The RooStats Project

K.Cranmer (New York University, ATLAS)
L. Moneta (CERN, ROOT)
G. Schott (Karlsruhe Institute of Technology, CMS)
W. Verkerke (Nikhef, ATLAS and RooFit)
and also contributions from
D. Piparo (CMS), M. Pelliccioni (CMS), A. Lazzaro (ATLAS)
K. Belasco (ATLAS), M. Wolf (CMS), S. Schmitz (CMS)
RooStats

New project to create advanced statistical tools needed by LHC experiments

Joint contribution ATLAS, CMS ROOT and RooFit
  developments followed by ATLAS and CMS statistics committees

Common framework for statistical calculations
  work on arbitrary models and datasets
  implement most accepted techniques (frequentists, Bayesian and likelihood based methods)
  provide utility for combinations

Built on top of RooFit
  provides generic and convenient description of models (probability density function or likelihood functions)
  tools to easily generate modes (workspace factory)

“Gaussian::g(x[2,-10,10],m[0],s[3])”
Statistical Applications

Common purposes:

- point estimation: determine the best estimate of a parameter
- estimation of confidence (credible) intervals
  - multi-dimensional contours or just a lower/higher limit
- hypothesis tests: evaluation of p-value for one or multiple hypotheses (significance)
- goodness-of-fit: how well a model describes the data

Analysis combination:

- Performed at analysis level: full information available to treat correlations

For these things RooStats can help you
Overview of RooStats Classes

Defined interfaces to statistical methods

- interval estimation
- hypothesis test

```
Defined interfaces to statistical methods

IntervalCalculator
- BayesianCalculator
- MCMCCalculator
- NeymanConstruction
- FeldmanCousins
- HypoTestInverter

CombinedCalculator
- ProfileLikelihoodCalculator
- HybridCalculator

HypoTestCalculator

Computed p-values, significance, CLs, ...

ConfidenceInterval:
- LikelihoodInterval
- MCMCInterval
- PointSetInterval
- SimpleInterval

LowerLimit & UpperLimit: bounds of the interval

+ other code and "Utilities"
```
Method based on properties of the likelihood function

**Profile likelihood function:**

\[
\lambda(\mu) = \frac{L(x|\mu, \hat{\nu})}{L(x|\hat{\mu}, \hat{\nu})}
\]

- maximize w.r.t nuisance parameters \( \nu \) and fix POI \( \mu \)
- maximize w.r.t. all parameters
- \( \lambda \) is a function of only the parameter of interest \( \mu \)

Uses asymptotic properties of \( \lambda \) based on Wilks’ theorem:

- \( \lambda \) is a gaussian function \( \Rightarrow -2\log \lambda \) is a parabola around the minimum

**Method of MINUIT/MINOS**

- lower/upper limits for 1D
- contours for 2 parameters

```cpp
ProfileLikelihoodCalculator plc(*data, *model, *POI);
plc.SetConfidenceLevel(0.683);
LikelihoodInterval* interval = plc.GetInterval();
double lowerLimit = interval->LowerLimit(*S);
double upperLimit = interval->UpperLimit(*S);
LikelihoodIntervalPlot plot(interval);
plot.Draw();
```
Hypothesis Test with Profile Likelihood

Profile Likelihood can be used for hypothesis tests using the asymptotic properties of the profiled likelihood ratio:

\[ \lambda(\mu) = \frac{L(x|\mu, \hat{\nu})}{L(x|\hat{\mu}, \hat{\nu})} \]

- Null hypothesis (H₀): \( \mu = \mu_0 \)
- Alternate hypothesis (H₁): \( \mu \neq \mu_0 \)

- Distribution of \(-2\log\lambda\) is asymptotically a \( \chi^2 \) distribution under H₀

- \( p \)-value and significance can then be obtained from the \(-2\log\lambda\) ratio

\[ \text{significance: } n_\sigma = \sqrt{-2 \log \lambda} \]

```
S->setVal(0); // set value of POI to zero
plc.SetNullParameters(*S);
HypoTestResult* hypotest = plc.GetHypoTest();
double alpha = hypotest->NullPValue();
double significance = hypotest->Significance();
```
Hybrid Calculator

**Hypothesis test calculator** based on sampling using toys a test statistic, \( Q \) for the two hypotheses (e.g. B and S+B)

- by default use the likelihood ratio
- optionally one can use the number of events or the profile likelihood ratios:

\[
Q_{LEP} = \frac{L_{S+B}(\mu = 1)}{L_B(\mu = 0)}
\]

\[
Q_{TEV} = \frac{L_{S+B}(\mu = 1, \nu)}{L_B(\mu = 0, \nu')}
\]

or

\[
\lambda(\mu) = \frac{L_{S+B}(\mu, \nu)}{L_{S+B}(\hat{\mu}, \hat{\nu})}
\]

Returns a **HypoTestResult** with the hypothesis' p-values

- \( CL_{sb} \), \( CL_b \), \( CL_s \) and the significance

**Frequentist method with a Bayesian integration** (Cousins-Highland) of the nuisance parameters (toys are generated varying the nuisances according to their pdf's)

**Class redesign in 5.27** (to use same test statistic classes as in Neyman construction)

```c++
// create H.C. using model for B and S+B and nuisance Pdf
HybridCalculatorOriginal hc(*data,*modelSB,*modelB,nuisPar,nuisPdf);
HybridResult* result=hc->GetHypoTest();
HybridPlot* plot=result->GetPlot("hcPlot","p Values Plot",100);
plot->Draw();
// retrieve CLsb, CLb and CLs
double clsb  = result->CLsplusb();
double clb   = result->CLb();
double cls   = result->CLs();
double significance = result->Significance();
```
**Neyman Construction**

**Frequentist approach to interval estimation**

- **Construct a** $1-\alpha$ **confidence region on the parameter** $\theta$

  - Determine the distribution of the test statistic $x$ for many values of $\theta$, $f(x|\theta)$
  - Determine each time the $1-\alpha$ CL region on $x$ (for a given ordering rule)
  - Look at the value of $x$ obtained in data ($x_0$)
  - The intersection defines the confidence region on $\theta$ [ $\theta_1$ ; $\theta_2$ ]

- Different ordering rules possible (shortest interval, most probable, central,...)
  - Feldman-Cousins: ordering rule based on the profile likelihood ratio

- All this can be done in RooStats
  - **NeymanConstruction** or **FeldmanCousins** classes

- Toy MC generation for estimating test statistics distribution at each $\theta$
  - RooStats provides different test statistics classes

- PROOF can also be used for sampling the test statistics (CPU intensive process)
Hypothesis Inverter

Invert result from the hypothesis test (e.g. from HybridCalculator) to compute a limit (*HypoTestInverter* class)

- run many runs of the Hybrid calculator scanning the POI
- obtain an upper limit from the $CL_{sb}$ or $CL_s$ curve
- method used at LEP and Tevatron
- can run using a fixed grid or automatically to minimize toys runs
- estimate also of the expected errors

![Graph example](image-url)
Bayesian Analysis

Bayesian Theorem

\[ P(\mu | x) = \frac{\int L(x | \mu, \nu) \Pi(\mu, \nu) d\nu}{\int \int L(x | \mu, \nu) \Pi(\mu, \nu) d\mu d\nu} \]

BayesianCalculator class

- posterior and interval estimation using numerical integration
- working only for one parameter of interest
- support for different integration algorithms:
  - adaptive (one dim or multi-dim) or using MC (Vegas, Miser from MathMore/GSL)
  - limited to few nuisance parameters (\(\lesssim 10\))
- can compute central (or lower/upper limits) and shortest intervals
- provide plot of posterior and interval

Example:
95% CL central interval

BayesianCalculator bc(*data, *modelPdf, *POI);
bc.SetConfidenceLevel(0.683);
bcs.SetLeftSideTailFraction(0.5);
SimpleInterval* interval = bc.GetInterval();
double lowerLimit = interval->LowerLimit(*S);
double upperLimit = interval->UpperLimit(*S);
MCMC Calculators

**MCMCCalculator** class

- integration using Markov-Chain Monte Carlo (Metropolis Hastings algorithm)
- can deal with more than one parameter of interest and many nuisance parameters
- possible to specify *ProposalFunction*
- can visualize posterior and also the chain

**BATCalculator** class

- provided by the BAT package (not part of Roostats) in their last release 0.4.1
  - see also http://www.mppmu.mpg.de/bat/
- valuable alternative for cross-checks
- various options for controlling the Markov chain
Workspace class in RooFit (**RooWorkspace**) with:
- full model configuration
- PDF and parameter/observables descriptions
- uncertainty/shape of nuisance parameters
- (multiple) data sets
- Maintain a complete description of all the model
- possibility to save entire model in a ROOT file
- Combination of results joining workspaces in a single one
- All information is available for further analysis
- common format for combining and sharing physics results

```cpp
RooWorkspace workspace("Example_workspace");
workspace.import(*data);
workspace.import(*pdf);
workspace.defineSet("obs","x");
workspace.defineSet("poi","mu");
workspace.importClassCode();
workspace.writeToFile("myWorkspace")
```
ModelConfig

- **ModelConfig** class in RooStats
- Contains the meta information about meaning of parameters and pdf's
  - Bayesian tools need prior for all parameters
  - Hybrid prior for only nuisances
- Input to RooStats calculator classes
- ModelConfig can be imported in workspace for storage and later retrieval

```cpp
// specify components of model for statistical tools
ModelConfig modelConfig("G(x|mu,1)");
modelConfig.SetWorkspace(workspace);
// set components using the name of ws objects
modelConfig.SetPdf("normal");
modelConfig.SetParameterOfInterest("poi");
modelConfig.SetObservables("obs");

// Bayesian tools would also need a prior
modelConfig.SetPriorPdf("prior");

// can import ModelConfig into workspace too
workspace.import(*modelConfig);

// set and to import into workspace
modelConfig.SetPdf(*pdf);
```

Alternatively, ModelConfig can be used to import the components directly into the workspace.
Example Application: Higgs at LHC

- ATLAS-CMS toy Higgs combination was performed this summer
  - each experiment wrote down their likelihood based on number countings in several measurements and control samples (H→WW, \( M_H = 160 \text{ GeV} \))
  - full models was described using RooFit
  - RooStats tools were used for the statistical analysis
  - common parameter of interest: ratio of cross-sections \( r = \frac{\sigma}{\sigma_{SM}} \)

- CMS analysis: 3 observables, 37 nuisance parameters
- ATLAS analysis: 9 observables, 12 nuisance parameters

- ATLAS analysis more conservative, while CMS used NN and was much more powerful (it is just a toy study, real analysis will be different!)

- Test for discovery (estimation of significance)
  - profile likelihood ratio and Bayesian/frequentist (Hybrid)

- Cross section measurement
  - profile likelihood ratio, Feldman-Cousins, inverted hypothesis test using Hybrid Calculator and Bayesian MCMC (and using also BAT)

- 7 statistical techniques performed in just few days. A real success!
  - see http://indico.cern.ch/conferenceOtherViews.py?view=atlas&confId=100458
ATLAS–CMS Model

Graphical representation of the objects in the combined model

The full model has 12 observables and ~50 parameters

\[ \mu = \frac{\sigma_{BR}}{\sigma_{SM \, BR_{SM}}} \]
Profile Likelihood Results
from fitting the combined likelihood function

Significance

\[ \sqrt{-2 \ln L(r = 0, \hat{\nu}) / L(\hat{r}, \hat{\nu})} \]

<table>
<thead>
<tr>
<th></th>
<th>with systematics</th>
<th>r</th>
<th>signif.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATLAS</td>
<td>1.13</td>
<td>2.70</td>
<td></td>
</tr>
<tr>
<td>CMS</td>
<td>0.98</td>
<td>4.89</td>
<td></td>
</tr>
<tr>
<td>COMBI</td>
<td>1.02</td>
<td>5.58</td>
<td></td>
</tr>
</tbody>
</table>

Upper limit from \(-2\log \lambda\)

<table>
<thead>
<tr>
<th></th>
<th>with systematics</th>
<th>r</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATLAS</td>
<td>0</td>
<td>0.79</td>
<td></td>
</tr>
<tr>
<td>CMS</td>
<td>0</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>COMBI</td>
<td>0</td>
<td>0.25</td>
<td></td>
</tr>
</tbody>
</table>

Reminder: these numbers are not to be taken seriously; they are only examples that roughly approximate the same level of complexity that should be expected from the real results.
Hybrid Results (significance)

Run Hybrid Calculator with 3 test statistics

\[ Q_{\text{LEP}}, Q_{\text{TEV}}, \lambda(r) \]

Computing p-value for significance at 5 \( \sigma \) requires generation of many toys

Need to be efficient in toy generation and test statistics evaluation (e.g. fitting for \( \lambda \))

<table>
<thead>
<tr>
<th></th>
<th>test statistics</th>
<th>significance (no syst.)</th>
<th>significance (with syst.)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ATLAS</strong></td>
<td>( Q_{\text{LEP}} )</td>
<td>3.07 \pm 0.01</td>
<td>2.8 \pm 0.1</td>
</tr>
<tr>
<td></td>
<td>( Q_{\text{TEV}} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \lambda(r) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CMS</strong></td>
<td>( Q_{\text{LEP}} )</td>
<td>6.22 \pm 0.02</td>
<td>4.77 \pm 0.02</td>
</tr>
<tr>
<td></td>
<td>( Q_{\text{TEV}} )</td>
<td></td>
<td>&gt; 4.6</td>
</tr>
<tr>
<td></td>
<td>( \lambda(r) )</td>
<td></td>
<td>4.3 \pm 0.1</td>
</tr>
<tr>
<td><strong>COMBI</strong></td>
<td>( Q_{\text{LEP}} )</td>
<td></td>
<td>&gt; 4.6</td>
</tr>
<tr>
<td></td>
<td>( Q_{\text{TEV}} )</td>
<td></td>
<td>&gt; 3.5</td>
</tr>
<tr>
<td></td>
<td>( \lambda(r) )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Upper limits results

#### Result of Hypothesis test inversion

- **Feldman-Cousins (no syst.)**
  - Test stat: $\lambda(r)$
  - Rule: $CL_{S+B}$
  - Sampling: toys
  - UL ATLAS: $0.69 \pm 0.05$
  - UL CMS: -
  - UL COMBI: -

- **Profile LR (Wilks)**
  - Test stat: $\lambda(r)$
  - Rule: $CL_{S+B}$
  - Sampling: asymptotic
  - UL ATLAS: 0.79
  - UL CMS: 0.28
  - UL COMBI: 0.25

- **Feldman-Cousins++**
  - Test stat: $\lambda(r)$
  - Rule: $CL_{S+B}$
  - Sampling: toys
  - UL ATLAS: $0.78 \pm 0.05$
  - UL CMS: $0.26 \pm 0.02$
  - UL COMBI: $0.23 \pm 0.02$

- **Hybrid**
  - Test stat: $Q_{LEP}$
  - Rule: $CL_{S}$
  - Sampling: toys
  - UL ATLAS: $\sim 0.68$
  - UL CMS: $0.29 \pm 0.03$ (LandS)
  - UL COMBI: -

- **Hybrid**
  - Test stat: $Q_{LEP}$
  - Rule: $CL_{S+B}$
  - Sampling: toys
  - UL ATLAS: $\sim 0.61$
  - UL CMS: -
  - UL COMBI: -

- **Bayesian**
  - Test stat: n/a, flat prior on $r$
  - Rule: MCMC*
  - Sampling: toys
  - UL ATLAS: 0.72
  - UL CMS: 0.31
  - UL COMBI: 0.28

---

**good agreement among the different statistical methods!**
Other classes and utilities

- **HLFactory**: simple wrapper around the RooWorkspace
  - allows to define the likelihood model in a simple text file (data cards)
  - separation of physics model and C++ code
  - simple combination of multiple channels

- **Template Factory**: tool to build RooFit models from templated TH1 histograms (most used in analysis)
  - variational histogram provided for representing systematics

- **SPlot**: a technique used to produce weighted plots of an observable distribution

- **BernsteinCorrection**: utility to add polynomial corrections

- Number counting analyses utilities
Conclusions

- Code of Atlas and CMS combined and improved to form RooStats
- RooStats distributed with ROOT since end of 2008
  - Recommend using latest release 5.27.06, and soon (December) 5.28
- Allows statistical studies for LHC (and other) analyses
  - Speak common language for comparisons and combination
  - Use different statistical techniques with same model
    - Flexible enough to accommodate all/most cases
    - Most statistical methods one would need are there
- ATLAS–CMS Higgs combination has shown RooStats capabilities
- Need consolidation and to complete the validation
  - Work on better performance: parallelization (PROOF, GPU,...)
- Building expertise within ATLAS and CMS (tutorials last year)
- Open project, new contributors are welcome
Documentation and user support

- **RooStats TWiki:** [https://twiki.cern.ch/twiki/bin/view/RooStats/WebHome](https://twiki.cern.ch/twiki/bin/view/RooStats/WebHome)
- **RooStats users guide** (under development, to be completed)
- **ROOT reference guide:** [http://root.cern.ch/root/htmldoc/ROOSTATS_Index.html](http://root.cern.ch/root/htmldoc/ROOSTATS_Index.html)
- **RooFit’s users guide:** [http://root.cern.ch/drupal/content/users-guide](http://root.cern.ch/drupal/content/users-guide)
- **RooStats November tutorials:**
  - Lecture of L. Lista on statistics: [http://indico.cern.ch/conferenceDisplay.py?confId=73545](http://indico.cern.ch/conferenceDisplay.py?confId=73545)
  - Tutorial contents: [http://indico.cern.ch/conferenceDisplay.py?confId=72320](http://indico.cern.ch/conferenceDisplay.py?confId=72320)
- **RooStats user support:**
  - Submit bugs to ROOT Savannah: [https://savannah.cern.ch/bugs/?func=additem&group=savroot](https://savannah.cern.ch/bugs/?func=additem&group=savroot)
- **Contacts for statistical questions:**
  - ATLAS statistics forum: hn-atlas-physics-Statistics@cern.ch (Cowan, Gross et al)
  - CMS statistics committee: (Cousins, Demortier et al)
    - via hypernews: hn-cms-statistics@cern.ch or directly: cms-statistics-committee@cern.ch
Terminology

- **Observable** (or random variable): quantities that are directly measured by an experiment (e.g., candidates mass, helicity angle, NNet output) – they form a dataset

- **Model**: based on probability density function (PDF) that describes one or multiples observables – parametric or non-parametric. PDF are normalized such that their integral over any observable is 1

- **Parameters of interest**: parameters of the model that one wishes to estimate or constrain (e.g., particle mass, cross-section)

- **Nuisance parameters**: parameters of the model that are uncertain but not “of interest” (systematics-associated normalization or shape parameters)
  
  - treatment of systematic uncertainties depends on the statistical method used
Feldman–Cousins class

- Neyman construction with FeldmanCousins ordering rule

```cpp
RooStats::FeldmanCousins fc;
// set the distribution creator, which encodes the test statistic
fc.SetPdf(model);
fc.SetParameters(parameters);
fc.SetTestSize(0.05);  // set size of test
fc.SetData(*data);
fc.UseAdaptiveSampling(true);
// number counting analysis: dataset always has 1 entry with N events observed
fc.FluctuateNumDataEntries(false);
fc.SetNBins(30);  // number of points to test per parameter
```

- **SetNBins()** specifies the number of points to test on the parameter of interest
- for each point estimate the profile likelihood ratio distribution from the toy MC

Returns the CL as a PointSetInterval: information if tested points are inside or outside the interval