- In cases where you want to float normalisation of a two-component fit.
- Can add an extra degree of freedom to the pdf (extra normalisation)
- Can add a term to the likelihood (representing a constraint)

$$\mathrm{NLL} \equiv -\log\left(\mathcal{L}(\vec{p})\right) = -\sum_{i=0}^{N} \log\left(g(\vec{x}_i;\, \vec{p})\right) \boxed{+ N_{ex} - N_{ob}\log(N_{ex})}$$

$$N_{ex} = n_s + n_b$$

- In fact if you build a `RooAddPdf` with two yields, `RooFit` expects this

Fun questions

- Why work with NLL rather than likelihood itself?
- When was `MINUIT` written?
  - In what language?
  - What is `Minuit2`?
- When was ROOT’s original release?
- When was `RooFit`’s original release?
- Where does the Crystal Ball function come from?
  - Can you find the ‘standard’ reference?

2-2DFit.ipynb

Fun questions

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by **Battelle** *Since 1965*

- Why work with NLL rather than likelihood itself?
- When was `MINUIT` written?
  - In what language?
  - What is `Minuit2`?
- When was ROOT's original release?
- When was `RooFit`'s original release?
- Where does the Crystal Ball function come from?
  - Can you find the 'standard' reference?
- Why would you ever want a `RooProdPdf` as there are no cross terms?

3-Advanced-NLL-etc.ipynb

Fun questions

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

- Why work with NLL rather than likelihood itself?
- When was `MINUIT` written?
  - In what language?
  - What is `Minuit2`?
- When was ROOT's original release?
- When was `RooFit`'s original release?
- Where does the Crystal Ball function come from?
  - Can you find the 'standard' reference?
- Why would you ever want a `RooProdPdf` as there are no cross terms?
- What is the difference between projecting the likelihood and marginalising the likelihood?
  - Which one did we do in exercise 3?

Other references and resources

- software@belle2.org (is there a plan to make a stats@belle2.org ?)
- https://confluence.desy.de/display/BI/New+Physics+and+Statistics
- https://root.cern.ch/roofit-20-minutes

- ROOT/RooFit sub forum
- Google "RooFit"
  + thing you want to do
  - You are not alone!

- RooFit users manual
  - Old (2008)
  - …but comprehensive