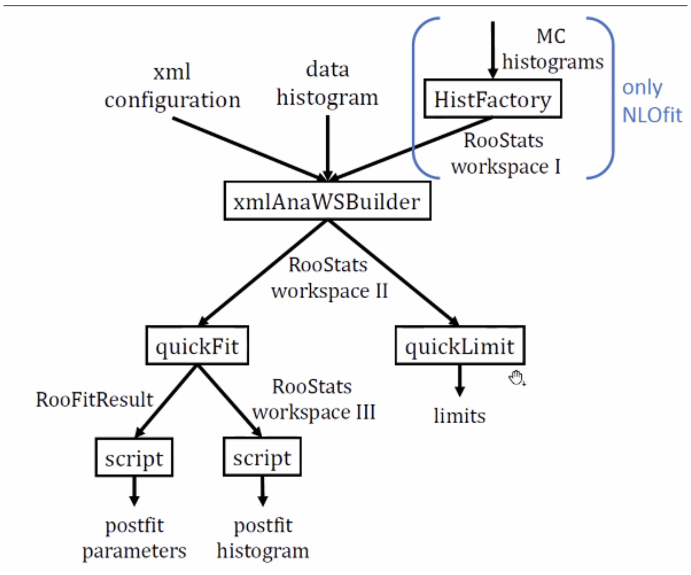
Walkthrough for statistical framework by Falk
==

Slides: [here]

Existing files are untouched but many more files available. Now the repo includes:
- ATLAS style package
- Bash scripts
- quickFit as a submodule

General working of the code

Various interconnected pieces, nice diagram:



This also shows how to give other inputs in terms of background estimation that isn't a fit - HistFactory

The output of quickFit isn't too useful → but Falk/CaterinaM have scripts for postfit parameters and histograms and correlation matrix

Q: how does quickLimit know about quickFit results? (Does it?)
Answer: it doesn't in the above, but it should run the same thing, so the parameters for any fit should be initialized in xmlAnaWSBuilder.
However, there is a possibility of plugging in Workspace III into quickLimit

Top-level XML - one per analysis is needed
- Each channel (signal region/control region) has a category file
    - Each of those has data (input histo) / sample (bkgs xml) specifiers
        - Bkg xml: for fit it's just one, the fit functions (with parameters that can be fitted)
            - To make parameters not float, set the range the same as the parameeter

        - NLOFit: reading external HistFactory histograms for each systematics
        - Signal xml works similarly as bkg xmls
            - Should contain also the systematics, but we haven't looked into it yet
    - Normalization: all input samples should be normalized to 1 and scaled by NormFactor (x Lumi)
        - One can set this to the # of events you expect in N
        - Have a global lumi and individual lumi per sample
- Need to specify all the parameter of interests (one per signal mass/width)

QuickFit

QuickFit has a number of parameters in the command line on top of the xml - see slides.

Falk added 2 python scripts in the directory:
- Extract fit from workspace (post-fit distribution)
- Extract [missing a piece]

Fitting automation

Use the scripts setup_buildAndFit.sh / _swift, _nlofit

Note that you have to change the xmlAnaWSBuilder config files every time you change something when you e.g. run over pseudodata → use "template" files with placeholders that get substituted in run_buildAndFit_swift.sh.

Automation = better to have a cluster available.

[missing a piece here]

Pseudodata

There is a python script to generate pseudodata (Poisson-fluctuated) - called generatePseudoData, Juno's PD generator can be used for fancier things.

Can be processed using the _loop.sh

Plotting limits

The script plotLimits_joined.py will read quickLimit output for different masses
The script createExtractionGraph.py will read quickFit output and then do signal extraction plots as a function of number of injected signal events
The coverage test scripts are createCoverageGraph.py / plotFalseExclusionCandles.py