8.882 LHC Physics

Experimental Methods and Measurements

Analysis Tips

[Lecture 9, March 4, 2009]

Physics
Colloquium Series

'09
Spring

The Physics Colloquium Series

Thursday, March 5 at ***4:15 pm*** *in room* ***10-250***

Marc Kamionkowski

California Institute of Technology

"Is the Universe Homogeneous and Isotropic? "

For a full listing of this semester's colloquia, please visit our website at **web.mit.edu/physics**

Organizational Issues

Due days for the documented analyses

- due date for 'Charge Multiplicity' is March 12
- work together and hand in one co-produced assignment
- Latex stub has been put onto our course page

Lecture Outline

Coding examples

- how to implement a yield study, as an example of design
- application to NTrks analysis?

Match Monte Carlo truth with reconstruction

- simple kinematic match
- full blown hit matching

Some Data Monte Carlo comparisons ....

What would I expect as hand-in?

Implementation of a Yield Study

Yield study*

- determine number of candidates per lumi and per run
- can be used in many different ways
  - determine whether the yield of some background per run/luminosity is constant
  - implement simple side band subtraction and determine whether signal yield is constant per run/lumi
  - identify potentially incomplete runs *etc.*
- likely candidate to be used in various analyses (generic)
- perfect opportunity to spend some time and decompose the problem

Describing design process in the following slides!

* taken from BottomMods/tools area

Yield Study Design

Step zero – worth to spent time?

- consider: is code likely to be used in various places?
  - will be used all over the place
  - perfect example for reducing complexity and avoid repetition
- is it generic and what does it depend on?
  - yield study can be used for any type of selection
  - study is a bunch of yields
  - yield contains: runNumber, nEvts, intLumi
  - looks like it is completely independent: no root, no BStntuple *etc.*
- one of those rare cases

Yield Study Design

Step one – what should the user see?

- determine typical application in more detail
  - per run: entry with run number and luminosity → BeginRun()
  - per event: increment count depending on selection → Event(int iE)
  - at the job end: show summary, per run, maybe show graphically → EndJob()
- determine the API (Application Programmer Interface)
  - YieldStudy(const char* name)
  - void SetRunNumber(int RunNumber, float Lumi)
  - void Increment()
  - void Print(int PrintLevel = 0)
  - int FillArrays(float *Idx, float *Yields, float *Errors)
  - [work with several yield studies: void Add(YieldStudy& rhs)]
  - minimize what user sees, without affecting effectiveness of tools

Yield Study Design

Step two – further decomposition useful?

- analyze pieces and judge
  - basically we have a list
  - list entries are somewhat more complex: yields
  - implement yield class
- follow steps for each component until everything layed out

Yield class

- float Value() return nEvt/lumi
- int RunNumber(); void SetRunNumber(int RunN)
- int NEvents(); void SetNEvents(int NEvt)
- float Lumi(); void SetLumi(float Lumi)
- void Print()
- void IncNEvents()

Yield Study Design

Details programmer can hide from the user

- run numbers are not contiguous
  - implementation needs to keep separate account of each run number that appeared and only create new entry if number has not yet appeared
  - luminosity is stored each time for the entire run (no adding needed)
- uncertainties can be intrinsically handled on the basis of Gaussian or Poissonian statistics (we use Gaussian)
- implementation of bunch of yields could be: an array, list, vector or map, whatever is most appropriate
- error handling for invalid luminosities of event numbers
- print level can be tuned to ones needs

Possible extensions: combine YieldStudies +/--

Yield Study: An Implementation

```
Ana/TUpsi.hh

YieldStudy fYields;

src/TUpsi.cc

TUpsi::TUpsi(const char* name, const char* title) :
  TStnModule (name,title), ....
  fYields    (name)          // Yield study
{ ....

int TUpsi::BeginRun()
{
  // Get a pointer to the Database Manager and access run summary
  TStnDBManager   *dbm = TStnDBManager::Instance();
  TStnRunSummary *rs   = (TStnRunSummary*) dbm->GetTable("RunSummary");
  // Run number and integrated luminosity
  int     r = GetHeaderBlock()->RunNumber();
  double l = rs->LumiTape();

  fYields.SetRunNumber(r,l); ....

int TUpsi::Event(int ientry)
{
    // Apply selection and fill histogram
    if (PassCuts())
      fYields.Increment(); ....

int TUpsi::EndJob()
{
  fYields.Print(PrintLevel()+1); ....
```

Use in NTrks Analysis?

Implemented selection of events

- if no track inside $|\eta|<0.5$
- expectation: some events should be selected
- yield (nEvent/Lumi) should be constant
- is this study useful? yes!
  - there could be 'bad' runs (tracker off, no event has tracks)
  - could be a run was incompletely processed (yield too small)
  - anything we cannot think of right now
- but careful: random tests of your sample might bias your view on what is going on
  - to fix a problem, like 'was the tracker off in this run?': cannot use yield as argument
  - we have to find the real reason for the problem and fix it independently of our analysis sample

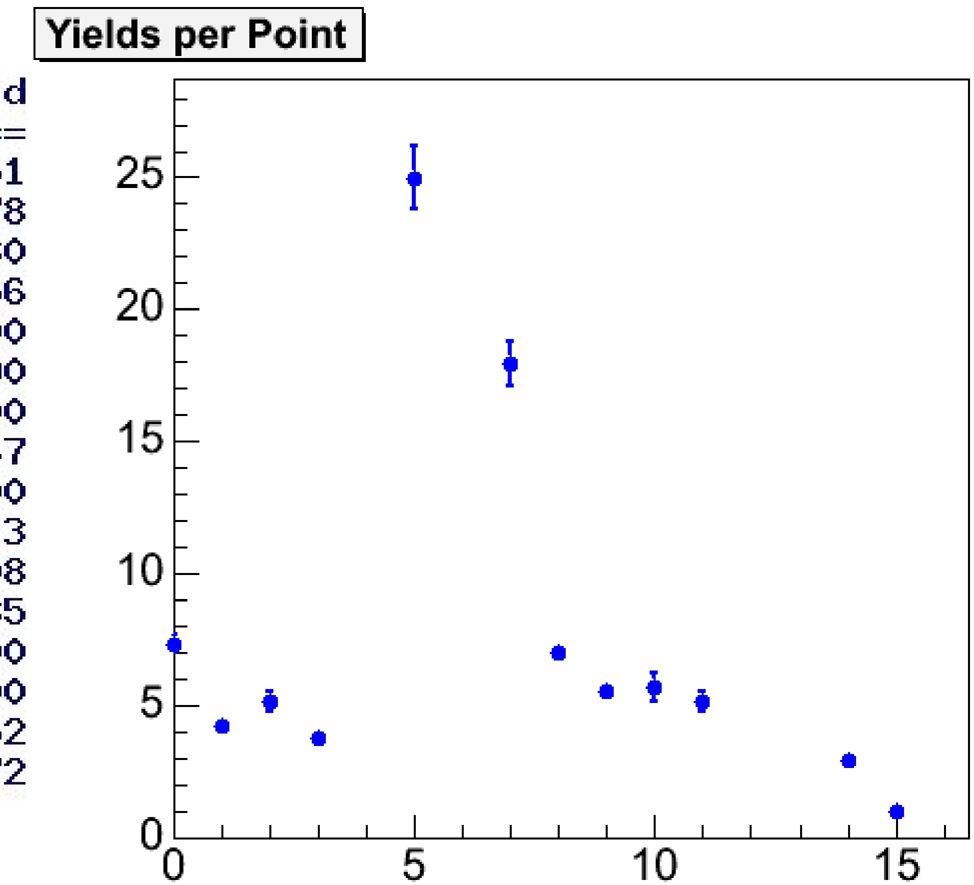
Yield Study Result in NTrks

Summary of Yields (NTrks): 16 Runs analyzed.

Idx	RunNumber	Luminos	Events	Yield
0	175079	60.93	447	7.33651
1	175088	66.89	285	4.26078
2	175143	31.56	163	5.16530
3	175146	74.29	281	3.78256
4	175089	0.00	652	-1.00000
5	175090	17.21	430	24.98490
6	175091	0.00	1032	-1.00000
7	175092	26.79	481	17.95167
8	175150	282.26	1975	6.99700
9	175155	78.02	432	5.53713
10	175195	19.56	111	5.67408
11	175196	36.71	189	5.14835
12	176584	0.00	683	-1.00000
13	176623	0.00	40	-1.00000
14	179103	518.87	1533	2.95452
15	182629	504.71	497	0.98472

==== TOTALS ====

-----	Runs	Luminosity	N Events	Yield
all :	16	1717.80	9231	5.37373



Conclusion: something is wrong here!

- expected yield to be constant but see wild variations?

NTrks Sample

Sample production

- several tens of millions of minimum bias events in CDF
- did not make sense to process them all (we have 240k)
- analysis will be systematics limited (see paper)
- took some 'random' subsample of the entire data
- run by run tests will not work
  - runs are not guaranteed to be contiguous: will have pieces
  - luminosity not available on smaller unit basis
- for our analysis these test should not be an important factor, though the ultimate experimenter's proof is always doing the study

Bottom line

- do not do this test because it cannot work on this sample

Matching MC Truth and Reconstruction

Concept

- MC generator create particles: $m_i, \vec{x}_i, \vec{p}_i$
- particle passes through detector and leaves traces
- reconstruction algorithm finds them: helix parameters
- connect reconstructed and generated information

Why is it useful?

- can study reconstruction algorithms in any detail
- understand detailed detector response
- calibrate or tune algorithms
- measure efficiency
- measure resolution

Matching MC Truth and Reconstruction

Simplest case

- kinematic matching
- use for example: z_0, p_T and φ_0
- need to find criteria to match: reconstructed and generator quantities are different.... what agreement defines match?
- resolutions of various kinematic quantities are the key
- its a bit of a bootstrap because matching is needed for resolutions
- algorithm is straight forward, needs very little input
- works well as long as there are few particles around
- works less well in dense environment (many tracks close together)
- depends on resolutions

Matching MC Truth and Reconstruction

Full blown solutions

- match on the basis of MC hits
- each hit 'knows' which particle created it
- analyze the reconstructed components of a track and determine which particle was responsible
- sounds like it should always work.... not really!
  - each hit can be created by a bunch of particles
  - a track does not necessarily contain only hits of one particle
  - imagine two track going very close to each other
  - determine MC particle mostly responsible for the reconstructed particle (dial matching: 80% of hits from matched MC particle?)
- hit matching is best you can do, no algorithm is perfect: make sure imperfections do not taint your result
- it is complex to implement and possibly use

Let's Look at Data and MC

First thing

- does MC describe data?
- conventional: data dots, MC line

Start from the top

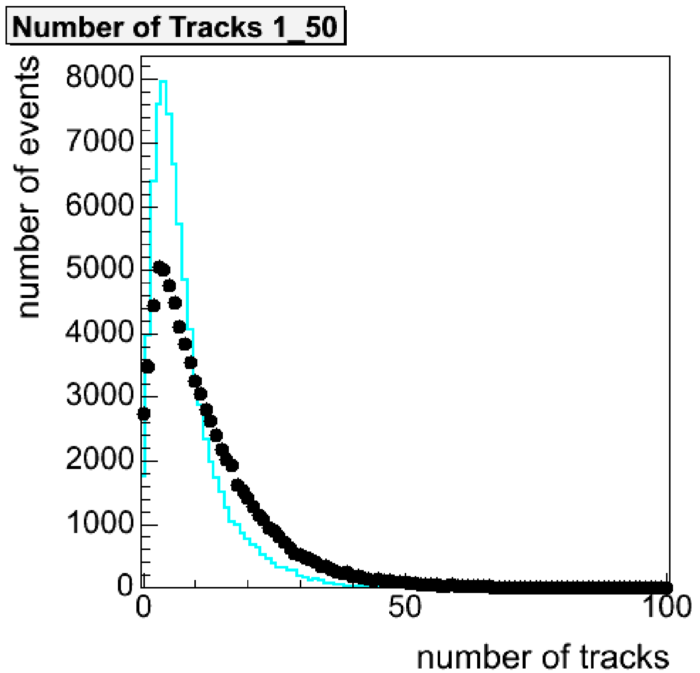
- number of tracks in $|\eta| < 1.5$ (0.5)
- does not agree! why?

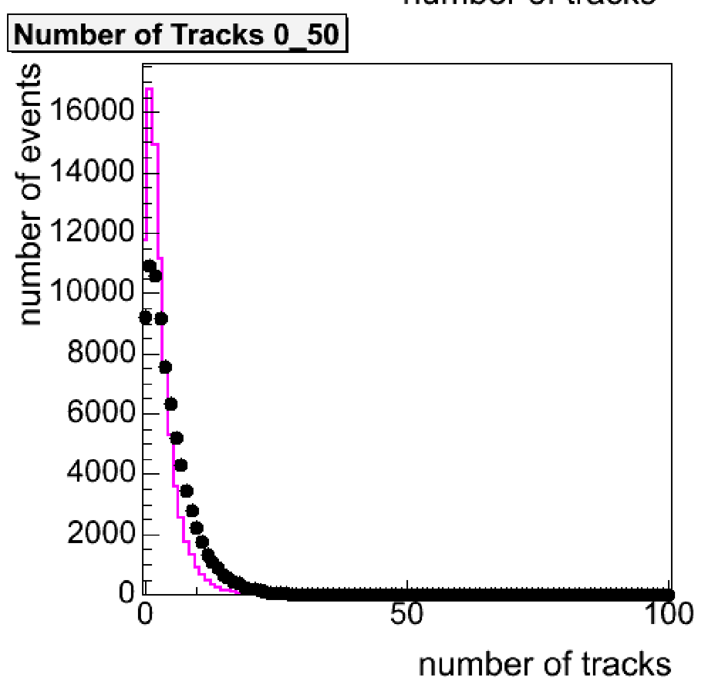
Monte Carlo

- simulates single event, no pile-up
- there should be more tracks on average in the data

Fix to make MC useful

- usually fix MC, this time fix data
- reject events with pile-up

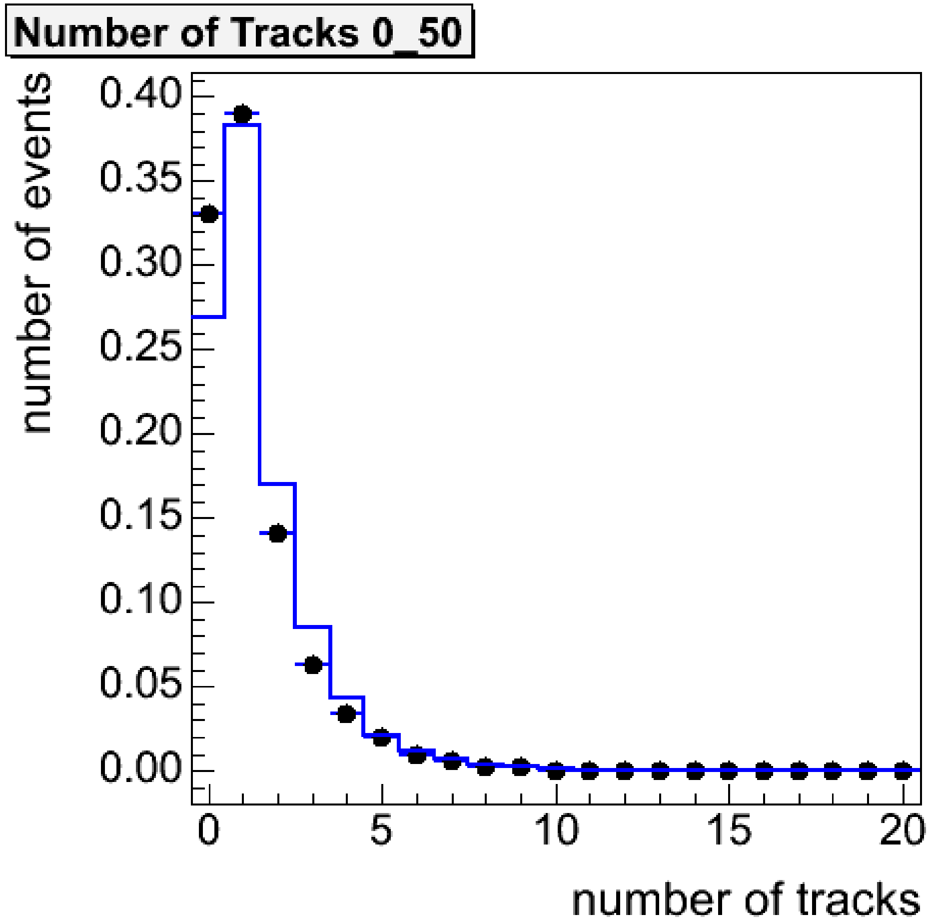




Comparing Data and Monte Carlo

Try very simple test

- calculate maximum *dz* between all tracks
- pile up events have large *dz* (*z* beam width 30cm)
- single collisions: small *dz*
- tried with 4 cm
- redo MC and data
- lost tracks (dangerous)
  - Ks, lambdas, secondaries ....
- but getting getter agreement
- detailed PV algorithm useful



Comparing Data and Monte Carlo

Still

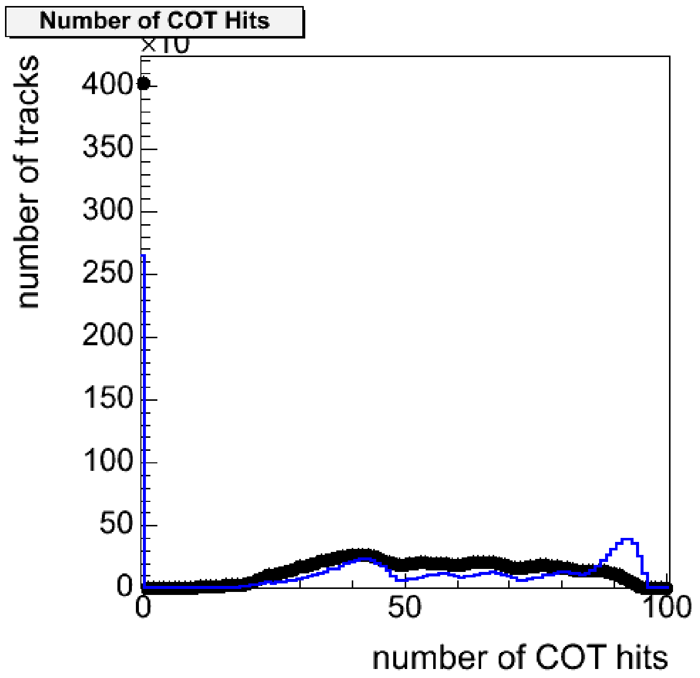
- does MC describe data?
- long list of pictures one can look at

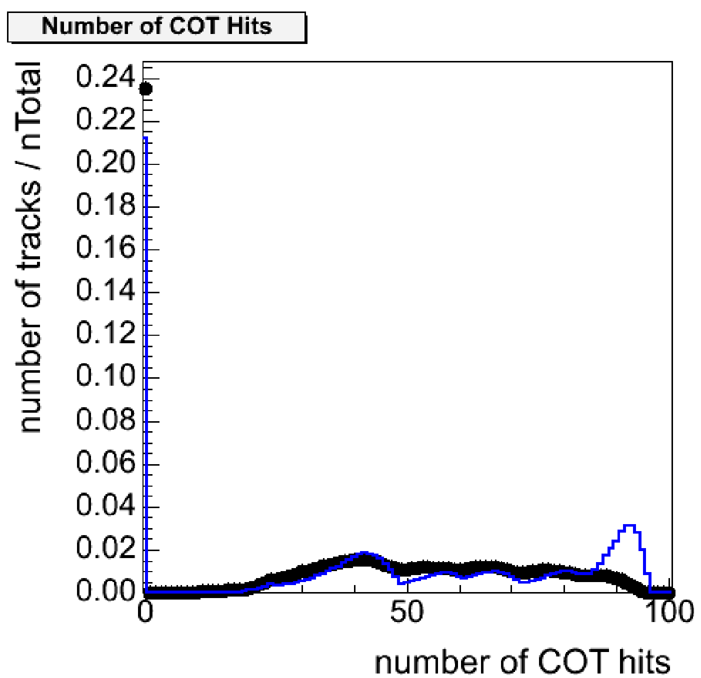
Start from the bottom

- number of COT hits on track
- does not agree! why?
- normalization will be off: fix it
- still does not agree

What is wrong?

- comparing all tracks
- at low momenta MC does not describe data

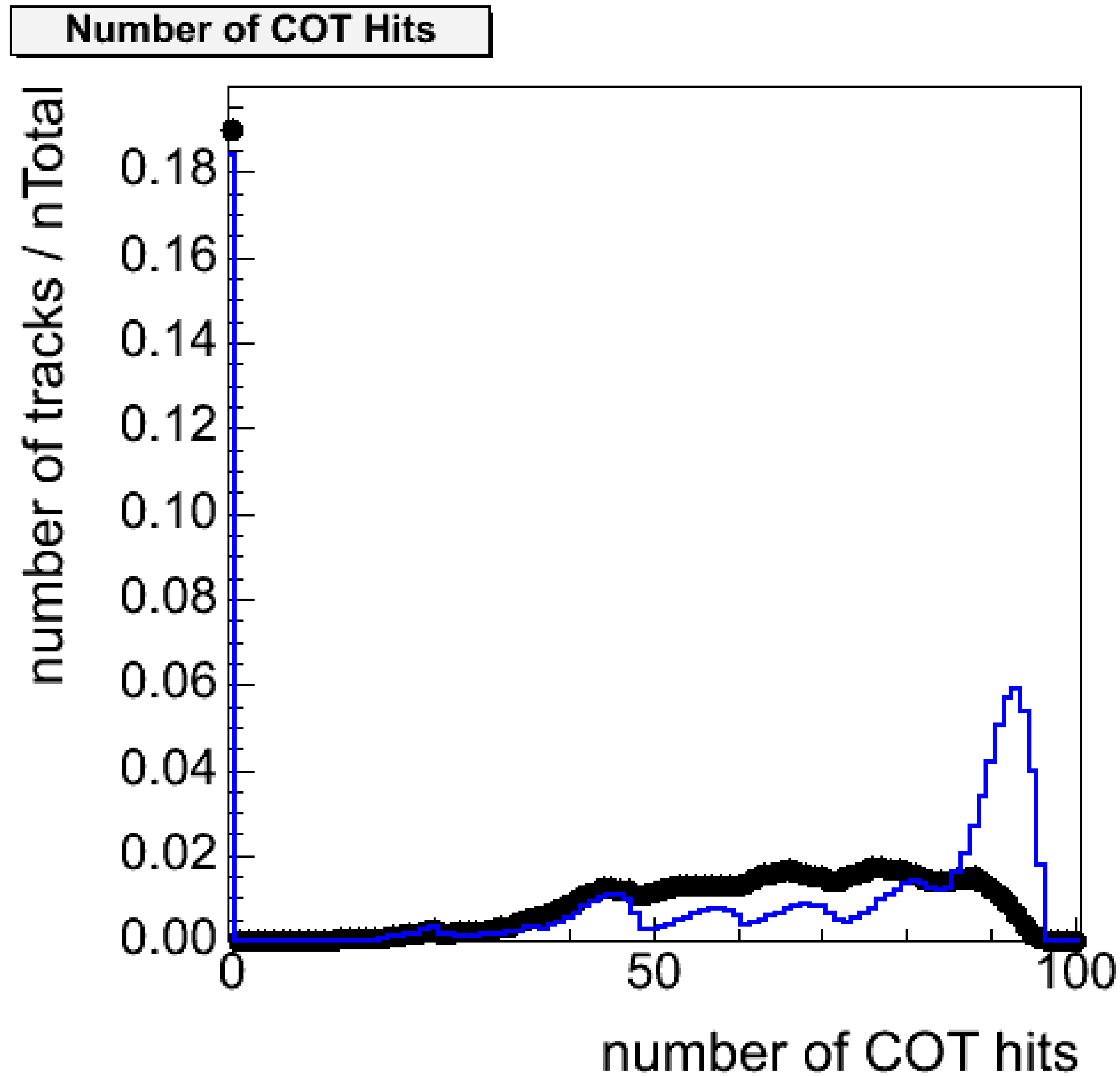




Comparing Data and Monte Carlo

Comparison status

- still does not agree
- but tracks at nHits=0 got much better
- few tracks seem to be lost
- tracking very inclusive
- top level comparison still wrong (revisit)



Summary

- little effect on analysis after cleaning tracks
- fix one effect at a time
- 'real' fix very elaborate

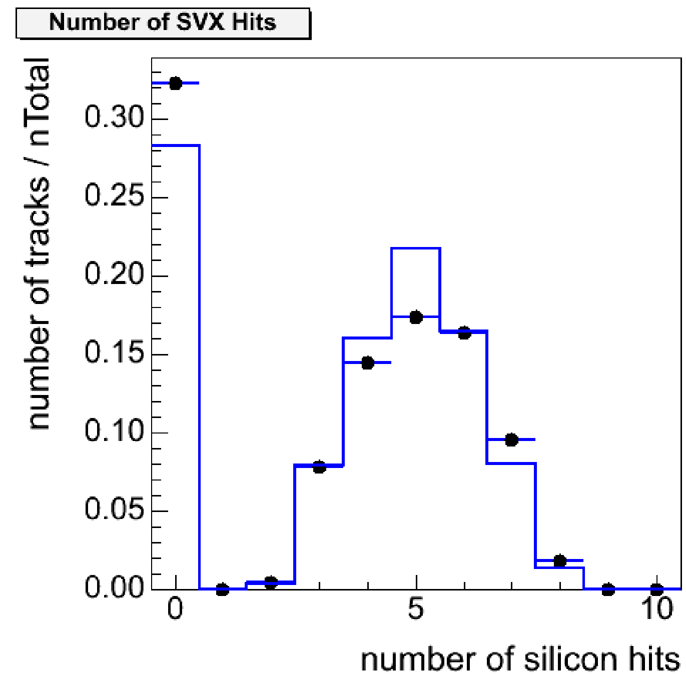
Comparing Data and Monte Carlo

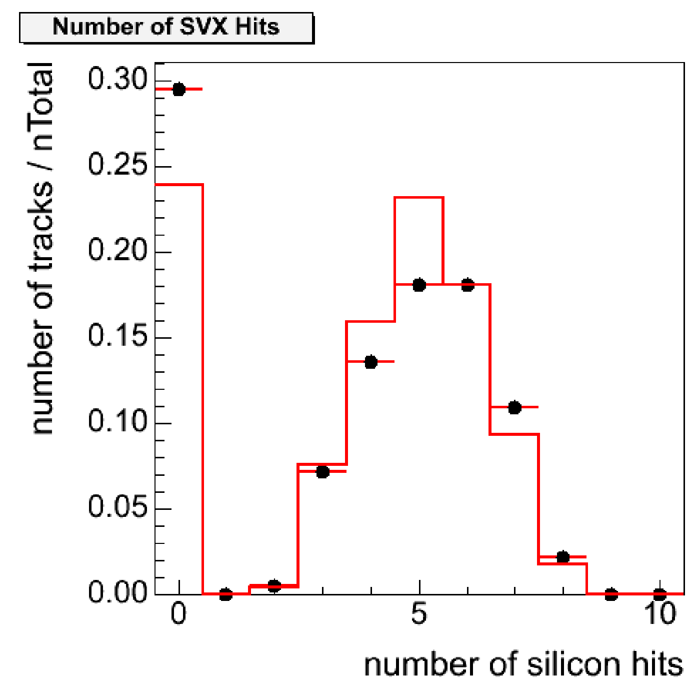
Check out silicon hits

- simulation of silicon easier
- occupancy is not as high
- agreement is not terrible
- clean tracks: still more tracks with zero hits

What is wrong?

- more complex problem (unsure)
- MC simulates for different runs
- silicon reconstruction was slightly different
- *z* distribution of events different?
- evaluate systematics? do we need silicon?





The Hand-In

Important components

- as a reference point for everybody choose
  - $|\eta|<0.5$ and $p_T > 0.5$ GeV to measure the track multiplicity
- explain what you want to measure and why
- compare data and Monte Carlo, in as far as needed
- how efficient is the tracking and how well do you know?
- explain your cuts
  - what they are used for? is there a good picture to show their effect?
  - do they affect your result? by how much?
- explain your fitting function (not how the fitter works) and the result
- determine the sources of uncertainty and their values

The Hand-In

Uncertainties

- no strict rules: use common sense and try to bracket them
- do not invent things, present an argument of why
- example:
  - how much does a cut on number of COT hits affect my analysis?
  - because I have corrected the effect of tracking inefficiency already using the Monte Carlo I am only affected if data and Monte Carlo do not agree (they do not)
  - in the area where I cut (at CtAx(St) > 10 hits) data and Monte Carlo do not disagree very much, just shift the cut to 22 hits, 12 corresponds to one additional super layer and is about the average hit shift

Is is possible to compare to existing results?

What to improve in the analysis, if you had time?

Conclusion

Number of track analysis due next week

- you can do a lot of things with data and the Monte Carlo
- focus on a restricted number of them and think them through
- if you have not covered everything in the hand-in, it is not going to be a disaster
- pick out a set of systematic uncertainties you want to cover and discuss them with numbers and arguments

The analysis is more difficult then it seems at first

- do not be disappointed if you cannot do it all, people have written theses on this topic

Next Lecture

Onia as Probes in Heavy Ion Physics

- what are onia? and bit of history
- why are they interesting in heavy ion physics?
- what can we do with them in High Energy physics?
- production and their decay
- general reconstruction of onia