PROCEEDINGS OF SCIENCE

The RooStats Project

L. Moneta [*]
CERN, Geneva, Switzerland
E-mail: Lorenzo.Moneta@cern.ch

K. Belasco

Princeton University, USA

K. S. Cranmer

New York University, USA

S. Kreiss

New York University, USA

A. Lazzaro [†]

CERN, Geneva, Switzerland

D. Piparo

KIT - Karlsruhe Institute of Technology, Germany

G. Schott

KIT - Karlsruhe Institute of Technology, Germany

W. Verkerke

NIKHEF - National Institute for Subatomic Physics, Netherlands

M. Wolf

KIT - Karlsruhe Institute of Technology, Germany

RooStats is a project to create advanced statistical tools required for the analysis of LHC data, with emphasis on discoveries, confidence intervals, and combined measurements. The idea is to provide the major statistical techniques as a set of C++ classes with coherent interfaces, so that can be used on arbitrary model and datasets in a common way. The classes are built on top of the RooFit package, which provides functionality for easily creating probability models, for analysis combinations and for digital publications of the results. We will present in detail the design and the implementation of the different statistical methods of RooStats. We will describe the various classes for interval estimation and for hypothesis test depending on different statistical techniques such as those based on the likelihood function, or on frequentists or bayesian statistics. These methods can be applied in complex problems, including cases with multiple parameters of interest and various nuisance parameters.

PoS(ACAT2010)057

PROCEEDINGS OF SCIENCE

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research, ACAT2010
February 22-27, 2010
Jaipur, India

[*] Speaker.

[†] Speaker replacing L. Moneta

1. Introduction

The Large Hadron Collider (LHC), which started to take data at the end of 2009, is expected to produce an unprecedented amount of data to be analyzed. These data offer a tremendous potential for the discovery of new physics and pose many challenges to the statistical techniques used in High Energy Physics (HEP). Different statistical approaches will be used for statistical inference of the data. That implies different techniques to be considered, each one answering different questions and using specific treatment of systematic uncertainties. Furthermore, statistical combinations of analyses of different search channels from a single experiment and combination of results from different experiments will be required. Up to now, in past experiments like those at LEP or at the Tevatron, dedicated code specific to the statistical method and to the analysis has been used to produce a result (for example an exclusion limit) and to perform the combinations.

For the LHC experiments, there has been a need to develop a generic and versatile software for implementing the required statistical tools. The code should be able to use, for the same input model, different statistical methods. The software needs also to be versatile in order to be able to cope with both simple analyses, such as those based on number counting, and complex ones which use the parametrization of experimental distributions.

Three major types of statistical techniques have been identified:

- **Classical / Frequentist**: this “school” of statistics restricts itself to making statements of the form “probability of the data given the hypothesis”. The definition of probability in this context is based on a limit of frequencies of various outcomes. In that sense it is objective.

- **Bayesian**: this “school” of statistics allows one to make statements of the form "probability of the hypothesis given the data", which requires a prior probability of the hypothesis. Often the definition of probability in this context is a “degree of belief”.

- **Likelihood-based**: this intermediate approach also uses a frequentist notion of probability (i.e. does not require a prior for the hypothesis), but it does not guarantee to have the properties (i.e. coverage) that frequentists methods aim to achieve (or achieve by construction). This approach does “obey the likelihood principle” (as do Bayesian methods), while frequentist methods do not.

These methods differ also in the way they incorporate the nuisance parameters for producing the results, so there is a need to have all of them.

In order to fulfill the previously described requirements, a new project, RooStats, was started at the end of 2008, merging previous code developed by Kyle Cranmer for Phystat 2008 [1] and from a CMS project, RooStatsCms [2]. RooStats is organized as a joint collaboration between ATLAS and CMS and is based on ROOT and RooFit. The developments are overseen by the ATLAS and CMS statistical committees. It has been built on top of ROOT [3], since ROOT is the most widely used tool in HEP for data analysis and provides the needed basic software (mathematical libraries, graphics libraries for plotting, I/O and dictionary libraries). RooStats is also based on the classes of the RooFit toolkit [4], which allows a convenient description of the data model and is a well-established software within the HEP community.

All statistical methods require a probability density function and/or a likelihood function as input model. RooFit provides classes for facilitating the declaration of models and a way to re-use them in multiple statistical methods, since it has no statistical notion of parameters and observables. Therefore, it can work naturally with both frequentist and bayesian techniques. RooStats can then be considered as providing high-level statistical tools, while RooFit provides the core data modeling infrastructure.

1.1 Type of Statistical Questions

One of the first steps in any statistical analysis is to carefully pose the question that one wishes to answer. Most of these questions can be classified as follows:

- **Parameter Estimation**: find the most likely (‘best fit’) values of the parameters of a model, given the data.
- **Hypothesis Testing**: based on the data accept or reject a given hypothesis. Often one tests a null hypothesis against an alternative. When the hypothesis has no free parameters it is called ‘simple’ and when it has free parameters it is called ‘composite’.
- **Confidence intervals**: find a region in the parameter space that is consistent with the data. In the frequentist setting, one desires for this interval or region to ‘cover’ the true parameter point with a specified probability (or confidence). In the Bayesian setting, one wishes for the interval or region to contain some fixed amount of the posterior probability.
- **Goodness of Fit**: quantify how well a model fits the data, without reference to an alternative.

RooFit provides the functionality for parameter estimation while RooStats provides tools for confidence interval estimations and hypothesis testing. Classes for goodness of fit are currently provided in the core ROOT mathematical libraries[1].

2. RooStats Interfaces

The fundamental design philosophy adopted for RooStats reflects a clear mapping between mathematical concepts and software implementation. We have tried to identify the fundamental concepts in statistics and then we assign them the corresponding C++ classes or interfaces. The design of the interfaces follows the schema given by the statistical questions shown above.

As shown in figure 1, RooStats provides a common interface for confidence interval calculations, `IntervalCalculator`, and another interface for performing the hypothesis tests, `HypoTestCalculator`.

The `IntervalCalculator` produces, as result, a confidence interval, which is represented by the RooStats `ConfInterval` interface. The resulting type of `ConfInterval` will depend on the implementation used for the interval calculator. There are many types of intervals, they can be simple one dimensional intervals or disconnected regions in multiple dimensions. The common interface provides just the functionality to query whether a given point is inside the interval.

[1] see the ROOT functions `TH1::Chi2Test` and `TMath::KolmogorovTest`

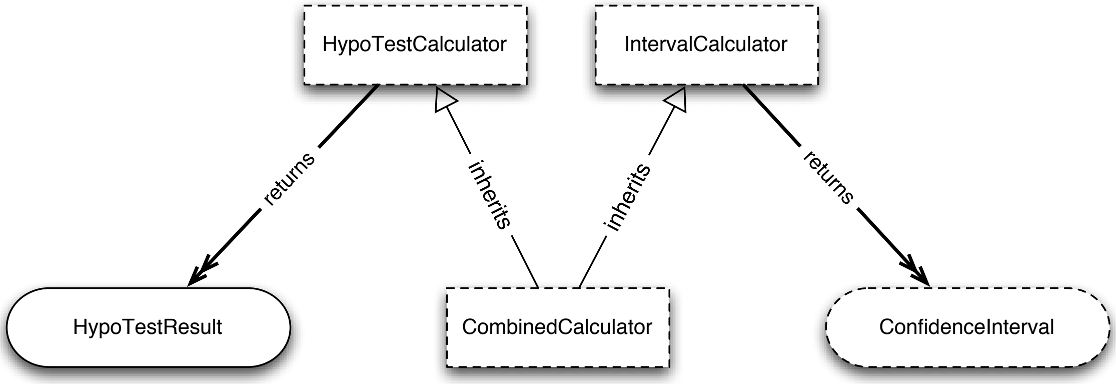

Figure 1: A class diagram of the interfaces for hypothesis testing and confidence interval calculations. The diagram shows the classes used to return the results of these statistical tests as well.

The `IntervalCalculator` interface allows the user to set the input model, the data set, the parameters of interest, the nuisance parameters and the confidence level or size of the test.

The `HypoTestCalculator` is the interface for performing hypothesis tests. The user gives as input the model, the data set and the parameter lists specifying the null and the alternate hypothesis. The result of the class is an object implementing the `HypoTestResult` interface. From this interface the user can retrieve the p values of the null and alternate hypothesis, the confidence level values and the corresponding significance, defined in terms of one-sided Gaussian standard deviations.

3. RooStats Calculators

The interfaces described above are then implemented in RooStats by different classes depending on the statistical techniques used. We now describe briefly all these statistical methods and their corresponding implementations as RooStats classes. For a full description of the RooStats classes, see the RooStats reference documentation [2].

3.1 ProfileLikelihood Calculator

This calculator implements a likelihood-based method to estimate a confidence level and to perform an hypothesis test for a given parameter value. The likelihood function can be defined by the equation:

$$L(\underline{x},\underline{\theta}) = \prod_{i=1}^{N} f(\underline{x}_i,\underline{\theta}), \tag{3.1}$$

where $\underline{x} = (x_a, x_b, x_c, \ldots)$ is a set of N measured quantities, whose distributions are described by a joint probability density function, $f(\underline{x},\underline{\theta})$, where $\underline{\theta} = (\theta_1, \theta_2, \theta_3, \ldots)$ is a set of K parameters. For simplicity, let's focus on one dimensional case where we are have a parameter of interest, θ_0 and

[2] see http://root.cern.ch/root/htmldoc/ROOFIT_ROOSTATS_Index.html

$K-1$ nuisance parameters. The profile likelihood is a function of θ_0 and it is defined as

$$\lambda(\theta_0) = \frac{L(\theta_0, \underline{\hat{\hat{\theta}}}_{i\neq 0})}{L(\hat{\theta}_0, \underline{\hat{\theta}}_{i\neq 0})} \tag{3.2}$$

The denominator, $L(\hat{\theta}_0)$ is the maximum likelihood value, obtained from the best fit to the data, while the numerator is the maximum value obtained by fixing θ_0 and varying the remaining $K-1$ parameters. It can be shown (Wilks's theorem) that asymptotically $-2\ln\lambda(\theta_0)$ is distributed as a $\chi^2(1)$ distribution ($\chi^2(n)$ in the case of n parameter of interest). This is not surprising since in the asymptotic limit the Likelihood function becomes a Gaussian centered around the maximum likelihood estimator $\hat{\theta}_0$, and the profile likelihood curve has a parabolic shape:

$$-2\ln\lambda(\theta_0) = -2(\ln L(\theta_0) - \ln L(\hat{\theta}_0)) = n_\sigma^2, \text{ with } n_\sigma = \frac{\theta_0 - \hat{\theta}_0}{\sigma}. \tag{3.3}$$

where σ represent the Gaussian standard deviations of the parameter θ_0. From this construction, it is possible to obtained the one- or two-sided confidence interval we are interested in (see for example figure 2). Even in case of non parabolic log-likelihood functions, it can be shown, due to the invariance property of the likelihood ratios, that this approach is still valid. This method is also called MINOS in the physics community, since it is implemented by the MINOS algorithm of the Minuit program [6].

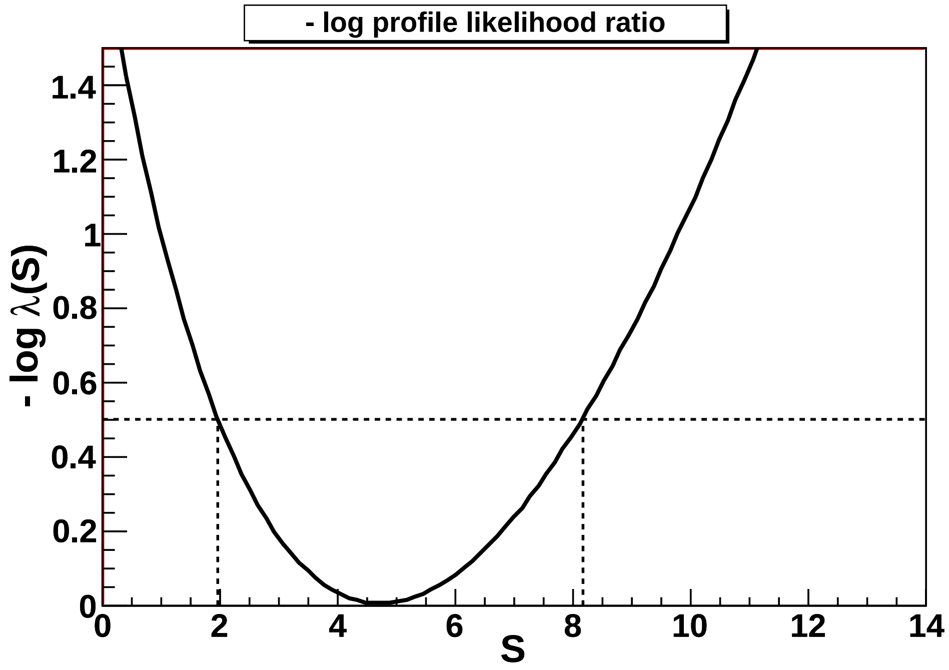

Figure 2: Plot of the log profile likelihood curve ($-\log\lambda$) as function of the parameter of interest, S. The one σ interval (66.8% CL) is obtained from the intersect of the $-\log\lambda$ curve with the horizontal dashed line ($-\log\lambda = 0.5$).

Given the fact that asymptotically $-2\ln\lambda$ is distributed as a χ^2 distribution, an hypothesis tests can also be performed for the null hypothesis H_0, corresponding to $\theta = \theta'$, and the alternate hypothesis H_1 for $\theta \neq \theta'$.

This likelihood-based technique for estimating an interval and performing an hypothesis test is provided in RooStats by the `ProfileLikelihoodCalculator` class. The class implements both the `IntervalCalculator` and the `HypoTestCalculator` interfaces. When estimating the interval, it returns as result a `LikelihoodInterval` object, which, in the case of multi-parameters of interest, represents a multi dimensional contour. When performing the hypothesis test, an `HypoTestResult` is returned with the significance and p-value for the null hypothesis. Another class exists, `LikelihoodIntervalPlot`, to visualize the likelihood interval in the case of one or two dimension (figure 2 shows the obtained plot in one dimensional problem).

3.2 Bayesian Calculators

These calculators are based on the Bayes theorem to estimate the posterior probability distribution for the desired parameters of interest. From the posterior probability distribution the Bayesian credible interval can then be obtained. Input to the calculators are the model and the data sets which are used to build the likelihood function and also the prior distributions of the parameters of interest and of the nuisance parameters, if those are present. These are integrated (marginalized) by each Bayesian calculator to obtained the posterior distribution. RooStats provide two different types of Bayesian calculators, the `BayesianCalculator` class and `MCMCCalculator`, depending on the method used for performing the integration.

The `BayesianCalculator` class works only for one single parameter of interest and it uses analytical or numerical integration to compute the posterior probability. The result of the class is a one dimensional interval (class `SimpleInterval`), which is obtained from the cumulative posterior distribution.

The `MCMCCalculator` uses a Markov-Chain Monte Carlo to perform the integration. It runs the Metropolis-Hastings algorithm to construct the Markov Chain. The class allows to configure the Metropolis-Hastings algorithm, via number of interations and burn-in-steps. Moreover, it is possible to replace the default uniform proposal function with any other distribution function. The result of the `MCMCCalculator` is a `MCMCInterval`, which can compute the confidence interval on a desired parameter of interest from the obtained Markov Chain. `MCMCInterval` integrates the posterior where it is the tallest until it finds the correct cut-off height C to give the target confidence level $1-\alpha$, such that

$$\int_{f(\mathbf{x})\geq C} f(\mathbf{x})\, d^n x = 1-\alpha.$$

To perform this calculation, the posterior can be represented using an histogram in one or multi-dimensions or using a kernel estimator. The class `MCMCIntervalPlot` class can be used to visualize the interval and the Markov chain.

Users can input the RooStats model also to the Bayesian Analysis Toolkit (BAT) [7], a different software package implementing Bayesian methods with the Markov-Chain Monte Carlo. In its latest release, BAT provides a class, `BATCalculator` which can be used with a similar interface to the RooStats `MCMCCalculator` .

3.3 Neyman Construction

The Neyman construction is a pure frequentist method to construct an interval at a given confidence level value, $1-\alpha$, in order that if we repeat the experiment many times the interval will contain the true value a fraction $1-\alpha$ of the time. A detailed description of the method is given in [5]. RooStats provides a class, `NeymanConstruction` implementing this technique. The class derives from `IntervalCalculator` and returns as result a `PointSetInterval`, a concrete implementation of `ConfInterval`.

The Neyman Construction is not a uniquely defined statistical technique, it requires that one specifies an ordering rule or ordering principle, which is usually encoded by choosing a specific test statistic and limits of integration (corresponding to upper/lower/central limits). As a result, the RooStats class must be configured with the corresponding information before it can produce an interval. These have been generalized with the interfaces `TestStatistic`, `TestStatSampler`, and `SamplingDistribution`. Common configurations, such as the unified Feldman-Cousins approach, where the ordering is based on the profile likelihood ratio as test statistics and it is described in their paper [8], can be enforced by using the `FeldmanCousins` class.

The Neyman construction considers every point in the parameter space independently, no assumptions are made that the interval is connected or of a particular shape. The result consists then of a set of scanned points with the information if they are inside or outside the interval (`PointSetInterval` class). The user indicates which points in the parameter space to perform the construction or provides a range with the number of points which will be scanned uniformly in a grid. For each scanned point, the calculator will obtain the sampling distribution of the chosen test statistics. This is typically obtained by toy Monte Carlo sampling, but other techniques exist and they can in principle be used, although there are not yet implemented in RooStats. The tool also implements a generalization of the Feldman-Cousins procedure when nuisance parameters are present as described in [1, 9].

3.4 Hybrid Calculator

This calculator implements an hybrid approach for hypothesis testing. It consists of a traditional frequentist toy Monte Carlo method, like in the Neyman construction, but with a Bayesian marginalization of nuisance parameters [10]. Hence the technique is often referred to as a "Bayesian-Frequentist Hybrid". For example, we define the null hypothesis, H_b, that no signal is present over the background and H_{sb} the alternate hypothesis that signal is also present. In order to quantify the degree to which each hypothesis is favored or excluded by the experimental observation, one chooses a test-statistics which ranks the possible experimental outcomes. A commonly used test statistics is the ratio of the likelihood function in both hypotheses: $Q = L_{sb}/L_b$ and the quantity $-2\ln Q$ may also be used instead of Q. Alternative choices of test statistics can be the number of events or the profiled likelihood ratio (see equation 3.2).

From the comparison of the test statistics value of Q, observed in the data (Q_{obs}), to the probability distribution of Q (dP/dQ) expected for the b and $s+b$ hypotheses, the confidence levels, CL_{sb} and CL_b, can be computed. Since the functional forms of the test statistic distributions dP_{sb}/dQ and dP_b/dQ are not always known a priori, a large amount of toy Monte Carlo experiments are performed. Two types of pseudo datasets are generated: with signal plus background and with the background-only hypothesis. Figure 3 provides an example of the obtained distributions from the two pseudo data sets and how they compare with the observed test statistic value in the data.

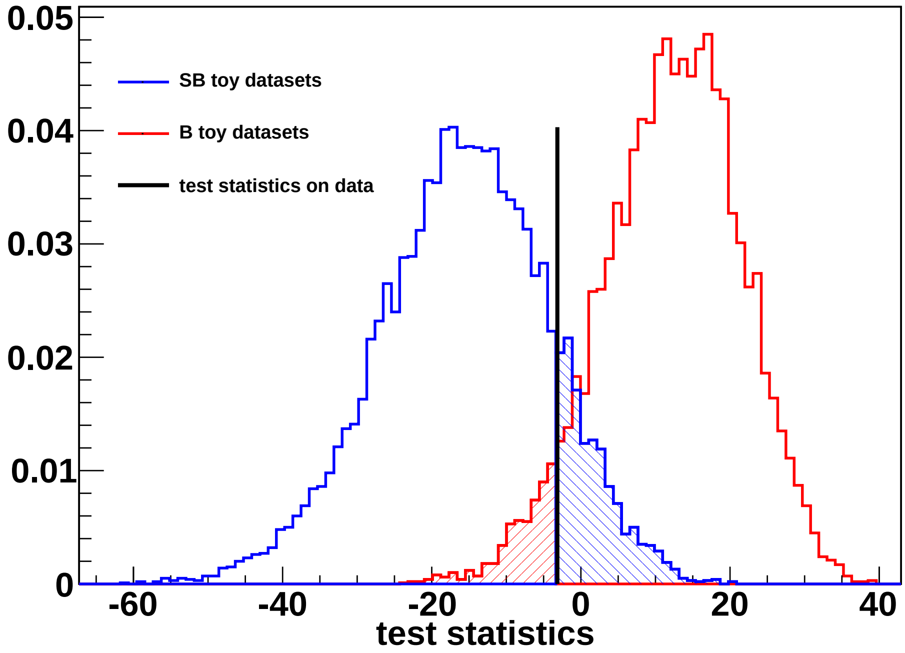

Figure 3: Result from the hybrid calculator, the distributions of $-2\ln Q$ in the background-only (red, on the right) and signal+background (blue, on the left) hypotheses. The black line represents the value $-2\ln Q_{obs}$ on the tested data. The shaded areas represent $1-CL_b$ (red) and CL_{sb} (blue).

Systematics uncertainties are taken into account using Bayesian Monte Carlo sampling. For each toy Monte Carlo experiment, the effective value of the nuisance parameters is varied before generating the toy sample. The whole phase space of the nuisance parameters is thus sampled through Monte Carlo integration. The final net effect consists in a broadening of the test statistic distribution and thus, as expected in presence of systematic uncertainties, a degraded separation of the hypotheses.

All this is implemented in Roostats by the `HybridCalculator` class. Input to the class are the model for the null hypothesis ($s+b$) and the model for the alternate (b only) hypothesis, the data sets and optionally the prior distribution for the nuisance parameters which is sampled in the toy generation process. The `HybridCalculator` class provides various choices for the test statistics, such as the number of events, the likelihood ratio or the profiled likelihood ratios. The results of the `HybridCalculator` consists of the test statistic distribution for the two hypothesis, from which the hypothesis p-value and significance can be obtained. Different results can be merged together in order to be able to run the calculator in distributed computing processes. The `HybridPlot` class provides a way of plotting the result, as shown for example in figure 3.

By varying the parameter of interest representing the hypothesis being tested (for example by varying the signal cross-section) one can obtain a one sided confidence interval (i.e. an exclusion limit). RooStats provides a class, `HypoTestInverter`, which implements the interface `IntervalCalculator` and performs the scanning of the hypothesis test results of the hybrid calculator for various values of one parameter of interest. By looking where the confidence level curve of the result intersects the desired confidence level, an upper limit can be derived, assuming the interval is connected.

4. Combination of Results

Combining results from multiple experiments in order to enhance sensitivity of a measurement or improve the power of a hypothesis test is common. The challenge of combining results is primarily logistical, since a proper combination requires low-level information from each experiment be brought together to form one large statistical test. Again, this is hindered by the fact that the ingredients to the combination are heterogeneous (eg. different formats, technologies, and conventions). A major advancement that was made by the RooStats project is the concept of the workspace. The power of the workspace is that it allows one to save data and an arbitrarily complicated model to disk in a ROOT file. These files can then be shared or archived, and they provide all the low-level ingredients necessary for a proper combination in a unified framework. A direct advantage of this is a digital publishing of the results.

The `RooWorkspace` class of RooFit provides the low-level functionality for storing the full model and the data and, in addition, it provides a convenient functionality to create easily the model via a string interface (workspace factory).

5. RooStats Utilities

In addition to the previously described methods for estimating confidence intervals and performing hypothesis tests, RooStats provides also some statistical utilities. These include functions to calculate the p-value or Z-value (e.g. significance in one-sided Gaussian standard deviations) for a number counting experiments.

Another utility is the class `RooStats::SPlot`, which implements the $_s\mathcal{P}lot$ technique [11] to disentangle signal from background via an extended maximum likelihood fit and with a tool to access the quality and validity of the fit by producing distributions for the control variables. `RooStats::SPlot` complements the ROOT `TSPlot`[3] class with the possibility to use arbitrary models created with RooFit.

The `BernsteinCorrection` is another utility class to augment a nominal probability density function with a polynomial correction term. This is useful for incorporating systematic variations to the nominal probability density function. The Bernstein basis polynomials are particularly appropriate because they are positive definite.

The `HLFactory` is a user friendly wrapper around the RooFit workspace class to build the input model for RooStats from a text file. Other high level factory tools are planned and will available in future versions of RooStats.

[3] see http://root.cern.ch/root/html/TSPlot.html

6. Conclusions

RooStats is a new software package providing the major statistical tools required for the LHC data analysis. These tools are implemented by using common interfaces, so they can be re-used with the same input model and they can be applied both in simple number counting problems and in complex ones, with multi parameter of interests and various nuisance parameters.

RooStats is distributed together with ROOT and its latest production version, released in December 2009, is 5.26.00. Examples of usage are provided in the `tutorials/roostats` directory of the ROOT distribution as ROOT macros. There are macros as example for each of the RooStats calculator. In addition to numbers (interval limits, significance, etc..) they produce also statistical plots of the result (confidence level contours or test statistics distributions like in figure 3). The examples are based on models with Poisson statistics (number counting experiments) or Gaussian signal model with flat background and they take into account systematic uncertainties in the nuisance parameters. Furthermore, a tutorial of the `FeldmanCousins` class is provided, which uses the same model of the neutrino oscillation search described in [8].

RooStats is starting now to be used by both ATLAS and CMS for producing the results of their data analysis and it is then planned to be used in the future for their analysis combinations.

Acknowledgements

We are thankful to the member of the ATLAS and CMS statistics committees for the exchange of ideas, advice and encouragement. We also wish to thank R. Cousins, G. Piacquadio, M. Pierini, G. H. Lewis and M. Pelliccioni which have contributed to the developments of RooStats.

References

[1] K. S. Cranmer, *Statistics for the LHC: Progress, Challenges and Future*, proceedings of the PHYSTAT LHC Workshop, CERN 27-29 June 2007, 47.

[2] D. Piparo, G. Schott and G. Quast, *RooStatsCms:: a tool for analysis modelling, combination and statistical studies*, J. Phys.: Conf. Ser. 219 032034 (2010)

[3] I. Antcheva et al, *ROOT - A C++ framework for petabyte data storage, statistical analysis and visualization*, Comput. Phys. Comm. 180 (2009) 2499.

[4] W. Verkerke, *Statistical Software for the LHC*, proceedings of the PHYSTAT LHC Workshop, CERN 27-29 June 2007, 169.

[5] F. James, *Statistical Methods in Experimental Physics 2nd Edition*, Word Scientific 2006.

[6] F. James, *MINUIT Reference Manual*, CERN Program Library Writeup D506.

[7] A.Caldwell, D. Kollar, K. Kröninger, "BAT - The Bayesian Analysis Toolkit", arXiv:0808.2552.

[8] G. Feldman and R. D. Cousins, *Unified approach to the classical statistical analysis of small signals* Phys. Rev. D 57, 3873 (1998)

[9] K. S. Cranmer, *Frequentist hypothesis testing with background uncertainty*, procedings of PhyStat 2003, SLAC (2003) [physics/0310108].

[10] R.D. Cousins and V.L. Highland, “Incorporating Systematic Uncertainties into an Upper Limit”, Nucl. Instrum. Meth. A320 (1992) 331.

[11] M. Pivk and F. R. Le Diberder, Nucl. Inst. Meth.A 555, 356 (2005)