Collie: A Confidence Level Limit Evaluator

Version 4.00

Wade Fisher
Fermilab
(Dated: February 16, 2010)

This note describes the algorithms used within the COLLIE software package used to construct confidence levels and evaluate exclusion limits. Following the discussion of algorithms and definitions, a discussion of diagnostic tests available for assessing the quality of fits performed by COLLIE's fitting method is presented.

Contents

I. INTRODUCTION

A typical high-energy new physics search analysis is ultimately described by a final variable (or variables) chosen to be sensitive to a parameter of the search system. The result of the search are distributions of this final variable for the new physics process, one or more modeled background processes, and the observation from data. These final variable distributions become the input to statistical calculations.

In general, the final variable distributions are designed to describe two distinct hypotheses that will be compared to data. The first hypothesis is intended to describe a specified new physics process (henceforth signal, using HEP jargon) in addition to the predicted background processes that are expected to comprise the majority of the data sample. This is referred to as the TEST or signal-plus-background (S+B) hypothesis. The second hypothesis is a subset of the first obtained by removing the signal model and is referred to as the NULL or background-only hypothesis. Both are compound hypotheses and depend on a set of parameters that affect the final variable, but are not of immediate interest. Examples of such parameters are integrated luminosities, efficiencies, acceptances, and background cross sections. Referred to as *nuisance parameters*, the values of these ingredients are important in the extraction of limits on the parameter of interest in the new physics model, and any uncertainty in these nuisance parameters will generally degrade the sensitivity of the search to the parameter of interest.

Given a properly described model, an analyzer may be interested in determining statistical quantities that can be derived from a comparison of the model to data. The COLLIE software suite is designed to calculate statistical quantities including p-values for signal and background processes (See Sec IV), limits on model parameters, and measurements of cross sections. This note describes the algorithms within the COLLIE package, the derivation of statistical quantities, the proper usage of COLLIE's application program interface, and the rare circumstances in which the results from COLLIE may be made unreliable due to incorrect usage.

I.A. Brief Primer

To make the following text more accessible, there are a few ideas and terms that should first be defined and discussed. This section is intended as a refresher for reader. This list is by no means comprehensive, but meant as a small primer on the concepts involved in hypothesis testing. This text assumes the reader is familiar with probability and statistics on a basic level consistent with the reviews of these two topics found in [1].

- Null Hypothesis: The NULL hypothesis represents a model that is either believed to be true or is used as the basis (control) of a test. For example, in a search for new physics the NULL hypothesis would represent the model in which there is no new physics. This is also commonly referred to as the background-only (B-Only) hypothesis.

- TestHypothesis: The TEST hypothesis is the alternative model that has been established for testing against the NULL hypothesis. For example, in a search for new physics the TEST hypothesis represents the model in which there is new physics that has a distinguishable effect. This is also commonly referred to as the signal-plus-background (S+B) hypothesis.

- Final Variable: The term final variable refers to an observable related to the data that is used to quantify the final results of a search. This final variable is desired to have qualities that enable the effective identification of two or more classes of event types. The efficiency of this classification ultimately determines the ability of a search to distinguish hypotheses.

- Test Statistic: A test statistic is quantity derived from a data sample and used to quantify the degree to which the data are consistent with the TEST and NULL hypotheses. Also referred to as an "Ordering Rule", the test statistic is used to order the outcomes of individual datum relative to one another in the two hypotheses. Though the term test statistic is also commonly (and correctly) used to describe a final variable, in this document the nomenclature of test statistic will solely refer to the statistical test used to compare data to the TEST and NULL hypotheses.

- Simple Hypothesis: A simple hypothesis is a fully specified model, that is, one in which every parameter is given a definitive value. Typically, these values are the true values of the parameters. For example, a Gaussian distribution in which the mean and width are both specified is a simple hypothesis.

- Compound Hypothesis: A compound or composite hypothesis is a model that does not fully specify the probability distribution function. For example, a Gaussian distribution in which only the mean is specified is a

compound hypothesis. The uncertainty to which the remaining parameters are unknown can be parameterized in any manner appropriate to the hypothesis.

- Parameter of Interest: In a hypothesis test, the parameter of interest is the model parameter that specifies the difference between the TEST and NULL hypothesis. For example, in a model describing the number of events counted in a particle physics experiment ($N = L \times \epsilon \times \sigma$) either the luminosity L, efficiency ϵ, or cross section σ could be the parameter of interest, but not all simultaneously. Each parameter (L, ϵ, or σ) can be used to specify a class of unique models, but a specific model must be given by fixed values of the remaining parameters.

- Nuisance Parameter: A nuisance parameter is a parameter of a model (or hypothesis) that is unspecified but not of immediate interest to the test. For example, in a model describing the number of events counted in a particle physics experiment ($N = L \times \epsilon \times \sigma$) either the luminosity L, efficiency ϵ, or cross section σ could be the parameter of interest, and the remaining parameters are nuisance parameters. If all nuisance parameters have zero uncertainty for their true values, the model can be presented as a simple hypothesis. If any parameter has non-zero uncertainty for its true value, the model is a compound hypothesis. As noted above, any uncertainty on the parameter of interest can be specified as appropriate to the model.

- Conditional Probability: Conditional probability is the likelihood of an outcome (X) assuming the occurrence of another outcome (Y) (the probability of X given Y). This probability can be written as $P(X|Y)$.

- Joint Probability: Joint probability is the simultaneous probability of two specific outcomes.

- Marginal Probability: Marginal probability is the unconditional likelihood of the outcome X regardless of the outcome Y: $P(X)$. If the value Y can be specified as the probability of a random variable Z having a given outcome, the marginal probability can be obtained by summing the joint probabilities over all outcomes of Z. This process is referred to as marginalization and can be viewed as an integration of the conditional probability $P(X|Y)$ over the probability function for the variable Y, $p(Y)$:

$$P(X) = \int P(X|Y)\, p(Y)\, dY \tag{1}$$

- Prior Probability: Prior probability describes the likelihood of a parameter in the absence of a definitive measurement of the true value of the parameter. A prior probability (or just prior) is a marginal probability. In the previous example, the probability $p(Y)$ describes a prior for the parameter Y.

- Confidence Interval: A confidence interval is an interval in the space of the parameter tha tis associated with a confidence level. The interpretation of the confidence level, however, depends on how probability is interpreted: relative frequency, or degree of belief.

- Confidence Level: In a Frequentist interpretation, a confidence level is a guaranteed lower bound on the fraction of intervals that contain the true value of their associated parameter. In a Bayesian approach, it is the probability, interpreted as a degree of belief, that the parameter lies within the given interval.

  For example, if a measurement of the luminosity of a data sample is found to have an uncertainty, it is commonly described by the region enclosing 68.3% (1 σ) of the possible values of the true parameter: $L = 1000 \pm 60\text{pb}^{-1}$. In this example, the confidence interval is $940 \leq L \leq 1060$ and the confidence level is 68.3%.

- p-value: A p-value defines the probability that a test would find a result more extreme than the observed result based on a purely random sampling (i.e., $P(x \geq x_{obs})$).

In the following, the above definitions will be used to generate a picture of the hypothesis testing procedure used in the COLLIE package. As discussed above, typical high-energy physics search analyses can be described by the numbers of events observed in data and those expected in the TEST and NULL hypotheses. Histogramming data is equivalent to separating events into semi-correlated bins, so we will briefly ignore that complication and focus on a single-bin analysis. To further simplify things, we'll assume that the NULL hypothesis can be defined by a single background class. Assuming that the TEST hypothesis is specified by an increase in the expected number of events relative to the NULL hypothesis, the two hypotheses can be written as follows:

$$\text{NULL} : N_{evts}^{Null}(\phi_B) = L \times \epsilon_B(\phi_B) \times \sigma_B \tag{2}$$

$$\text{TEST} : N_{evts}^{Test}(\phi_B, \phi_S) = L \times \epsilon_B(\phi_B) \times \sigma_B + L \times \epsilon_S(\phi_S) \times \sigma_S \tag{3}$$

in which L represents luminosity, ϵ_B and ϵ_S represent the efficiency to select background and signal events, ϕ_B and ϕ_S represent the uncertainties on the efficiency for selecting background and signal events, and σ_B and σ_S represent background and signal cross sections. Here, both the TEST and NULL hypotheses are compound hypotheses. Furthermore, the TEST hypothesis is equivalent to the NULL hypotheses with the inclusion of a parameter of interest (in this example, the signal cross section σ_S) and signal-specific nuisance parameters. The luminosity, background cross section, and selection efficiencies are nuisance parameters. The efficiencies have an uncertainty associated with their values.

By specifying a test statistic Γ of some form (the exact form is not relevant, but let's assume $\Gamma(D) = D \times \ln(1 + N_{evts}^{Signal}/N_{evts}^{Background})$ where D is the number of data events), we can evaluate three different test statistic values: the value for the observed data ($D = N_{evts}^{obs}$), the value if the data were equal to the nominal background-only expectation ($D = N_{evts}^{Null}$), and the value if the data were equal to the nominal signal-plus-background expectation ($D = N_{evts}^{Test}$). The second two values are insufficient to determine confidence intervals and levels, as they represent only two possible outcomes with which to compare the data. Because they represent point probabilities, it is unlikely that the observed test statistic will be exactly equal to either value. In order to derive confidence intervals, we must first describe the probability distribution function (PDF) of the test statistic for the TEST hypothesis and the NULL hypothesis.

To describe the test statistic PDF for a given hypothesis, we make the assumption that the data we observe are one possibility sampled randomly from a Poisson distribution with a mean value given by the expected number of events for the hypothesis in question. This Poisson distribution describes the probability for different values of D in $\Gamma(D)$ and thus defines the PDF for Γ in the hypothesis in question. If no nuisance parameters had uncertainties, our hypothesis would be simple and the PDF would be fully specified. In other words, if we could specify a fixed value for each uncertain nuisance parameter, we could define the conditional probability of Γ for that fixed set of nuisance parameter values. In the case that any nuisance parameter is uncertain, we must attempt to describe the PDF in terms of our lack of knowledge by marginalizing our Poisson PDF. The marginalization is performed by assuming different values for the selection efficiency nuisance parameters, which in turn changes the predicted numbers of signal and background events used as the mean value of the Poisson. This marginalization procedure can be interpreted as constructing a superposition of many simple hypotheses, each defined by a choice of nuisance parameter values. With a pre-specified distribution of possible values for the selection efficiency (*i.e.*, a prior), we can define the marginal PDF by integration or Monte Carlo sampling. Integration algorithms can be computationally expensive, so the COLLIE package marginalizes the Poisson PDF by randomly sampling nuisance parameter values from their pre-specified priors. This procedure specifies the distribution of possible outcomes for a hypothesis, which is also commonly referred to as the prior predictive ensemble for the hypothesis.

For each outcome in the prior predictive ensemble, the test statistic Γ can be evaluated which defines in turn the Γ PDF ($P(\Gamma)$) for this ensemble. For a pre-defined confidence level α_0, a confidence interval in χ can be defined by integration:

$$1 - \alpha_0 = \int_{\Gamma_0}^{\Gamma_1} P(\Gamma)\, \mathrm{d}\Gamma \tag{4}$$

wherein Γ_0 and Γ_1 define the confidence interval satisfying the condition. Alternatively, by using a pair of reference test statistic (Γ_{ref0}, Γ_{ref1}), the confidence level can be specified in reverse fashion:

$$1 - \alpha_{\mathrm{ref}} = \int_{\Gamma_{\mathrm{ref0}}}^{\Gamma_{\mathrm{ref1}}} P(\Gamma)\, \mathrm{d}\Gamma \tag{5}$$

in which α_{ref} is the resulting confidence level for this choice of confidence interval. In the context of a cross section limit, a lower bound is generally defined by a value of zero (i.e., the cross section for new physics is zero). In this manner, the confidence levels for the TEST and NULL hypotheses can be evaluated for any value of Γ_{ref}. The three most common values are the observed value of Γ, the median value of the NULL hypothesis Γ PDF, and the median value of the TEST hypothesis Γ PDF. These confidence levels correspond to a specific value of the parameter of interest (e.g., signal cross section) used to generate the prior predictive ensembles and to define the value of Γ. Thus, a specific value of the signal cross section could be chosen to satisfy a desired confidence level (say, 95%) for any reference value of Γ.

This brief example describes the basics of the calculations performed in the COLLIE package. The construction and use of these confidence intervals is discussed in more precise detail in the following chapters.

II. GENERATION OF PSEUDO-DATA

In this note, I will refrain from engaging the argument between Frequentist and Bayesian statistics, as this choice has already been widely debated: see Refs. [2–4] for examples. COLLIE adopts a semi-Frequentist construction for the estimation of the likelihood distributions associated with a comparison of the TEST and NULL hypotheses. These probability distribution functions (PDFs) are generated numerically via the distribution of a pre-specified test statistic evaluated in a large array of data events generated via artificial pseudo-experiments (pseudo-data).

The pseudo-data model assumes that the data collected by an experiment is stochastically sampled from a Poisson parent distribution. Thus, the event rates observed are the coherent sum of contributing physical processes (*i.e.*, signal and background processes) each and can be simulated via random Poisson trial. COLLIE requires input signal, background, and data distributions to be binned and the number of pseudo-data events in each bin is determined via uncorrelated random Poisson trials per bin. Each Poisson trial is seeded with a mean value taken from the sum of the contributing processes in that bin. The effect of uncertainties on nuisance parameters is incorporated by re-deriving the nominal signal and background predictions before throwing Poisson trials, as described below. In this way, the results of repeated data-collection processes is simulated. Individual sets of pseudo-data are generated for the TEST and NULL hypotheses independently. The distributions of the test statistic evaluated for these sets of pseudo-data are referred to as the prior predictive ensembles for the TEST and NULL hypotheses.

II.A. Treatment of Uncertainties

Purely Frequentist prescriptions for incorporating uncertainties on nuisance parameters exist [5], but are currently algorithmically limited due to a large computational overhead. The model chosen within COLLIE is a numerical implementation of the Cousins-Highland method [6]. This choice is inherently a Bayesian approach, and is a departure from a purely Frequentist treatment. The Cousins-Highland approach is a simple smearing of likelihoods on the nuisance parameter in question. This can be viewed analytically as a transformation of a compound hypothesis with a nuisance parameter η (*i.e.*, $P(x|H,\eta)$) to a superposition of simple hypotheses $p(x|H)$ via integration:

$$p(x|H) = \int P(x|H,\eta)\ \pi(\eta)\ \mathrm{d}\eta \tag{6}$$

where x denotes a given set of data being interpreted in the hypothesis H and $\pi(\eta)$ is a prior density for η and is typically a Gaussian distribution. The numerical implementation proceeds by sampling $p(x|H)$ and is described below. Given a choice of test statistic used to order outcomes, the sampled PDF of $p(x|H)$ is referred to as the prior predictive ensemble for hypothesis H.

COLLIE models systematic uncertainties using a prior PDF specified by $\pm1\sigma$ deviations on the nuisance parameter in question. This parameterization makes the implicit assumption that the prior PDF is approximately Gaussian (e.g., $2\sigma \simeq 2{\times}1\sigma$). Analyzers can propagate the $\pm1\sigma$ deviations through the entire event selection and analysis process, and the impact on the final variable can be determined. COLLIE allows the prior distribution for each uncertainty to be parameterized using either a Gaussian or log-normal distribution. The log-normal PDF is obtained following the derivation in [7]:

$$\sigma_{LN} = \frac{\sigma_G}{\mu_G} \tag{7}$$

$$\mathrm{mode}_{LN} = \ln(\mu_G) + \left(\frac{\sigma_G}{\mu_G}\right)^2 \tag{8}$$

where σ_{LN} and $mode_{LN}$ represent the RMS and mode of the log-normal distribution, respectively, and σ_G and μ_G represent the RMS and mean of the Gaussian distribution, respectively. Random values can be drawn from this distribution via an extension to the Box-Muller method for generating normally-distributed random values [8]. Given a random number drawn from a Gaussian distribution with $\mu_G = 0$ and width σ_G (R_G), a log-normal can be obtained via:

$$R_{LN} = \exp\left(\mathrm{mode}_{LN} + \sigma_{LN}\frac{R_G}{\sigma_G}\right) \tag{9}$$

which will match the mode of Gaussian and log-normal distributions. In order to ensure linearity, the COLLIE package adopts an alternative formulation which results in a matching of the mean of Gaussian and log-normal distributions:

$$R_{LN} = \exp\left(\sigma_{LN}\frac{R_G}{\sigma_G}\right) \tag{10}$$

The method for incorporating nuisance parameter uncertainties is as follows. The number of predicted events in each bin (p_i) is the sum of any contributing sources of events. Each contributing source of events is a function of a set of k nuisance parameters ($\vec{\eta_k}$). The nominal prediction (p_i^0) is defined by the central values of the nuisance parameters $\vec{\eta_i^0}$:

$$p_i(\vec{\eta_k}) = \sum_{j=1}^{N^{\rm S}} p_{ij}(\eta_1, \cdots, \eta_k) \tag{11}$$

$$p_i^0(\vec{\eta_k}) = \sum_{j=1}^{N^{\rm S}} p_{ij}(\eta_1^0, \cdots, \eta_k^0) \tag{12}$$

where the index j runs over the number of contributing event sources N^S. Changes in the values of the nuisance parameters can be described as departures from the nominally predicted number of events:

$$p_i(\vec{\eta_k}) = \sum_{j=1}^{N^{\rm S}} p_{ij}(\vec{\eta_k^0}) \prod_{k=1}^{N^{\rm Par}} \frac{p_{ij}(\eta_k)}{p_{ij}(\eta_k^0)} \tag{13}$$

$$= \sum_{j=1}^{N^{\rm S}} p_{ij}^0 \prod_{k=1}^{N^{\rm Par}} \frac{p_{ij}(\eta_k^0) + \frac{\partial p_{ij}(\eta_k)}{\partial \eta_k}(\eta_k - \eta_k^0)}{p_{ij}(\eta_k^0)} \tag{14}$$

$$= \sum_{j=1}^{N^{\rm S}} p_{ij}^0 \prod_{k=1}^{N^{\rm Par}} 1 + R_k\ \sigma_k\ \frac{\partial p_{ij}(\eta_k)/\partial \eta_k}{p_{ij}(\eta_k^0)} \tag{15}$$

$$= \sum_{j=1}^{N^{\rm S}} p_{ij}^0 \prod_{k=1}^{N^{\rm Par}} 1 + R_k\ \sigma_{ijk} \tag{16}$$

$$= p_i(\vec{R_k}) \tag{17}$$

wherein the index k runs over the number of nuisance parameters in the model, $\sigma_{ijk} = \sigma_k \cdot \frac{\partial p_{ij}(\eta_k)/\partial \eta_k}{p_{ij}(\eta_k^0)}$ defines the fractional change in the number of events for the specified nuisance parameter k for the specified event source j, and $R_k = (\eta_k - \eta_k^0)/\sigma_k$ represents the deviation from the central value of the nuisance parameter in units of a characteristic deviation σ_k. In a system of units in which all central values of nuisance parameters are taken as zero and characteristic deviations are given as ± 1 standard deviation distributions, the R_k values can be described as a unit prior PDF with a mean value of zero. This formulation explicitly ignores the higher order terms in the series expansion, which is simply a statement that the predicted numbers of events depends linearly on each nuisance parameter.

For each pseudo-experiment, the value of each nuisance parameter (*i.e.*, the R_k value) is randomly drawn from its specified prior PDF. For a given set of k fluctuated nuisance parameters, the predicted mean values used as the Poisson mean in the random Poisson MC trials are re-derived as demonstrated in Eqn. 17 for a single bin of a histogrammed distribution. In practice, the prior PDF for a given nuisance parameter can be chose as Gaussian or log-normal. COLLIE also allows for cases in which the positive and negative deviations are asymmetric. In these cases, a purely linear treatment (*e.g.*, allowing different positive and negative Gaussian deviations) leads to a discontinuous PDF with an infinite derivative (*i.e.*, a dimidiated Gaussian). Following [9], this problem can be alleviated by parameterizing the efficiency deviations quadratically as shown in Eqn 18. This formulation is, admittedly, not rigorous. However, it is a consistent mathematical model that provides a natural and satisfactory solution to the problem. A comparison of the dimidiated Gaussian and the quadratic matching function can be seen in Fig. 1. For symmetric uncertainties, this equation reduces to Eqn 17.

$$p_{ij}(\vec{R_k}) = p_{ij}^0 \prod_{k=1}^{N^{\rm Par}} 1.0 + R_k \left(\frac{\sigma_{ijk}^+ + \sigma_{ijk}^-}{2}\right) + R_k^2 \left(\frac{\sigma_{ijk}^+ - \sigma_{ijk}^-}{2}\right) \tag{18}$$

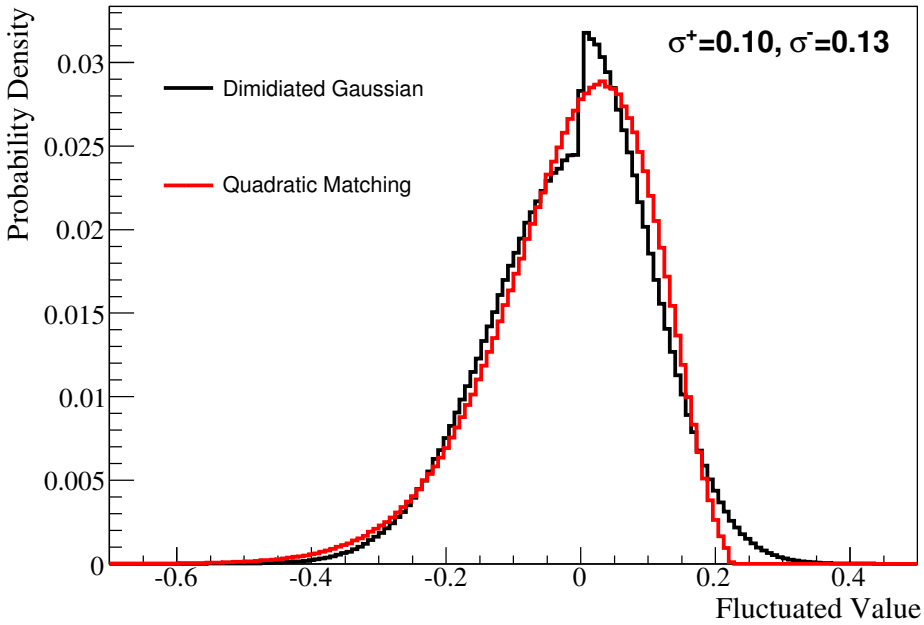


FIG. 1: A comparison of dimidiated Gaussian and quadratically-matched PDFs.

In this model, the random fluctuations R_k are designed to be referenced from the nominal efficiency. For example, in the case of the Gaussian parameterization the fluctuation can be understood as drawing a random number from a Gaussian with mean value 0.0 and a width of 1.0. In this way, the cumulative result of smearing the k nuisance parameters is a multiplicative propagation of efficiency-like terms. For the Gaussian parameterization, the priors are forced to be positive-definite to avoid non-physical fluctuation values. This Gaussian truncation is performed by re-throwing trials which result in negative values, effectively introducing skewness to the Gaussian prior and resulting in a mean and RMS different from the specified values. The impact of this truncation becomes non-trivial for uncertainty sizes of roughly 30%. For such uncertainties, it is advisable to consider the log-normal prior and the COLLIE package prompts users to do so.

Due to the multiplicative model for the accumulation of fluctuated uncertainties, the prediction for the number of events in a given bin can be zero if any of the fluctuated uncertainties is zero. This is a consequence of Eqn. 17, rather than the truncation of prior PDFs. To avoid singularities in the calculation of test statistics, background fluctuations are given a lower bound of 1×10^{-8} of the nominal prediction. Bins predicted to have zero background events are left at zero. In cases in which the nominal background prediction is zero and the signal or data predictions are non-zero, singularities are avoided by summing the zero-background bin to the nearest bin with the highest signal to background ratio. The COLLIE package alerts users that they have bins with zero background and non-zero signal. In such instances, users are advised to determine whether or not a zero-background prediction is appropriate. If the analysis does not warrant a truly background-free region, users should re-evaluate their background model. Techniques such as smoothing can be used to address fluctuations due to limited statistics in simulated events.

II.A.1. Correlated Uncertainties

Nearly all analyses will contain nuisance parameters which are common to more than one background or signal source and also amongst combined channels. These correlated uncertainties must be handled properly to ensure the model used to generate the prior predictive ensembles is correct. For example, the uncertainty on a measurement of the luminosity is correlated between channels as well as between signal and background sources. As each uncertainty is sampled from a Gaussian distribution for each Poisson trial of the calculation, Eqn.17, it is sufficient to ensure that each correlated uncertainty is randomly sampled only once per iteration. Using the previous example, one must ensure that the sampled uncertainty value for luminosity is the same for all affected event sources for each Poisson trial, although it will change between trials. In this way, each correlated uncertainty is ensured to have the same fluctuation for all event sources and channels. To incorporate fractional correlation coefficients within a signal or background source, a matrix of correlation coefficients can be introduced to Eqn.17, though this case is not yet incorporated into

COLLIE. However, anti-correlated systematics can be simulated by introducing a systematic model which manifests in oppositely signed fluctuations for different background or signal sources. This option is described in Section XI.

II.A.2. Flat Uncertainties

Many nuisance parameter uncertainties affect all bins of a final variable uniformly. For example, the uncertainty on luminosity is expected to only impact the global normalization of the predicted numbers of events. For these types of uncertainties, COLLIE allows a "flat" parameterization to be specified giving the positive and negative fractional uncertainties:

$$\Delta_i^+ = \frac{N(+1\sigma) - N(0\sigma)}{N(0\sigma)} \tag{19}$$

$$\Delta_i^- = \frac{N(0\sigma) - N(-1\sigma)}{N(0\sigma)} \tag{20}$$

In this manner, the positive (Δ_i^+) and negative (Δ_i^-) fractional deviations are given as the absolute value of the deviation (e.g., $\Delta_i^+ = 0.06$ and $\Delta_i^- = 0.06$ for $\pm 6\%$). The remaining sign conversions are handled internally.

II.A.3. Uncertainties Impacting Shapes

In cases where a binned final variable has been chosen (i.e., more than one bin), the relative shapes of the signal and background distributions are relied upon to increase the sensitivity of an analysis. However, when considering systematic uncertainties, the certainty of this shape can be called into question. In general, scenarios in which signal and background shapes have large shape uncertainties relative to each other are poor choices for a multi-binned analysis. Nonetheless, such uncertainties require a measurement of how they impact the prior predictive ensemble. There is no simple way to calculate how uncertainties on variable shapes impact sensitivity calculations. In many instances, the most straightforward way is to perform the calculation for several instances of smeared final variable distributions.

However, COLLIE allows users to specify uncertainties that vary from bin to bin for each input distribution. These values are intended to correspond to the $\pm 1\sigma$ changes in final variable rate and shape obtained by propagating an uncertainty through the analysis chain. The positive and negative fluctuations are specified separately by the user and can be given as fractional per-bin uncertainties or appropriately-normalized alternative final variable shapes per event source.

II.A.4. Statistical Uncertainties

The statistical uncertainty associated with the number of Monte Carlo events per histogram bin must be accounted for in the calculation of the prior predictive ensemble. This is done performing an uncorrelated Gaussian fluctuation for expectation in each bin of each event source. The width of the Gaussian prior is set by the per-bin uncertainties. If users have correctly specified the per-bin statistical uncertainties in their input histograms, COLLIE can use these directly in the calculation. If not, COLLIE will ignore the per-bin uncertainties and expect the statistical uncertainty to be entered as a systematic uncertainty labeled "Statistics".

III. CHOICE OF TEST STATISTIC

Given properly-generated sets of pseudo-data for both the TEST and NULL hypotheses, the next step is to evaluate each set using a pre-specified test statistic to formulate the prior predictive ensemble. In general, the test statistic (or statistical test) is constructed to maximize both signal density and the isolation of signal from background. This procedure is intended to provide an intrinsic discrimination between signal-like outcomes and background-like outcomes. Following the Neyman-Pearson lemma, one should be able to define regions containing the final variable (x) in which the ratio of PDFs for the TEST and NULL hypotheses satisfies a minimum criteria chosen to achieve a desired signal significance, as indicated in Eqn. 21.

$$Q(x) = \frac{f(x \mid \textsc{Test})}{f(x \mid \textsc{Null})} \tag{21}$$

One may understand this statement by considering Eqn. 21 to represent the criterion with which one may obtain the highest signal purity for a given signal efficiency. In practice, the joint PDFs $f(x \mid \textsc{Test})$ and $f(x \mid \textsc{Null})$ are not explicitly known and are evaluated using a classification scheme that generates some analysis-dependent final variable.

Given an appropriately chosen classifier specifying signal and background PDFs, one may proceed to construct a test statistic. The likelihood-ratio test statistic given in Eqn. 21 can be shown to be a reliable choice for searches with small statistics [10, 11]. The Collie provides two independent constructions of this test statistic, each designed for specific uses. The first is a Poisson likelihood ratio and the second is a profile likelihood ratio.

III.A. The Poisson Log-Likelihood Ratio

The nominal test statistic compares likelihoods for the Test and Null hypotheses. By treating the two hypotheses as Poisson counting experiments with expected numbers of signal (s) and background (b) and the observed number of data (d) we can construct a Poisson likelihood ratio:

$$Q(s, b, d) = \frac{e^{-(s+b)}(s+b)^d/d!}{e^{-b}(b)^d/d!} \tag{22}$$

In this context, the values of s and b are given by the Monte Carlo templates and d is defined by either the observed data or pseudo-data used to populate distributions of Q. To include multiple bins and/or multiple channels, a joint likelihood can be formed by the multiplicative juncture of the probabilities:

$$Q = \prod_{i=1}^{Nchannels} \prod_{j=1}^{Nbins} \frac{e^{-(s_{ij}+b_{ij})}(s_{ij}+b_{ij})^{d_{ij}}/d_{ij}!}{e^{-b_{ij}} b_{ij}^{d_{ij}}/d_{ij}!} \tag{23}$$

$$= \prod_{i=1}^{Nchannels} \prod_{j=1}^{Nbins} e^{-(s_{ij})} \left(\frac{s_{ij}+b_{ij}}{b_{ij}}\right)^{d_{ij}} \tag{24}$$

where the index i runs over the number of channels and the index j runs over the number of bins in each channel. By recasting the test statistic as a negative log-likelihood ratio (NLLR) Γ a mathematically compact version can be obtained:

$$\Gamma = -2\ln(Q) = 2 \sum_{i=1}^{Nchannels} \sum_{j=1}^{Nbins} (s_{ij} - d_{ij}\ln(1 + s_{ij}/b_{ij})) \tag{25}$$

Although Eqn 25 does not truly represent a Gaussian χ^2 function, it approximates the χ^2 for large numbers of events and provides a more accurate description for small numbers of expected events. This test statistic has the desirable benefit of being monotonically increasing in the number of candidate data events and ensures a non-negative change in sensitivity for each additional channel and/or bin.

In scenarios in which the size of systematic uncertainties is of the order of the Poisson variance or larger, the sensitivity of an analysis can be severely degraded by the averaging described in Eqn. 6.

III.B. The Profile Likelihood Ratio

An alternative test statistic, the profile likelihood ratio, is described in detail in Ref. [12]. This test statistic relies on the minimization of a Poisson χ^2 function to determine the best fit of a system of background models to a data sample. Essentially, the likelihood function is parameterized by the nuisance parameters and their uncertainties. The following equation demonstrates the modified χ^2 used for describing a specified hypothesis:

$$\chi^2(H) = -2\ln P(data \mid H,\theta) = 2\sum_i^{Nbins}\left[(p(H)'_i - d_i) - d_i \ln\left(\frac{p(H)'_i}{d_i}\right)\right] + \sum_k R(H)^2_k \tag{26}$$

where both $p(H)'_i$, describing the predicted number of events in bin i, and R_k are adopted from Eqn. 17. In this way, the R_k values can be fit to minimize Eqn. 26 for a specified point hypothesis.

COLLIE provides two implementations of this profile likelihood maximization technique. The most general application follows from redefining the test statistic to be the negative log-likelihood of the likelihoods maximized independently for H_1 and H_0:

$$\begin{aligned} -2\ln\left(Q(data \mid \theta_0,\theta_1)\right) &= -2\ln\left(\frac{P(data \mid H_1,\hat{\theta_1})}{P(data \mid H_0,\hat{\theta_0})}\right) &(27)\\ &= 2\ln P(data \mid H_1,\hat{\theta_1}) - 2\ln P(data \mid H_0,\hat{\theta_0}) &(28)\\ &= 2\sum_i^{Nbins}\left[(\hat{p}(H_1)'_i - \hat{p}(H_0)'_i) - d_i\ln\left(\frac{\hat{p}(H_1)'_i}{\hat{p}(H_0)'_i}\right)\right] + \sum_k\left(\hat{R}(H_1)^2_k - \hat{R}(H_0)^2_k\right) &(29)\end{aligned}$$

where θ_1 represents the set of nuisance parameters for H_1, θ_0 represents the set of nuisance parameters for H_0, $\hat{\theta_1}$ represents the set of nuisance parameter values that maximize the likelihood for H_1, and $\hat{\theta_0}$ represents the set of nuisance parameters that maximize the likelihood for H_0. The parameters $\hat{p}$ and $\hat{p}$ represent the predicted numbers of events in bin i given $\hat{\theta_1}$ and $\hat{\theta_0}$, respectively. The parameters $\hat{R}$ and $\hat{R}$ represent the central values of the nuisance parameters as defined by $\hat{\theta_1}$ and $\hat{\theta_0}$, respectively.

A second, less reliable, technique arises from the desire to reduce the calculation intensity required for the maximization of the profile likelihood. In this prescription, rather than performing a fit to both TEST and NULL hypotheses, a single fit can be performed to the NULL hypothesis:

$$-2\ln\left(Q(data,\theta)\right) = -2\ln\left(\frac{P(data \mid H_1,\hat{\theta})}{P(data \mid H_0,\hat{\theta})}\right) \tag{30}$$

where $\hat{\theta}$ represents the set of nuisance parameter values that maximize the likelihood for H_0. When choosing to fit to the NULL hypotheses, regions of large signal contamination must be excluded to avoid a bias in the fit. In this case, the user must choose a cutoff point which defines the maximum amount of signal contamination per bin in the fit. An appropriate choice is to exclude bins in which the signal contamination is larger than the background fluctuations. COLLIE allows this cutoff point to be specified in units of $\ln(1+s/b)$ and a conservative default value of 0.005 is preset. Users may change this value as appropriate. Increasing this value allows more signal contamination while decreasing to zero will remove all bins with a non-zero signal prediction. This approach, though faster, will generally result in a less stringent constraint of systematic uncertainties due to a reduced number of bins included in the fit model. This decrease manifests in a less powerful statistical statement about physics models being tested.

The COLLIE interfaces to the MINUIT[13] libraries to perform all fit minimizations. Within the minimization procedure, all systematics are constrained to $\pm 4\sigma$. All systematics inherit an implicit lower bound due to the non-negative requirement enforced for each bin of each individual MC template. This lower bound may be outside $\pm 4\sigma$ and is determined by the nature of the systematic uncertainty and the MC template. In all cases described in this note, it is implicit that the construction of any fitted function depends on the quality of the models being fit. It will require great care to properly utilize such a function and careful attention to standard fit diagnostics is strongly advocated by the author. A detailed list of fit diagnostics is described in Sec. VIII.

IV. CALCULATION OF P-VALUES

Once two ensembles of pseudo-experiments have been generaged, one each for the TEST and NULL hypotheses), the prior predictive ensemble for each can be generated by evaluating each pseudo-data set with a chosen test statistic. These two prior predictive ensembles for the two hypotheses can then be used to calculate p-values. These p-values correspond to the probability to observe an outcome in the pseudo-data that is less signal-like than observed in data. These p-values are commonly referred to as confidence levels, though this interpretation is not rigorous. For this

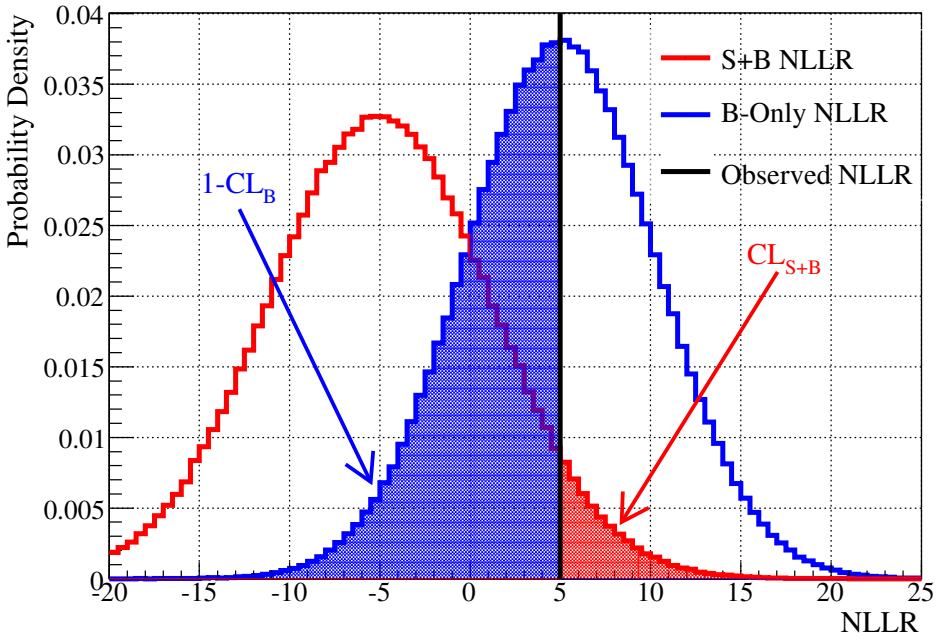


FIG. 2: Example distributions for a NLLR test statistic evaluated for the TEST (red) and NULL (blue) hypotheses. The shaded red (blue) correspond to the values CL_{S+B} $(1 - CL_B)$.

special case, in which cross section limits are physically bounded from below by zero, confidence levels and p-values have similar interpretations as explained below.

Given the prior predictive ensemble for a given hypothesis defined by the test statistic ($\chi = -2\ln(Q)$), one can calculate a confidence level (CL) corresponding to the outcomes which give a result appearing less like the TEST hypothesis than a reference value. Here we retain the confidence level nomenclature:

$$CL = \wp_H(\chi \geq \chi_{ref}) = \int_{\chi_{ref}}^{\infty} \frac{\partial \wp_H}{\partial \chi} d\chi \tag{31}$$

where $\wp_H(\chi > \chi_0)$ refers to the semi-infinite integral of the PDF of the hypothesis H, given by $\frac{\partial \wp_H}{\partial \chi}$. In all test statistic formulations within COLLIE, the signal-like outcomes are more negative than the outcomes more compatible with the background-only hypothesis. In this way, confidence levels and p-values (PVs) for a reference data set (e.g., observed) can be constructed for the TEST (S+B) and NULL (B) hypotheses:

$$CL_{S+B} \;=\; PV_{S+B} = \wp_{S+B}(\chi \geq \chi_{\rm ref}) = \int_{\chi_{\rm ref}}^{\infty} \frac{\partial \wp_{S+B}}{\partial \chi} d\chi \tag{32}$$

$$CL_B \;=\; 1 - PV_B = \wp_B(\chi \geq \chi_{\rm ref}) = \int_{\chi_{\rm ref}}^{\infty} \frac{\partial \wp_B}{\partial \chi} d\chi \tag{33}$$

With this construction, one can interpret these confidence levels in the following manner:

- CL_{S+B} is the probability for the TEST hypothesis to produce an outcome more background-like than that observed in the data. This is also the p-value for the TEST hypothesis.
- CL_B is the probability for the NULL hypothesis to produce an outcome more background-like than that observed in the data. In instances in which the data contains a signal-like excess, the value $1 - CL_B$ (the p-value for the NULL hypothesis) can be used to evaluated the one-sided Gaussian significance of the signal-like excess.

As the test statistic χ is a function of the signal rate, these confidence levels are also functions of the signal rate: $CL_{S+B}(s(x))$, $CL_B(s(x))$, where $s(x)$ defines the parameterization of the signal rate in variable x. Thus, the PDFs of the prior predictive ensembles are dependent on the signal rate and must be re-evaluated for different signal

parameters. Examples of these values and the corresponding test statistic PDFs can be seen in Figure 2. In this example, the distributions are integrated from the observed outcome to evaluate the confidence levels. In principle, the observed value will not match the background-only model perfectly (as in the example figure) and a comparison of the relationship of the TEST, NULL, and observed test statistics can yield direct information about the general agreement between data and the background model. Upon inspection of Eqns. 25 and 29, the median NLLR values for the NULL hypothesis ($d_i = b_i$) will have positive values and the median NLLR values for the TEST hypothesis ($d_i = s_i + b_i$) will have negative values. Neglecting terms higher in order than $(s/b)^2$, the two median values should be symmetric about zero.

An example of this is shown in Figure 3 in which the NLLR distributions from Figure 2 correspond to a model parameter of 10. Included in this figure are the median NLLR values for the TEST hypothesis (NLLR_{S+B}), NULL hypothesis (NLLR_B), and the observed data (NLLR_{obs}). The shaded bands represent the 1 and 2 standard deviation (σ) departures for NLLR_B from the median value. These curves can be interpreted as follows:

- The separation between NLLR_B and NLLR_{S+B} provides a measure of the discriminating power of the search. This indicates the ability of the analysis to separate the TEST and NULL hypotheses.
- The width of the NLLR_B distribution (shown here as one and two standard deviation (σ) bands) provides an estimate of how sensitive the analysis is to a signal-like fluctuation in data, taking account of the presence of systematic uncertainties. For example, when a 1-σ background fluctuation is large compared to the signal expectation, the analysis sensitivity is thereby limited.
- The value of NLLR_{obs} relative to NLLR_{S+B} and NLLR_B indicates whether the data distribution appears to be more signal-like or background-like. As noted above, the significance of any departures of NLLR_{obs} from NLLR_B can be evaluated by the integral of the NLLR_B distribution below NLLR_{obs} $(1 - CL_B)$.

The example figure demonstrates a scenario in which the data value is found to have the same value as the median value for the NULL hypothesis.

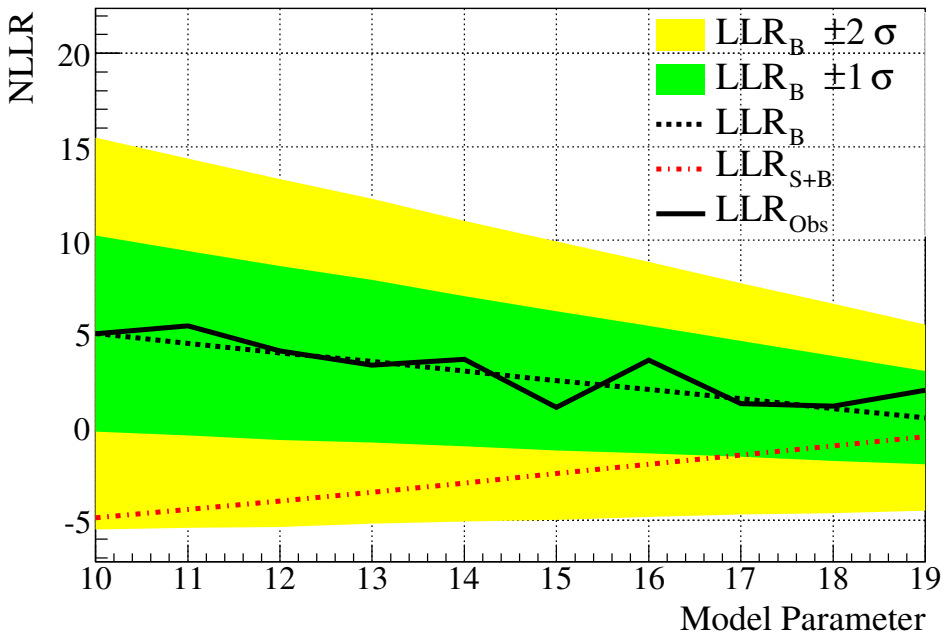


FIG. 3: Example of NLLR test statistic values evaluated as a function of a model parameter. Shown in the figure are the median values of the TEST and NULL test statistics, the observed value, and the one and two σ regions of the NULL test statistic PDF.

A traditional Frequentist hypothesis test relies solely on the value CL_{S+B} to evaluate exclusion limits for model parameters. This construction has the undesirable behavior of producing false exclusions in the instances where the data fluctuates down significantly below the background prediction. For example, consider the scenario of a simple counting experiment in which the number of data events is observed to be significantly lower than that predicted by the background-only model (*e.g.*, $D = 10$ and $B = 100$). This scenario would result in a NLLR value much more positive than the median value of the NULL hypothesis: *e.g.*, a value of +20 in Fig. 2. By inspection of Fig. 2, this

could be interpreted as a very strong constraint on the TEST hypothesis despite the fact that the NULL hypothesis does not seem to accurately model the data. Thus, large downward fluctuations in the data or poor background models can generate exclusions that may not be reproducible with larger statistics or a modified background model. To protect against this pathology, the use of the modified-Frequentist statistic referred to as CL_S is used. This formulation is given as:

$$CL_S(s(x)) = \frac{CL_{S+B}(s(x))}{CL_B(s(x))} = \frac{PV_{S+B}(s(x))}{1 - PV_B(s(x))} \tag{34}$$

where we've made the definition of CL_S in terms of both confidence levels and p-values. The exclusion condition is defined such that $CL_S(s(x_{limit})) < \alpha$. In this definition, α is the fractional confidence level specified and excludes signal at a confidence level of $1-\alpha$ (*e.g.*, $\alpha = 0.05$ for 95% CL) and x_{limit} is the value of the signal model parameter required to meet the condition. Within the COLLIE calculations, the confidence level criterion $1-\alpha$ can be specified at any value.

This construction is commonly referred to as the LEP method, the CL_S method, or the modified-Frequentist construction. The interpretation of the statistic CL_S is not the same as that for traditional Frequentist confidence levels or Bayesian credible levels and a detailed discussion of interpretation can be found in Refs. [10, 11]. A standard Frequentist approach would exclude the TEST hypothesis if the CL_S p-value was less than α. This requirement that CL_S itself be less than α is in fact more conservative and generally generates coverage values greater than $1-\alpha$. This can be appreciated by the following example. Consider the scenario for which $CL_{S+B} = 0.05$. If at the same time the value of NLLR_{obs} is equal to the median NULL hypothesis value (NLLR_B^{med}), the value of $CL_B = 0.50$ and, thus, $CL_S = \frac{0.05}{0.50} = 0.10$. A purely Frequentist interpretation seeking a confidence level for $1 - CL_{S+B}$ of 95% would be satisfied by this condition, but the CL_S technique would return a confidence level of 90%. As the value of CL_B becomes very small, $1-CL_S$ approaches 100% asymptotically. Likewise, as CL_B approaches 1.0, $1-CL_S$ approaches 95% as desired.

V. COMPUTATION OF LIMITS

The conditions for exclusion using the CL_S method are spelled out in Sec. IV. The remaining issue is the determination of the actual value of the signal parameter that satisfies the exclusion condition. There are two general techniques available to COLLIE users.

The first technique is the standard algorithm available within COLLIE and determines the exclusion parameter via successive approximation. This algorithm is based on Ridders' Method for finding the single root for a continuous real function [14]. This method proceeds by first finding parameters which bracket the desired CL_S value and generating successive linear approximations to determine the actual parameter value. This algorithm is quite accurate, but requires a choice of numerical precision. The confidence levels defined in Sec. IV have a binomial error that depends both on the number of pseudo-experiments generated and the desired α value. Due to this error, convergence of the successive approximation method requires the limit condition to be modified to $CL_S(s(x_{limit})) < \alpha \pm \beta$ where β is a small fraction of α (typical values: $\alpha = 0.05$, $\beta = \alpha/50$). The default value of this numerical precision in COLLIE is set to $\beta = 0.001$ and can be decreased for large numbers of pseudo-experiments ($N_{PE} > 2 \cdot 10^4$).

The second technique involves scanning signal parameters to determine the condition $CL_S(s(x_{limit})) = \alpha$. The statistic CL_S generates an error-function as the signal parameter is scanned from values producing $CL_S(s(x_{limit})) > \alpha$ to values producing $CL_S(s(x_{limit})) < \alpha$. By fitting an error function to this shape, one may determine $CL_S(s(x_{limit})) = \alpha$ from the fitted function even though the exact value x_{limit} may not have been tested. This method is also subject to the binomial errors associated with the pseudo-experiment statistics. Thus, the fit to an error-function may not be satisfactory if too few pseudo-experiments were generated. To use this method, COLLIE users must generate and inspect the fitted functions by hand. This technique allows the limit calculation to be parallelized and, thus, reduce the total time required for the calculation.

In summary, COLLIE users can choose from four different algorithms to perform calculations:

- CLFAST: This method ignores all systematic uncertainties. The results are not reliable and this algorithm should only be used for the testing of input files. In this calculation, the prior predictive ensemble **is not marginalized** over the nuisance parameter uncertainties.
- CLSYST: This method includes systematics in the calculations as specified above but with no fitting. In this calculation, the prior predictive ensemble **is marginalized** over the nuisance parameter uncertainties.
- CLFIT: This algorithm uses a single fit to the NULL hypothesis as demonstrated in Eqn 30.

- CLfit2: This algorithm fits both the NULL and TEST hypotheses as demonstrated in Eqn 29.

Flow diagrams for the CLsyst, CLfit, and CLfit2 algorithms can be seen in Figs 4- 6. These diagrams demonstrate the generation of pseudo-data and the evaluation of the test statistic for each algorithm.

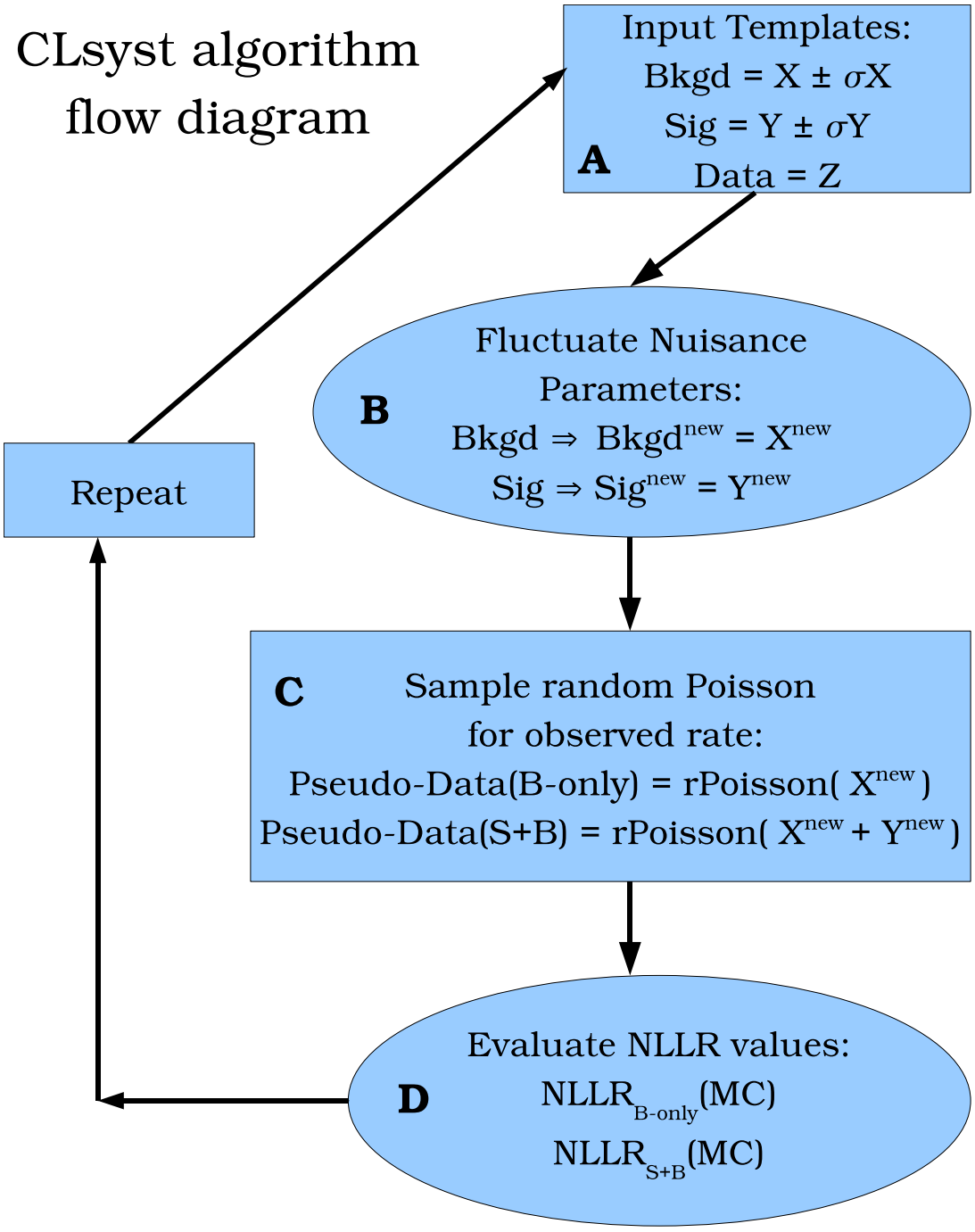


FIG. 4: Flow diagram for the CLsyst algorithm.

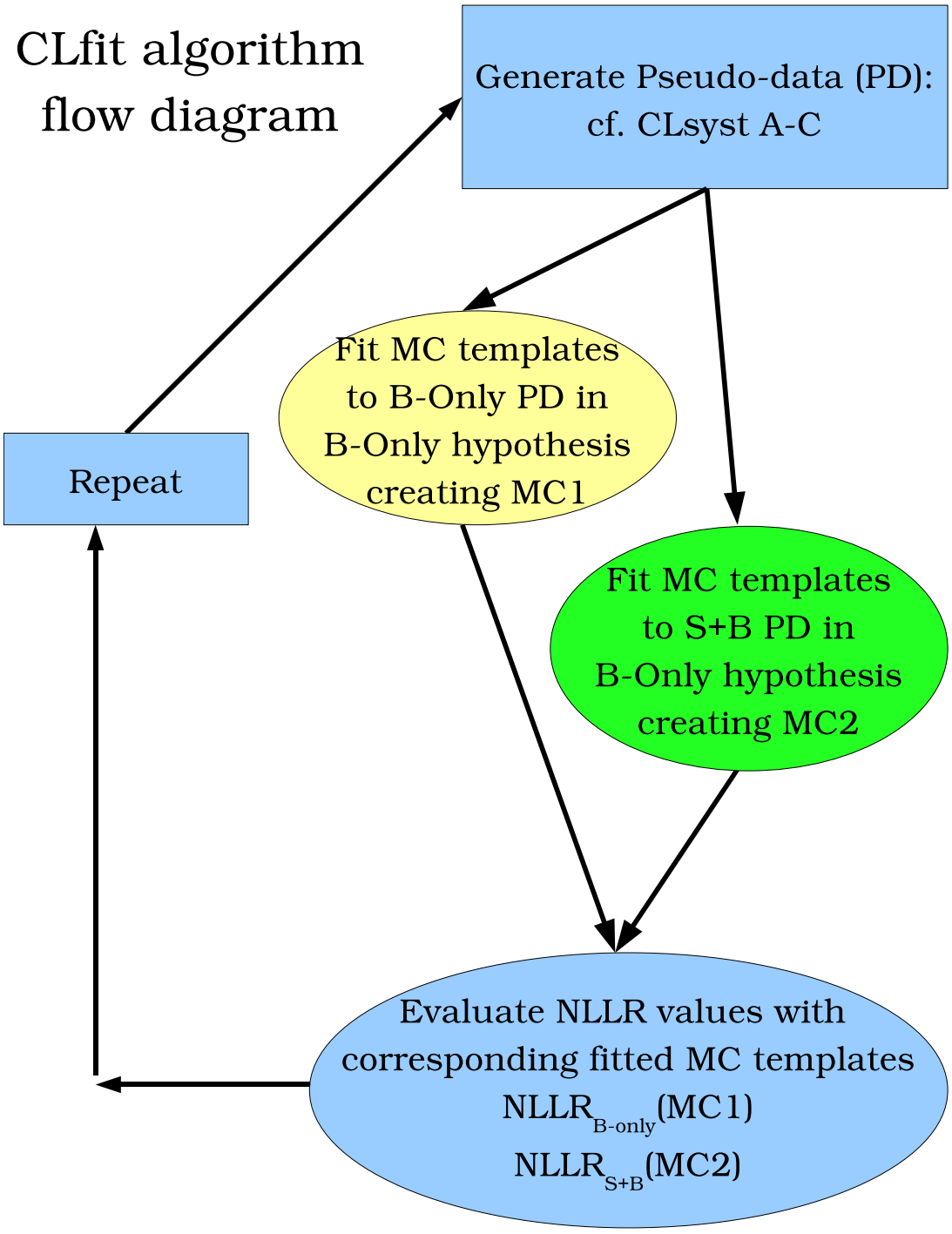


FIG. 5: Flow diagram for the CLfit algorithm, which fits nuisance parameters to a user-specified hypothesis (Background-Only in this example) in selected bins of data and pseudo-data which are expected to be dominated by background.

VI. FITTING

The fitting model within Collie is intended to be very flexible and is thus designed to be rather non-specific. As such, users are responsible for determining the quality, accuracy, and appropriateness of the data/MC fit model. The degrees of freedom available within the fit model are defined by the nuisance parameters , the signal and background MC templates, and the bins that are fit. As noted above, users also have a choice of prior PDFs to assign for the

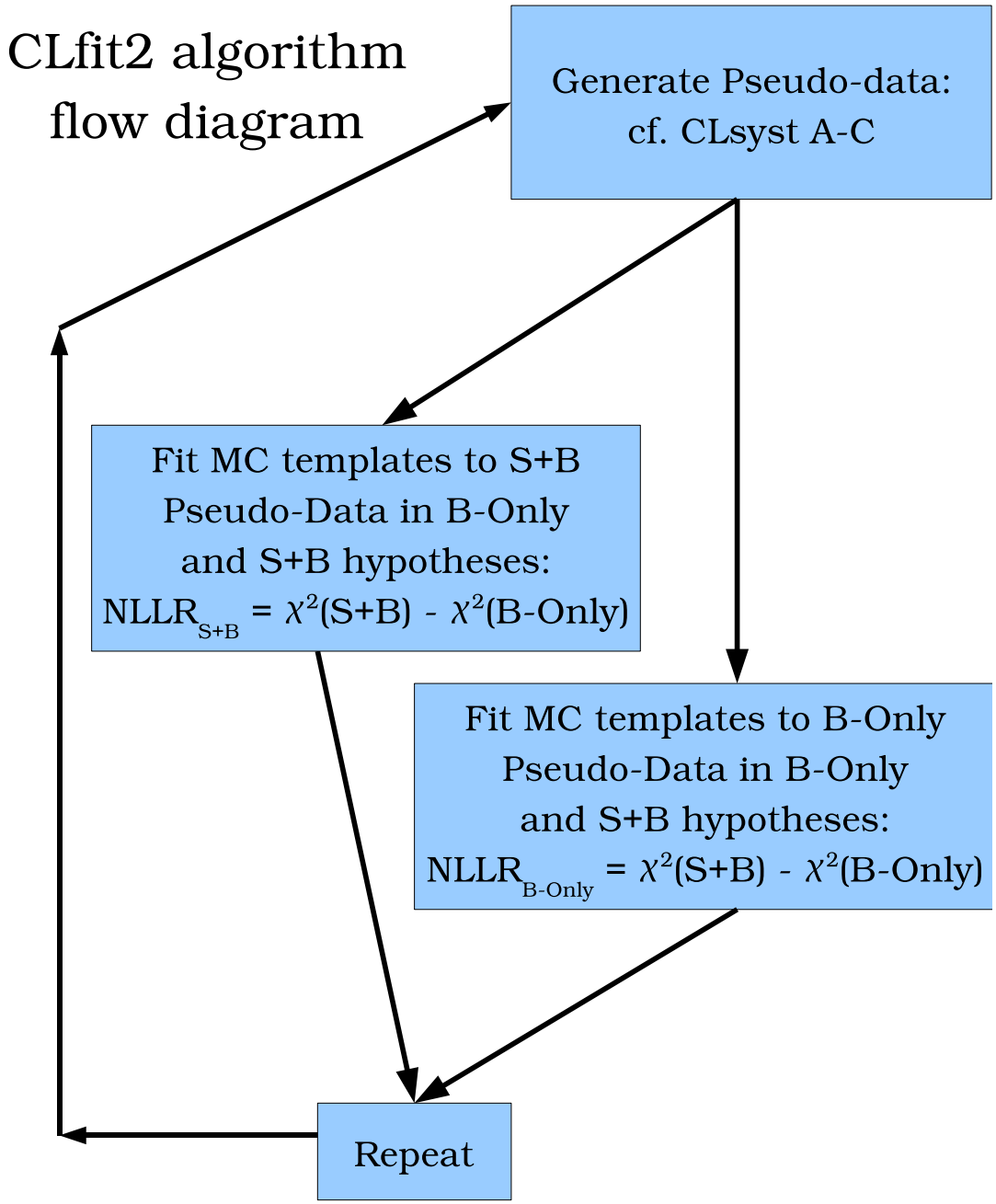


FIG. 6: Flow diagram for the CLfit2 algorithm.

uncertainty of a given nuisance parameter. The fit model can therefore be customized with the following options:

- MC templates: The signal and background templates are defined by the user. The binning of these templates should be appropriate for both the experimental resolution of the variable and the statistical uncertainty of the distribution. Sources of events that have the same set of nuisance parameters should be grouped to reduce computation time.
- Nuisance Parameters: Users must determine the array of orthogonal (mutually exclusive) nuisance parameter

uncertainties that are associated with each event source (ie, signal or background template) for a given analysis channel. In this case, orthogonal specifies that the uncertainties within a given source of events are each uncorrelated amongst each other. Correlations of nuisance parameters across samples and channels must be appropriately specified. When describing the uncertainty of a nuisance parameter in a shape-dependent form, users should take care to ensure the shape dependence is not dominated by statistical effects, which cannot be reliably fit by MINUIT.

- Unconstrained Parameters: The χ^2 function used in the fit model (Eqn. 26) includes a term that enforces a Gaussian constraint on nuisance parameter values. If a nuisance parameter has a large uncertainty ($\geq\sim 50\%$) and/or a poorly known true value, users may "float" the parameter in the fit by removing its Gaussian constraint. This essentially makes the nuisance parameter a free parameter in the fit, at the cost of the extra constraint arising from the nuisance parameter prior (the R^2 term in Eqn 26). When floating parameters and as with all instances, intentionally over-inflation of uncertainties will likely result in undesired outcomes and unreliable fits.

- Nuisance Parameter Priors: COLLIE currently provides for two prior PDFs to be assigned for nuisance parameters: Gaussian or log-normal. Gaussian PDFs are appropriate for uncertainties smaller than $\simeq 25\%$. Above these values, the log-normal PDF is recommended to ensure a vanishing probability distribution as the nuisance parameter approaches zero. This behavior is preferred over Gaussian truncation. In this case, the values of R_{LN} in Eqn. 10 are transformed to be equivalent to the Gaussian assumption in Eqn. 17.

- Bins: Users may select which bins are to be used in fits. This can be achieved by placing a global cut on rectangular regions of the final variable space or by excluding individual bins.

VI.A. Cross Section Measurement

Given the fit model described above, one might be interested in a single of of the TEST hypothesis to data in order to obtain a measurement of a model parameter. The fit model with in COLLIE is easily adapted to this scenario, but one consideration must be made in the interpretation of any results from the fit. As described above, the COLLIE fit model allows for the inclusion of priors for all nuisance parameters and for unconstrained parameters. Therefore, any fit result must be interpreted in the context of the user's fit model design. To meet this need, COLLIE includes a tool designed to extract information on the size of a physics signal cross section by fitting MC templates to data. This tool is constructed by allowing the signal cross section to be an unconstrained parameter in the fit. This is in contrast to the nominal fit model for confidence level calculation, in which no uncertainty is allowed for the signal cross section[17]. The resulting fit provides a measurement of the signal cross section most compatible with the data provided and a determination of the statistical and systematic uncertainties. The cross section measurement is returned in units of the nominal input cross section used in the normalization of the input signal histogram.

VI.B. Cross Section Measurement Significance

Following a measurement of a signal cross section, a natural next step is an estimation of the significance of that measurement. A common generalization of the significance of a measurement (*e.g.*, $A \pm B$) is that the significance is given by A/B standard deviations (*i.e.*, the distance from 0 in units of 1 standard deviation). This interpretation makes the assumption that the problem is characterized by a Gaussian posterior distribution. This assumption implicitly implies the satisfaction of a linearity condition on the standard deviation (σ) (*e.g.*, $2\sigma = 2.0 \times 1\sigma$). Though there are scenarios that satisfy these conditions, it is not a safe assumption in general. A more reliable estimate of significance can be obtained by determining the frequency of predicted outcomes relative to a reference value. This procedure is made available in COLLIE and is described below.

The implementation of this significance test in COLLIE is constructed by performing cross section measurement fits to pseudo-data generated in the same manner as described in Sec. II. Users may select the number of pseudo-data to be fit and the hypothesis from which they are drawn. The resulting cross section measurements are histogrammed and provided to the user for analysis. The cross section measurements are recorded in units of the nominal input cross section used in the normalization of the input signal histogram. Users may then integrate the histograms above arbitrary reference values to obtain p-values corresponding to the relative probability of the reference cross section.

These p-values correspond to the probability that the tested hypothesis will produce a signal cross section at least as large as the reference cross section, which may be interpreted in terms of Gaussian standard deviations. However, because the p-values correspond to infinite integrals above a reference point, the interpretation must be made as a one-sided Gaussian probability. Consider first that when the results of a measurement are properly described by a

Gaussian with unit standard deviation and mean value N, then the probability enclosed in the region $[N-n, N+n]$ is given by:

$$\mathrm{Erf}\left(\frac{n}{\sqrt{2}}\right) \tag{35}$$

where Erf indicates the error function. For a **Null** hypothesis p-value $\mathcal{P}^{Null}$, the one-sided Gaussian significance of n standard deviations can thus be determined as follows:

$$\mathcal{P} = \frac{1 - \mathrm{Erf}\left(\frac{n}{\sqrt{2}}\right)}{2} \tag{36}$$

$$n = \sqrt{2}\ \mathrm{ErfInv}\left(1 - 2\mathcal{P}\right) \tag{37}$$

where ErfInv indicates the inverse error function and the factor of 2 accounts for the one-sided integral. The interpretation of TEST hypothesis p-values is different, however. Because the median expected value is non-zero, there is a distinction of p-values above and below 50%. For a **Test** hypothesis p-values $\mathcal{P}^{Test} < 50\%$, the one-sided Gaussian significance of n standard deviations is given by Eqn. 36 and the interpretation is that the result is n standard deviations ***above*** the median expected value. For a **Test** hypothesis p-values $\mathcal{P}^{Test} > 50\%$ (*i.e.*, less than the median), the one-sided Gaussian significance of n standard deviations is given by:

$$\mathcal{P} = \frac{\mathrm{Erf}\left(\frac{n}{\sqrt{2}}\right)}{2} \tag{38}$$

$$n = \sqrt{2}\ \mathrm{ErfInv}\left(2\mathcal{P}\right) \tag{39}$$

and the interpretation is that the result is n standard deviations ***below*** the median expected value. The ROOT software package contains methods for both the error function and inverse error function (TMath::Erf(), TMath::ErfInverse()).

VII. EXAMPLE ANALYSIS

As a pedagogical example, one can consider a simple, contrived example analysis. The example consists of two backgrounds and one signal, each with associated systematic uncertainties. The data is constructed as the sum of the two backgrounds with the following changes:

- Bkgd 1: This background's data contribution is generated at 5% (0.33σ) higher than the prediction.
- Bkgd 2: This background's data contribution is generated at 2% (0.13σ) lower than the prediction.
- Bkgd 2: This background's shape is generated at 0.50σ higher than the prediction.

The data is designed to be just slightly more signal-like than the background-only model, but with zero signal contribution. The signal, background, and data distributions can be seen in Fig. 7 and the corresponding normalizations and systematics are found in Table I. The distributions for the shape-dependent systematic uncertainties are shown in Fig. 8. Using this example, p-values and 95% CL limits can be derived for each possible calculation: no systematics (CLfast), standard systematics (CLsyst), single-fit (CLfit), and double-fit (CLfit2). These results are shown in Tables II and III. The 95% CL limits are given in units of the ratio to the input signal cross section used to normalize the signal rate to the expected number of events. These results demonstrate the improvement found by using the two fitting methods, while the double-fitting method (CLfit2) returns the tighter limit of the two. Depending on sample size, analyzers should expect a similar behavior of the four calculations. The NLLR distributions for all four calculations are shown in Fig. 9. Each NLLR distribution is generated at the corresponding observed exclusion signal rate and represents the excluded conditions.

In this example, all fits are performed with three MC templates (one signal and two backgrounds) and six inter-correlated systematics. The minimization is performed by adjusting the central value of each nuisance parameter which is nominally equal to 1.0 in units of its absolute normalization. Based on the correlations for the nuisance parameter in question, the MC templates are adjusted and Minuit determines the minimum of the χ^2 function. All deviations of the nuisance parameters are evaluated in units of $N-\sigma$ and the R_k^2 term in the χ^2 function corresponds to the square of these $N-\sigma$ deviations. This term discourages large deviations from the nominal prediction to reflect the priors associated with the nuisance parameters.

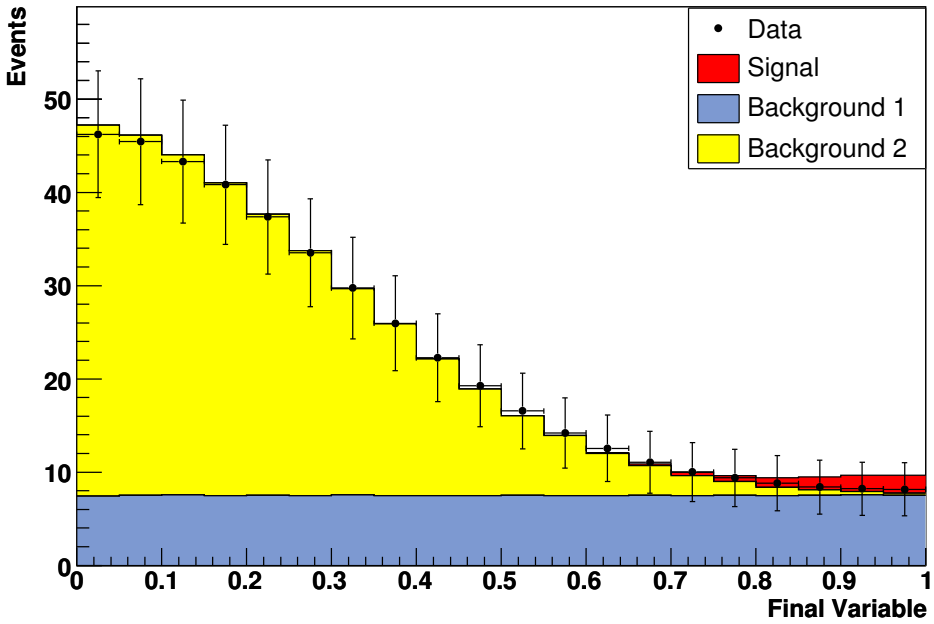


FIG. 7: Signal, background, and data distributions used in the example analysis.

The results for the CLfast calculation exclude information on systematic and statistical uncertainties, as described above. This calculation is for testing purposes only and represents the most optimistic separation of signal and background. The remaining three calculations can be compared to this scenario via the results listed in Tables II and III and Figure 9.

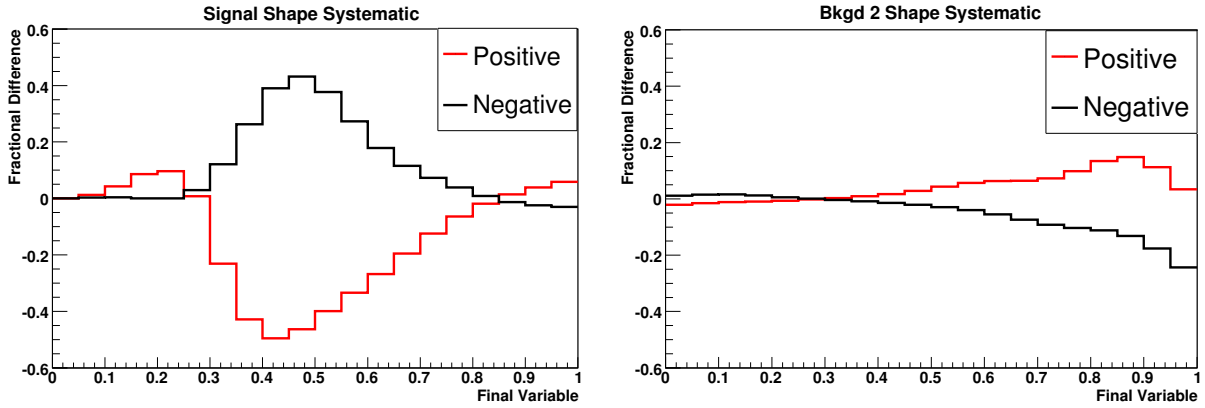


FIG. 8: Shape-dependent systematic uncertainties for signal (left) and background 2 (right). The value plotted are the fractional change from the nominal predictions for positive and negative 1-σ fluctuations.

Source	# Events	Lumi	Eff	Bkgd 1 Xsec	Bkgd 2 Xsec	Signal Shape	Bkgd 2 Shape
Signal	7.5	6%	10%	-	-	5%	-
Bkgd 1	150	6%	10%	15%	-	-	-
Bkgd 2	300	6%	10%	-	15%	-	5%
Data	451	-	-	-	-	-	-

TABLE I: Summary of specifics for the example analysis.

The results for the CLsyst include the marginalization effects of the nuisance parameter uncertainties. This effect can be seen as the expected and observed CL_{S+B} values both increase and the observed CL_B value moves closer to 50%. Both of these effects are due to the broadening of the NLLR distributions for the two hypotheses. The expected and observed limit ratios have a commensurate increase related to this decrease in significance.

The CLfit algorithm demonstrates an improvement over the CLsyst results. This algorithm fits all pseudo-experiments to the Null hypothesis using only bins with low signal contamination. Though this algorithm does not use the full information available in the data, it is expected to improve the constraint on the distribution of systematic fluctuations. In cases in which the data statistics are too low to constrain systematic fluctuations, little improvement will be seen over the CLsyst results. In this case, most notably, the width of the NLLR distributions decreases and the CL_B value moves yet closer to 50%. The expected and observed limit ratios improve (decrease) due to this improvement in relative sensitivity.

The CLfit2 algorithm fits each pseudo-experiment to both the Test and Null hypotheses using all bins regardless of signal contamination. Thus, one should expect a small improvement over the CLfit results. Here, the value of CL_B^{obs} moves the closest to 50% of all calculations and the width of the background-only NLLR distribution approaches that of the CLfast calculation. The expected and observed limit ratios both improve relative to the CLfit algorithm.

VII.A. The Error Matrix

The manner in which the CLfit and CLfit2 classes improves limits can be understood via inspection of the error matrices obtained in fits to the Test and Null hypotheses. These distributions are shown in Fig. 18. The large negative correlations are indicative of the constraint of the total normalization (increases in rate due to one systematic must be balanced by a decrease via another systematic). This effect should be able to be reproduced by introducing the resulting error matrices into the marginalization procedure (Eqn. 6). When using the error matrices in the nuisance parameter sampling procedure (separately using the Test and Null hypothesis error matrices for the Test and Null hypothesis pseudo-experiments, respectively), the distributions in NLLR for the CLfit2 algorithm can be reproduced. This behavior has been confirmed to indeed result in the same 95% CL cross section upper limit as obtained via the CLfit2 algorithm.

This is not a standard feature of the Collie package. Such an implementation is not a valid hypothesis test in so much that a different error matrix is being used for each hypothesis. Choosing one hypothesis also violates the test,

Calculation	CL_B^{obs}	CL_{S+B}^{obs}	CL_B^{exp}	CL_{S+B}^{exp}	Exp Limit Ratio	Obs Limit Ratio
CLFAST	0.62	0.22	0.51	0.14	1.9	2.1
CLSYST	0.60	0.32	0.51	0.24	3.0	3.3
CLFIT	0.58	0.26	0.51	0.20	2.5	2.7
CLFIT2	0.57	0.25	0.50	0.19	2.4	2.6

TABLE II: Summary of confidence level statistics for the example analysis. All CL values are evaluated at the nominal signal rate. All CL_B^{exp} are nominally equal to 0.50, but may vary slightly due to binning effects. The 95% CL limits are given in units of the ratio to the input signal cross section used to normalize the signal rate to the expected number of events.

Calculation	$NLLR^{obs}$	$NLLR^{B-Only}$	$NLLR^{S+B}$	RMS^{B-Only}	RMS^{S+B}
CLFAST	2.76	3.87	-4.32	3.72	4.37
CLSYST	6.90	8.53	-9.98	8.47	10.6
CLFIT	4.77	5.94	-6.77	5.98	7.15
CLFIT2	3.26	3.85	-3.98	3.90	4.08

TABLE III: Summary of NLLR values for the example analysis evaluated with the signal at the 95% CL expected limit rate.

as the error matrix depends on the signal size and test hypothesis. Thus, though this is a useful test, such a technique cannot be generalized.

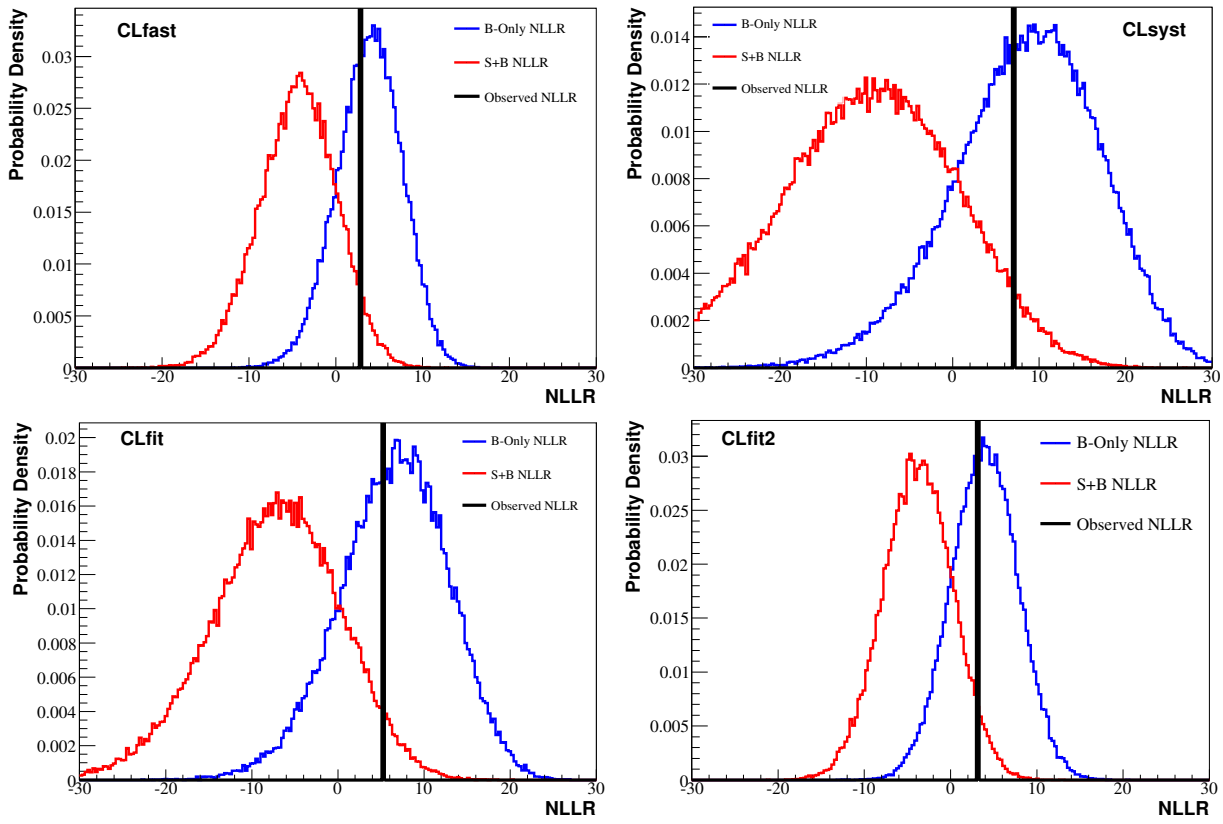


FIG. 9: NLLR distributions shown for the four different calculations. Each distribution is evaluated at the corresponding observed excluded signal rate.

VIII. COLLIE FIT DIAGNOSTICS

The class FitTest found in the subdirectory collie/limit/ is designed to pass the input distributions and their associated systematic uncertainties through a series of tests intended to assess the quality of the fit achieved by Collie's fitting method. Users must specify the number of fit trials via the method FitTest::setIterations(int niter) and also choose whether fits to pseudo-experiments should be performed via the method FitTest::testPE(bool choice). The test generates histograms stored in Root format, which can be viewed on canvases using the macros collie/limit/macros/fitResults.C. The following results are available (Reminder: Test refers to the signal-plus-background hypothesis while Null refers to the background-only hypothesis). Inspection of the results of these diagnostic tests should be considered a mandatory step in utilizing the fitting features of Collie. Any review of the results from Collie should include the figures produced in the fit test. The available fit diagnostics include the following:

- One canvas per systematic uncertainty containing three histograms showing the pull function for fits to data, the pull function for fits to pseudo-experiments, and the response of the χ^2 function to the specified systematic. These pull functions are evaluated for each fit (i) and are defined via the starting value of the nuisance parameter (V_i^{Start}), the final post-fit value of the nuisance parameter (V_i^{Fit}), and the uncertainty on the nuisance parameter ($\sigma(V_i)$). The uncertainty $\sigma(V_i)$ is taken from the $\pm 1-\sigma$ variations provided by the user to define the nuisance parameter prior PDF.

$$PULL_i = \frac{V_i^{Fit} - V_i^{Start}}{\sigma(V_i)} \tag{40}$$

  In the case of fits to data, TEST hypothesis pseudo-data are generated and fit to data. In this scenario V^{Start} is a random value chosen from the prior PDF for the nuisance parameter in question. In the case of fits to pseudo-experiments, the nominal TEST hypothesis model is fit to pseudo-data and V^{Start} is defined as the predicted value of the nuisance parameter. Examples of these distributions for the example analysis are shown in Figs 10-15.

  - The response of the χ^2 function is an indication of the nature of the constraint on the specified systematic. This function represents the change in χ^2 between the data and MC templates. The values are evaluated after performing a fit to a specific hypothesis (both the Test and Null hypotheses are shown) such that the χ^2 response function is calculated in the region of the global minimum. A well-modeled systematic uncertainty should generate a 2nd-order polynomial with a single, clear minimum. Systematics which give a near-linear response with no clear minimum are under-constrained and have little impact on the fit. That is, systematics that are very small or apply to backgrounds with small rates will have small impacts on the χ^2 and, thus, the fit. Response functions which become flat at large negative fluctuations are simply zeroing the backgrounds in question, which causes the χ^2 function to be insensitive to the nuisance parameter. The point of flat response indicates the valid response region for the specified size of the systematic (*i.e.*, there is no information available inside the flat region). An estimate of the constrainable size of the systematic can be obtained by evaluating the points where the χ^2 response function changes value by ± 1 to find ± 1-σ regions. The units of these changes are reported in the nominally specified 1-σ values ($\sigma(V_i)$). Thus, if the N $\sigma(V_i)$ values that satisfy $\Delta\chi^2 = 1.0$ are less than unity, the nominally specified systematic size may be overestimated. For example, if your luminosity systematic is 6.1% and $\Delta\chi^2 = 1$ occurs at 0.8 $\sigma(V_i)$, the effective constraint is 4.9%. However, this interpretation is not universally safe as it does not include effects from non-diagonal terms in the covariance matrix. More information on this aspect can be found in Section IX.
  - The pull functions are a second estimate of the nature of the constraint on the systematic in question. The previous tests probe the one dimensional response, while the pull function values reflect the response in N-dimensional space where N is the total number of systematics. There are two pull functions defined:
    * The first pull function (upper left) represents fits of Monte Carlo templates with smeared nuisance parameters to data. The nuisance parameter central values are chosen randomly from their prior PDF distributions. Thus, in this case the starting values of the nuisance parameters V_i^{Start} are not equal to zero and the fit data is always the same. The subsequent fits to the observed data thus give an indication of how well the Monte Carlo templates and the systematics model reproduce the observed data distribution. Due to the Gaussian constraint in the χ^2 function, the pull function distributions are expected to have a maximum width of 1.0. This scenario indicates that the data distribution does not

offer any additional information on the true value of the nuisance parameter beyond that included in the prior PDF. Widths less than 1.0, however, indicate that the nuisance parameter uncertainty can be constrained below the prior PDF by the data distribution. The central value of the pull distribution can give an indication of any offsets preferred due to residual differences between data and Monte Carlo. Differences between the background-only and S+B fits should be commensurate of the size of the signal being fit. In other words, a small signal isn't expected to be able to change fit parameters, but a large signal could do so. The mean values and widths arising from a Gaussian fit to the pull functions are given in the lower left region of the canvas.

* The second pull function (upper right) represents fits of the nominal Monte Carlo templates to pseudo-data. The pseudo-data are generated in the same manner as is used in populating the prior predictive ensemble. Thus, in this case the starting values of the nuisance parameters V_i^{Start} are equal to zero and the fit data is always different. These fits give an indication of what nuisance parameters are dominant in the fits to pseudo-data. This can be determined by inspection of the widths of the pull functions. In general, the wider the pull function the more important the nuisance parameter is in reducing the χ^2 function. Another way to to visualize this behavior is that if the reduction in χ^2 obtained from adjusting the nuisance parameter value is large compared to the increase from the Gaussian R^2 term from the prior, then the fit will prefer to move the nuisance parameter. If the reduction in χ^2 is small compared to the Gaussian R^2 term, the fit will prefer to leave the parameter alone. For nuisance parameters with a Gaussian prior, the values in the pseudo-data will be randomly centered around zero. Thus, one should expect the mean value of the pull functions to be very near zero. For log-normal priors or very asymmetric uncertainties, this may not hold. The mean values and widths arising from a Gaussian fit to the pull functions are given in the lower left region of the canvas. This plot also includes the nuisance parameter values as determined from fits to the observed data. This comparison allows a determination of the relative data/MC agreement within the apparent width of fits to pseudo-data, which is a measure of how well the prior predictive ensemble can reproduce the observe data.

- One histogram containing the MINUIT status following fits to data for the TEST and NULL hypotheses. A properly converged fit returns a value of 3 and any other result indicates non-convergence or uncertain error matrices. Analyzers need to investigate the source of any deviations from 3. One histogram containing the number of MINUIT iterations required to achieve the χ^2 minimization. The default maximum number of iterations is 10000 to avoid infinite loops. If the number of iterations approaches 10000, analyzers should increase the maximum number allowed. See Figure 16.

- Three canvases containing the S/B values, $\log_{10}(1+S/B)$ values, and data/MC comparison rebinned in $\log_{10}(1+S/B)$ determined from the nominal (non-fitted) signal and background distributions. If you are using the CLFIT class, you must choose a constraint cutoff to define the background sidebands. The $\log_{10}(1+S/B)$ distribution is what's used to define fitting regions. The $\log_{10}(1+S/B)$ rebinned distributions provide an opportunity to inspect the data/MC agreement in the bins that contribute the most to the search sensitivity. See Figure 17.

- Two canvases containing the two-dimensional correlation matrix obtained from MINUIT after a TEST and NULL fits to data. See Figure 18.

- One histogram containing the TEST and NULL central values for systematic uncertainties determined by fits to data in the TEST and NULL hypotheses, respectively. These are the central values of the fitted systematic parameters which minimize the χ^2 function. A second canvas shows a comparison of the systematic uncertainty values (in units of the input systematic) for the nominal values (equal to 1.0 by definition), the parameter errors determined by MINUIT's fit to data, and the parameters obtained from the pull functions for fits to data. The values determined for the COLLIE fits are given by the widths of the data fit pull functions. The values from MINUIT are the MINOS errors and represent a more sophisticated estimate of true parameter size. Each entry represents the square root of the corresponding diagonal element from the fit covariance matrix (each entry: $x = \sqrt{\sigma_{ij}^2}$, $i = j$). See Figure 19. Given that the example data and background MC distributions were designed to disagree in a specific manner but agree with the B-Only hypothesis better than the S+B hypothesis, it is useful to inspect the results of the background only fit:

  - The values of "Xsec1" and "Xsec2" in the data were set to $+0.33\sigma$ and -0.13σ relative to the predicted values, respectively. The fit converged to values of $+0.175\sigma$ and -0.12σ, respectively. Both of these nuisance parameters were found to have significant negative correlations with other parameters, especially "Eff" and "Lumi". The non-zero fit values for these parameters can account for the difference in fit value for "Xsec1" and its generated value. This is an instance in which the relative χ^2 reduction and moving a

nuisance parameter were forced to compromise (*i.e.*, $0.33^2 \simeq 6.25 \times 0.13^2$, thus the R^2 term played a larger role for the fit of"Xsec1").

- The "Bkgd2Shape" nuisance parameter in the data was set to $+0.50\sigma$ and the fit value was +0.06. This nuisance parameter is found to have little correlation with other parameters, but the correlations outlined in the previous bullet will tend to reduce the departure required for this parameter.
- When fitting to the S+B hypothesis, the fit is attempting to accommodate a signal that is not present in the data. Thus, in this fit the nuisance parameters naturally take on different values. Most of the rate nuisance parameters are lowered (to include more events from signal) and the cross section uncertainties allow a change in background shape.

- Two canvases containing the distributions of signal and background systematic variations relative to the nominal values for each bin of the final variable distribution. Each canvas contains a distribution for the non-fitted values (black), the values from the TEST fit, and the NULL fit. A bin with no change enters the plot with a value of 1.0. The "No Fit" values are determined by randomly sampling each nuisance parameter and entering the ratio of each altered bin value to the nominal. The mean value should be at 1.0 and the width should correspond to the total uncertainty. The fitted distributions are the values of the nuisance parameters following fits of MC templates to data, following a random sampling of nuisance parameters. These distributions give a measure of the total dispersion of nuisance parameter values before and after fitting to data. This distribution can be biased by bins with small uncertainties. The following two figures provide a per-bin view of the same values. See Figure 20.

- Two canvases showing the RMS of the uncertainties for each bin with no fitting and following fits to the TEST and NULL hypotheses. The distributions are constructed in the same manner as the previous two figures, but are plotted separately for each bin of the final variable distribution.

- Three canvases containing the signal, background, and data distributions before fitting, after the TEST fit, and after the NULL fit. The fourth canvas compares the background distributions before and after the fitting. These distributions demonstrate how the shape of the background changes under each "best fit" scenario. These comparisons show how different the TEST and NULL fits are and how well each fit reproduces the observed data. See Figure 22.

- Three canvases containing the number of Gaussian sigma per bin (residuals) based on a data/MC comparison. The values are calculated as the difference between data and MC divided by a symmetric assumption of the data's Poisson uncertainty in the bin (*i.e.*, the subtraction of the previous three figures divided by the data Poisson uncertainty). The comparison is performed for the baseline MC prediction, the TEST fit to data, and the NULL fit to data. The total $\chi^2/Ndof$ is also displayed on the figure. One should expect the post fit residuals to decrease. In this example, the S+B fit residuals increase, which simply indicates better agreement with the B-Only hypothesis. See Figure 23.

- Histograms containing the TEST and NULL χ^2 distributions for fits to data and pseudo-data in the TEST and NULL hypotheses, respectively. For large number of events, these distributions should roughly follow standard χ^2 distributions for N degrees of freedom, where N is determined by the number of bins fit and the number of fit parameters. For fits to data, the fit of the nominal MC templates to data is shown for each fit hypothesis. See Figure 24.

- Canvases containing "N-1" or deletion tests of the χ^2 values obtained after removing individual channels, individual systematic uncertainties, and individual bins for fits to each hypothesis. The deletion indicates that the corresponding parameter is not allowed to be included in the fit model: if it's a nuisance parameter, the parameter cannot change in the fit; if it's a bin, the fit ignores that particular bin; etc. The results are organized into a one-dimensional histograms showing the $\Delta\chi^2$ values following deletion of entire channels, individual nuisance parameters, and individual bins. A two-dimensional histogram of correlated deletions of nuisance parameters and channels is also available, but not shown here. See Figure 25.

A flow diagram demonstrating the evaluation of the pull functions and the per-bin dispersions can be seen in Fig. 26. COLLIE users who employ any fitting classes (CLFIT, CLFIT2, or PROFILELH) must perform this fitting test to analyze their fit construction. Failure to do so can result in unreliable results due to failed convergence or a poorly constructed fit.

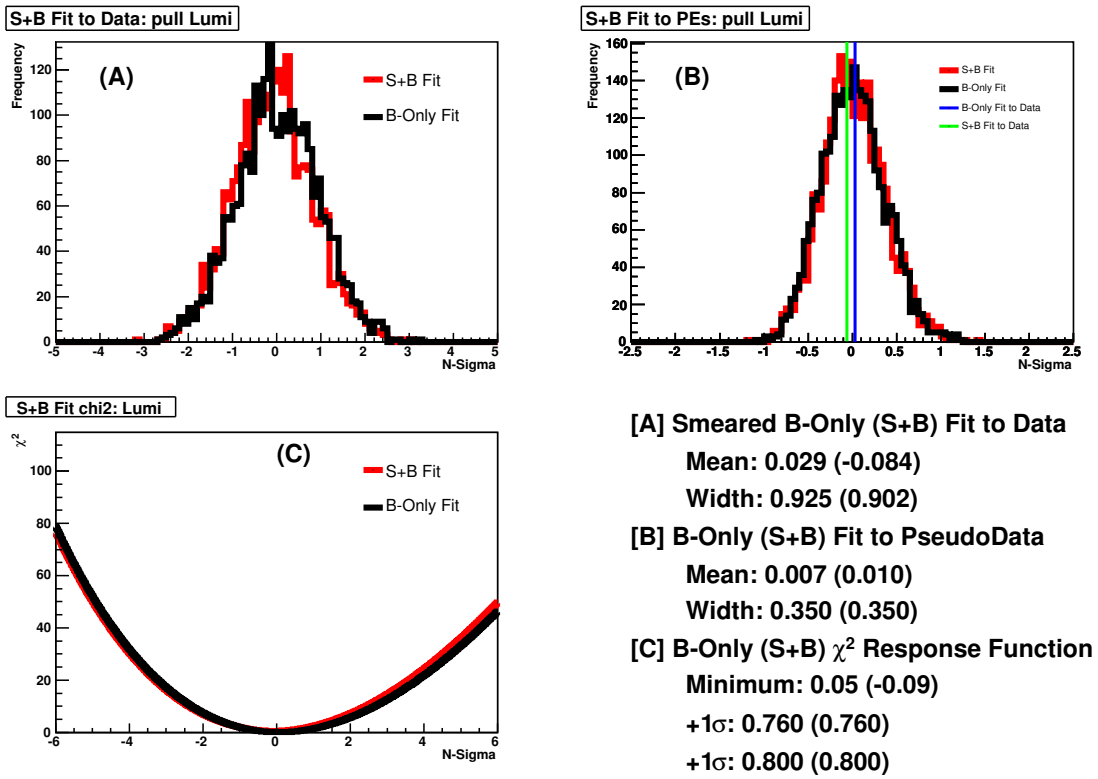


FIG. 10: Pull functions and χ^2 response function for the luminosity systematic in the test example analysis.

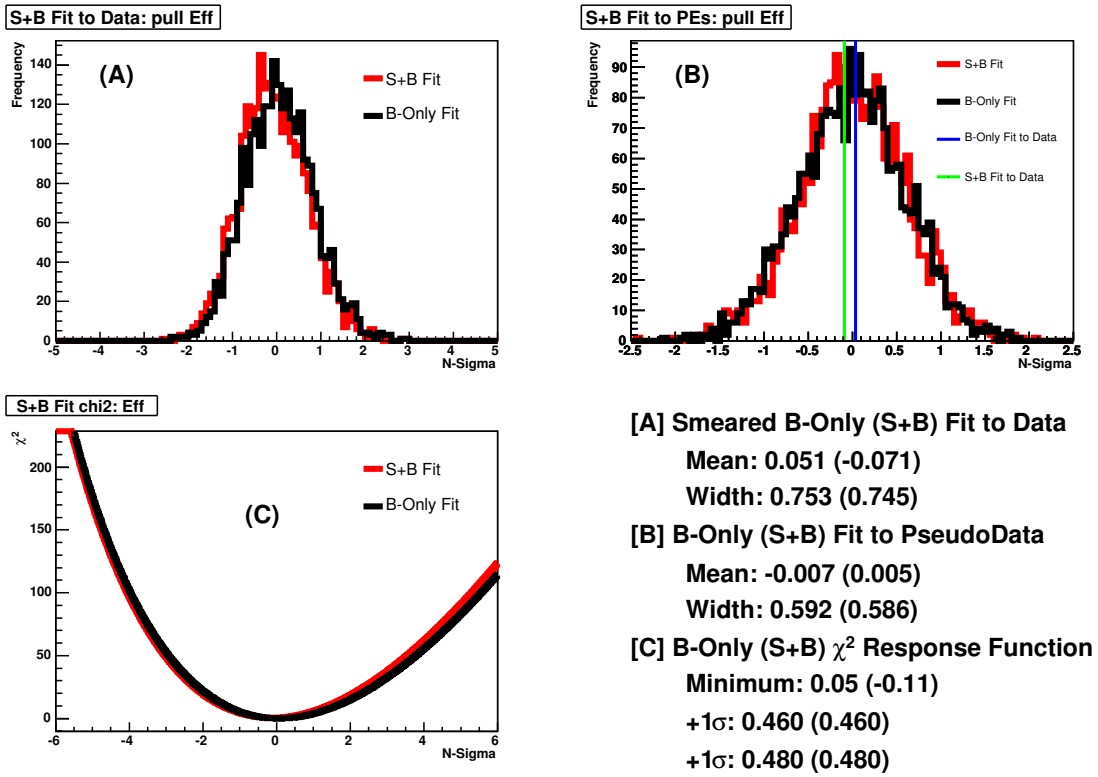


FIG. 11: Pull functions and χ^2 response function for the efficiency systematic in the test example analysis.

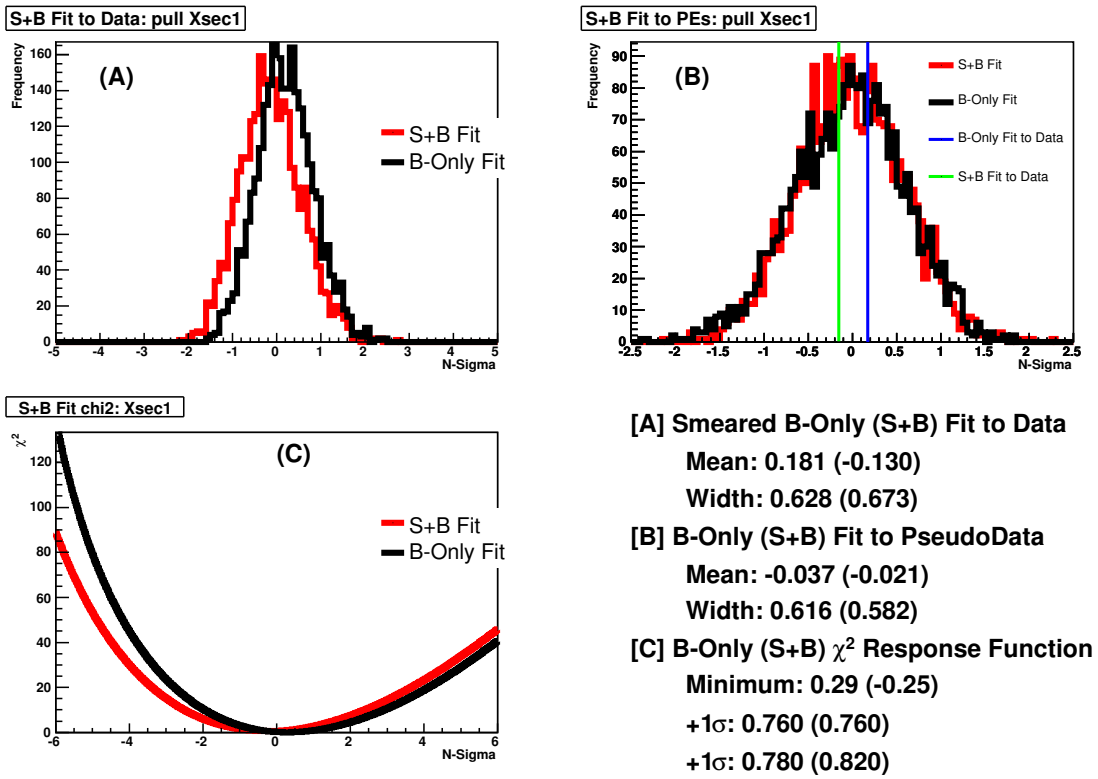


FIG. 12: Pull functions and χ^2 response function for the background 1 cross section systematic in the test example analysis.

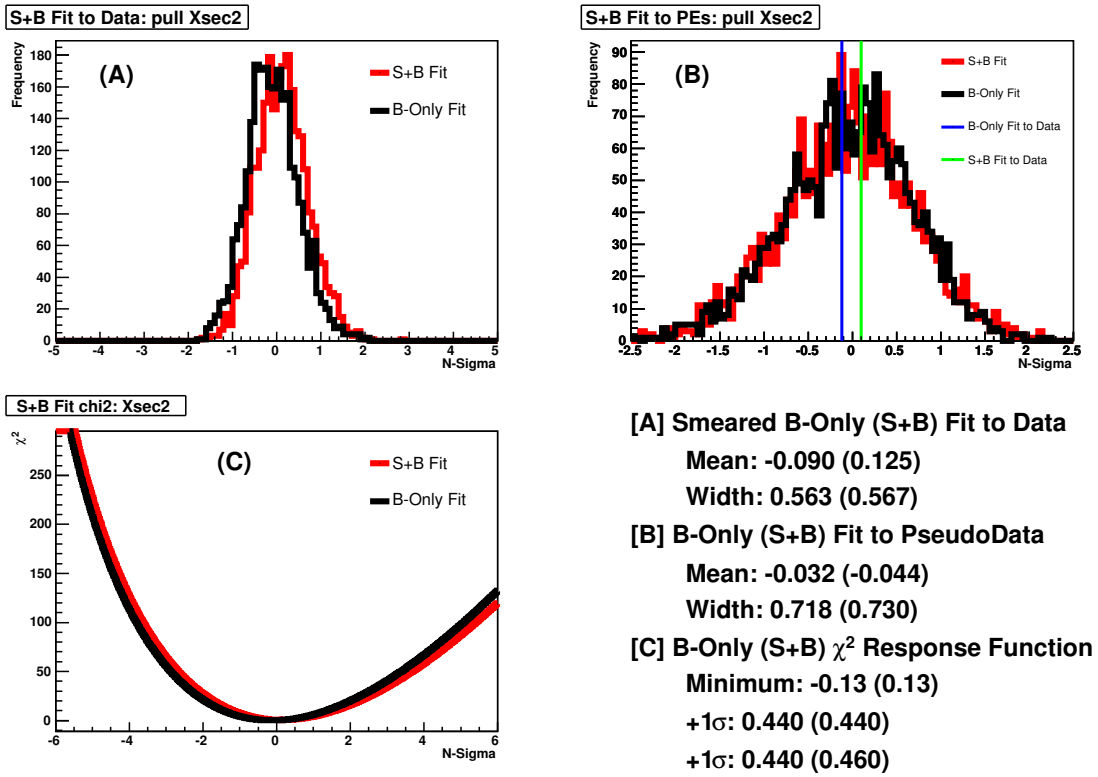


FIG. 13: Pull functions and χ^2 response function for the background 2 cross section systematic in the test example analysis.

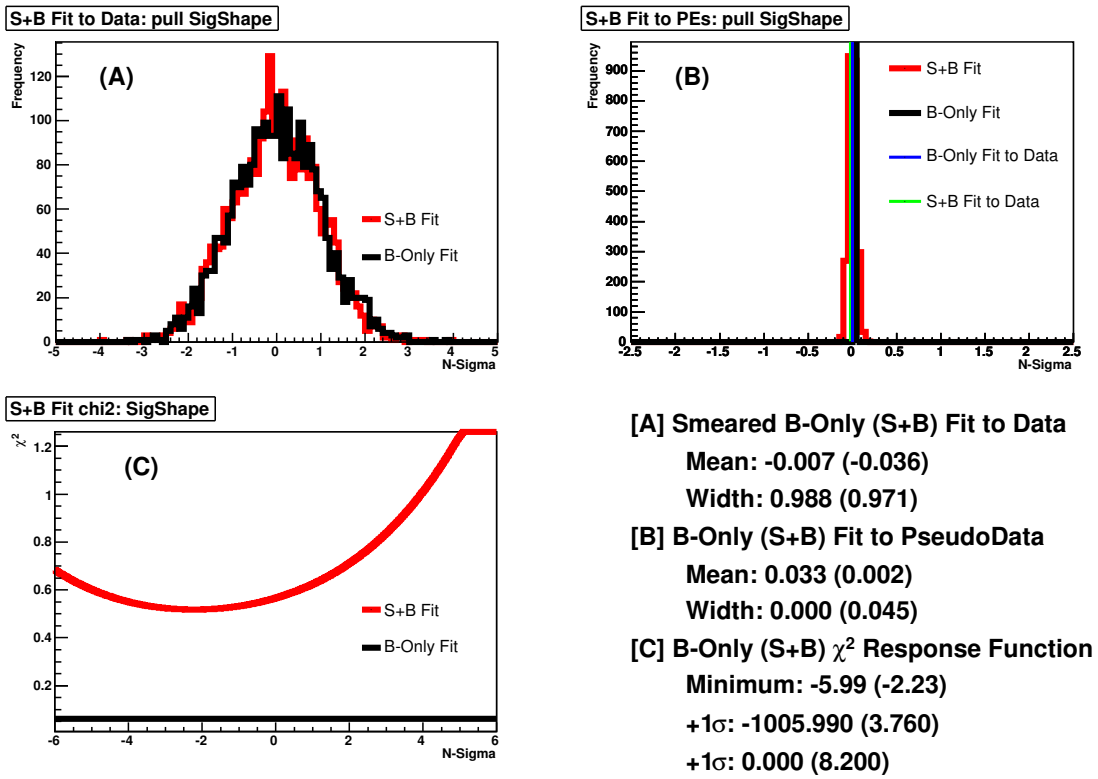


FIG. 14: Pull functions and χ^2 response function for the signal shape systematic in the test example analysis.

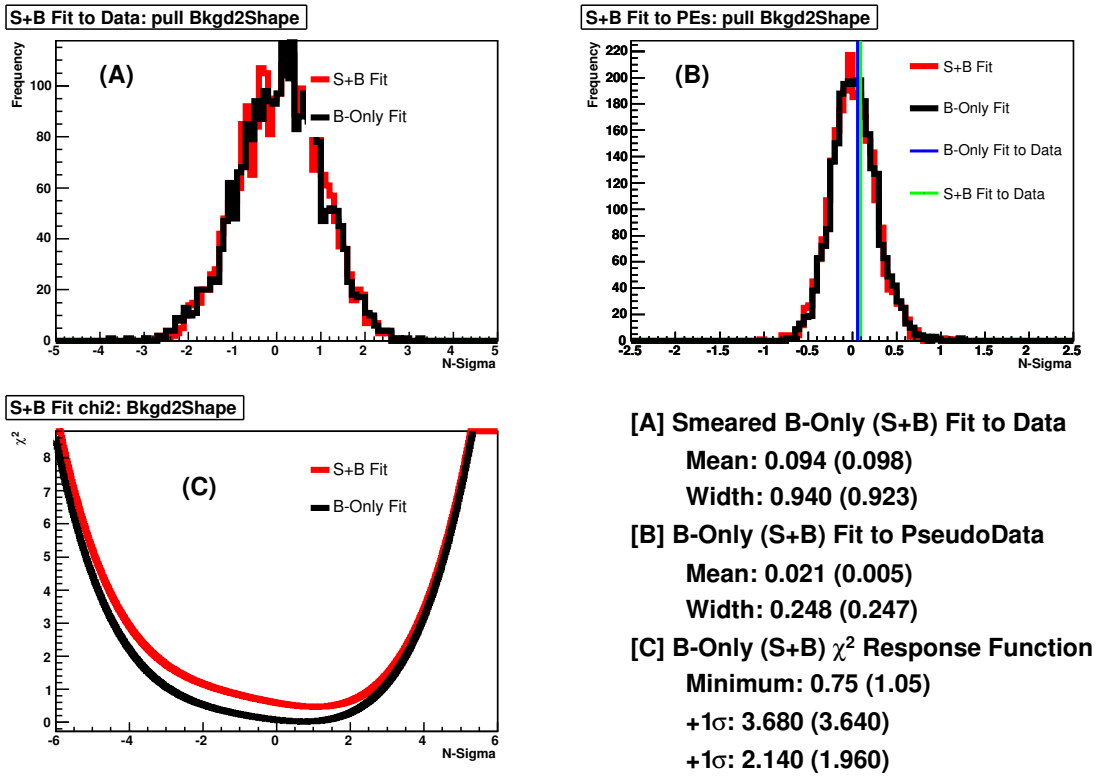


FIG. 15: Pull functions and χ^2 response function for the background 2 shape systematic in the test example analysis.

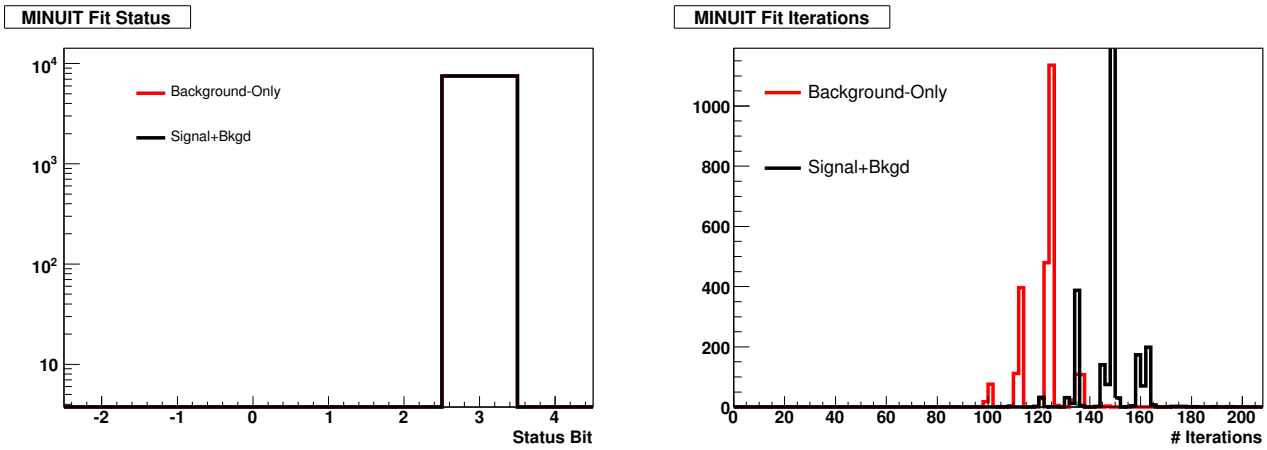


FIG. 16: MINUIT fit status and fit iterations for the example analysis.

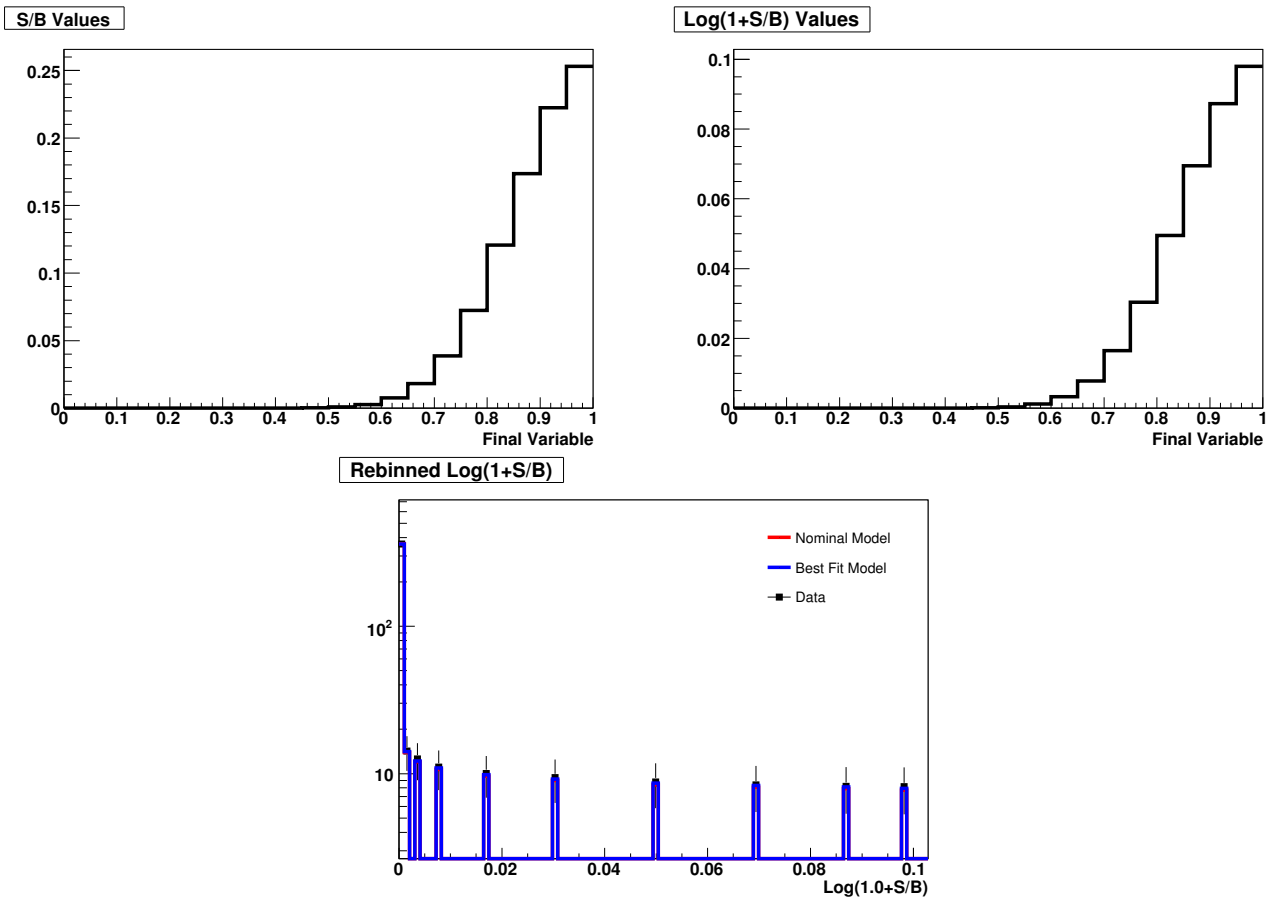
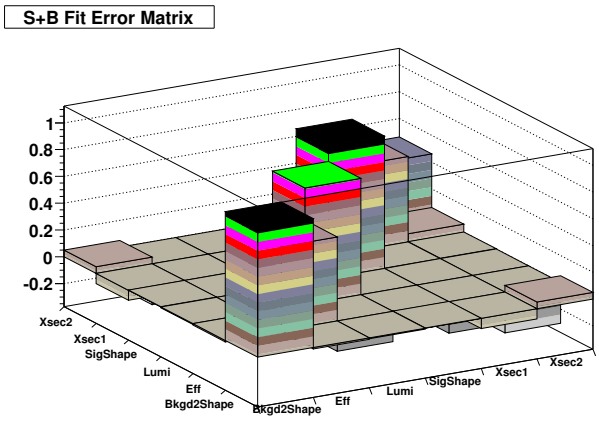
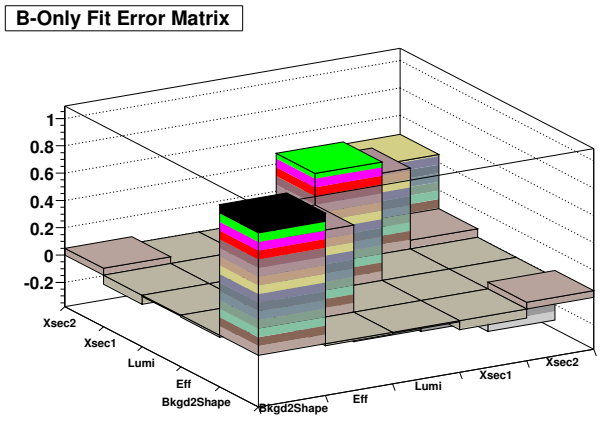


FIG. 17: Signal-over-background and $Log_{10}(1 + S/B)$ distributions for the example analysis.





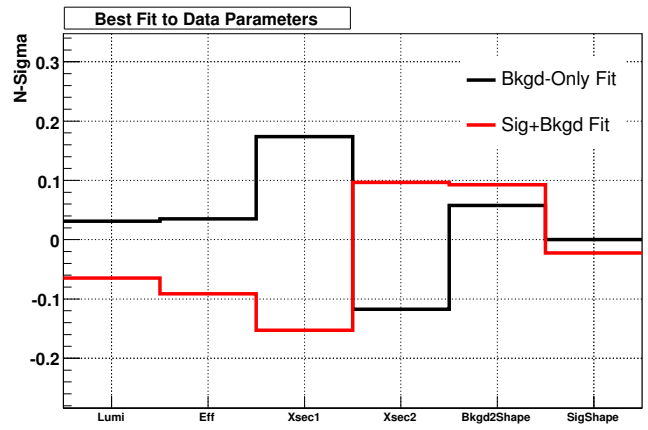
S+B Fit Error Matrix

	Bkgd2Shape	Eff	Lumi	SigShape	Xsec1	Xsec2
Xsec2	0.0498861	-0.305116	-0.180959	0.00255796	0.060902	0.461502
Xsec1	-0.0758935	-0.235554	-0.139703	0.000557648	0.638751	0.060902
SigShape	0.00193992	-0.00195443	-0.00115914	0.988442	0.000557648	0.00255796
Lumi	-0.00659031	-0.226233	0.869357	-0.00115914	-0.139703	-0.180959
Eff	-0.011112	0.628416	-0.226233	-0.00195443	-0.235554	-0.305116
Bkgd2Shape	0.961485	-0.011112	-0.00659031	0.00193992	-0.0758935	0.0498861

B-Only Fit Error Matrix

	Bkgd2Shape	Eff	Lumi	Xsec1	Xsec2
Xsec2	0.0502012	-0.311969	-0.188041	0.0630328	0.460078
Xsec1	-0.0664538	-0.227248	-0.136975	0.590834	0.0630328
Lumi	-0.00853941	-0.21538	0.868523	-0.136975	-0.188041
Eff	-0.0141673	0.638061	-0.21538	-0.227248	-0.311969
Bkgd2Shape	0.956953	-0.0141673	-0.00853941	-0.0664538	0.0502012

FIG. 18: Error matrices for fits to the TEST (left) and NULL (right) hypotheses for the example analysis. The same distributions are shown in three dimensions (top) and numerically (bottom).



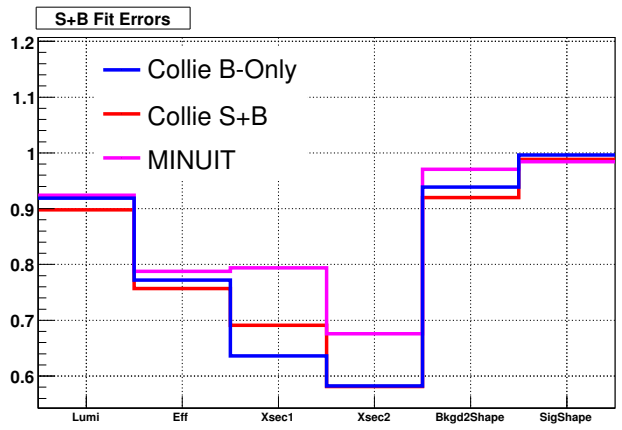


FIG. 19: Best fit values in units of σ for the TEST and NULL hypotheses in the example analysis (left). Estimates of uncertainty size in units of σ (right).

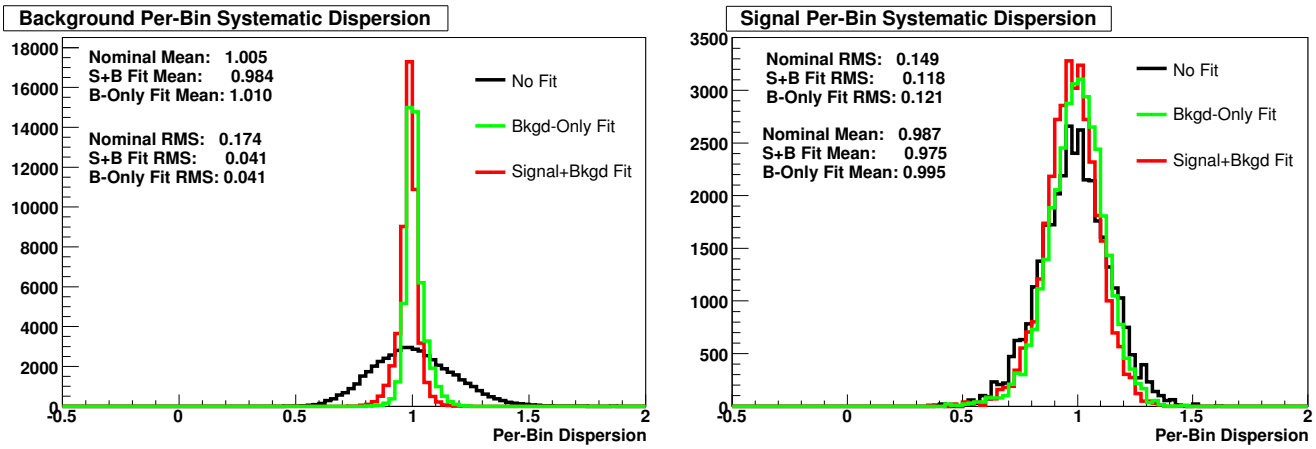


FIG. 20: Post-fit per-bin dispersions for the TEST and NULL hypotheses in the example analysis for background (left) and signal (right).

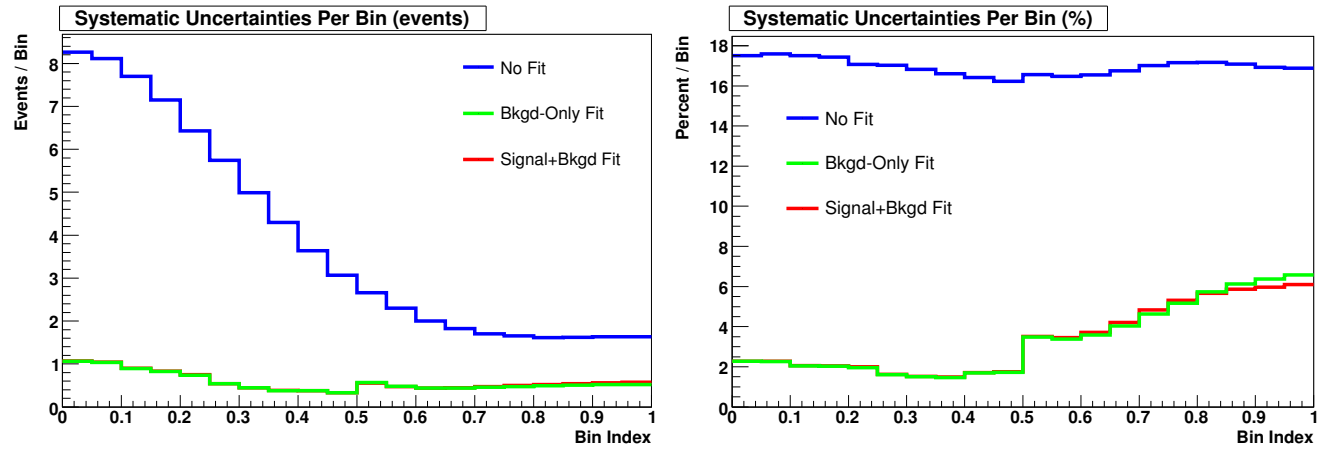


FIG. 21: The RMS uncertainty for each bin in units of the number of total events (left) and the fraction of the bin's nominal prediction (right) before and after fitting.

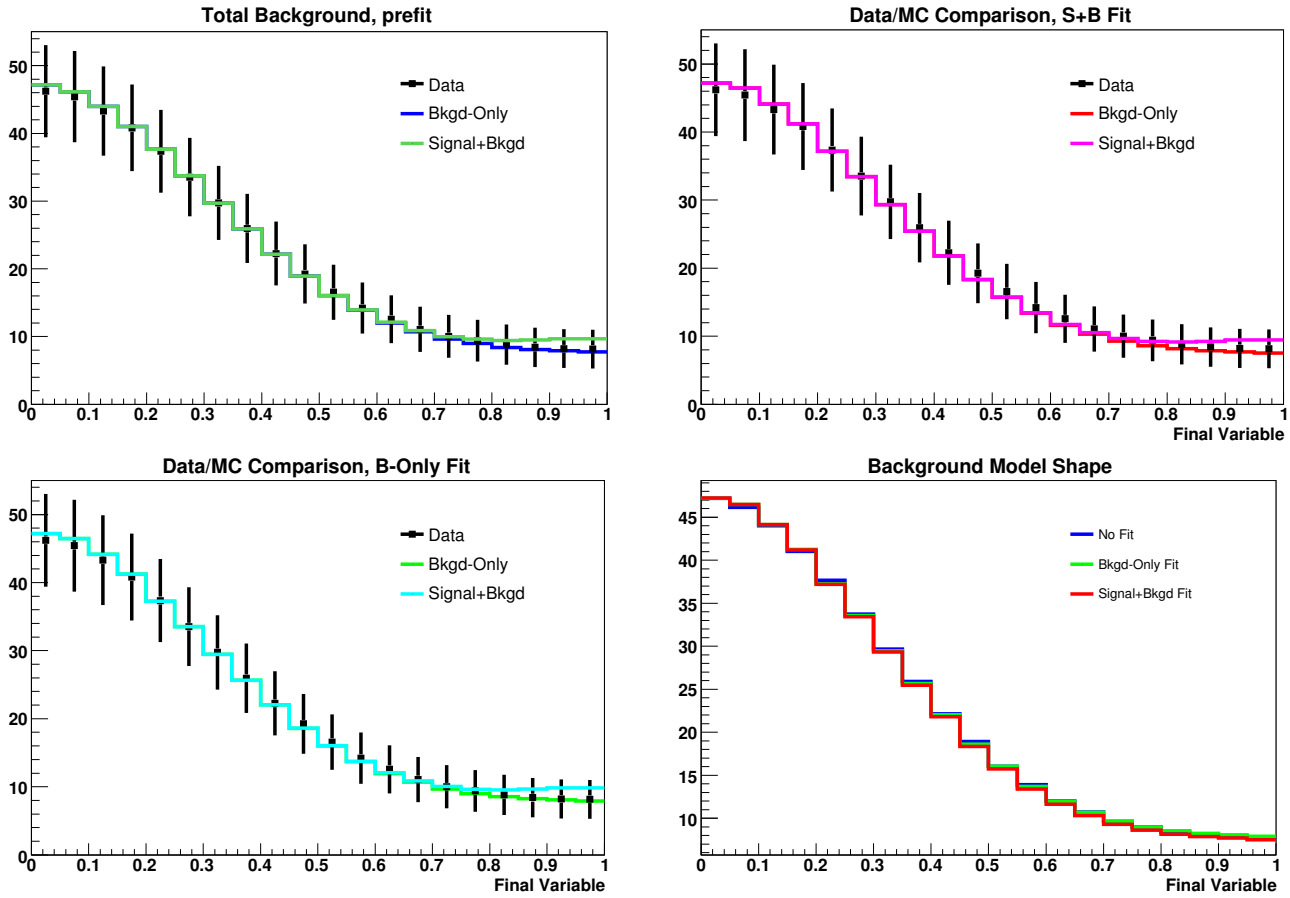


FIG. 22: Signal, background, and data distributions for the nominal predictions (top left), after the TEST hypothesis fit (top right), after the NULL hypothesis fit for the example analysis (bottom left), and a comparison of the three background models (bottom right).

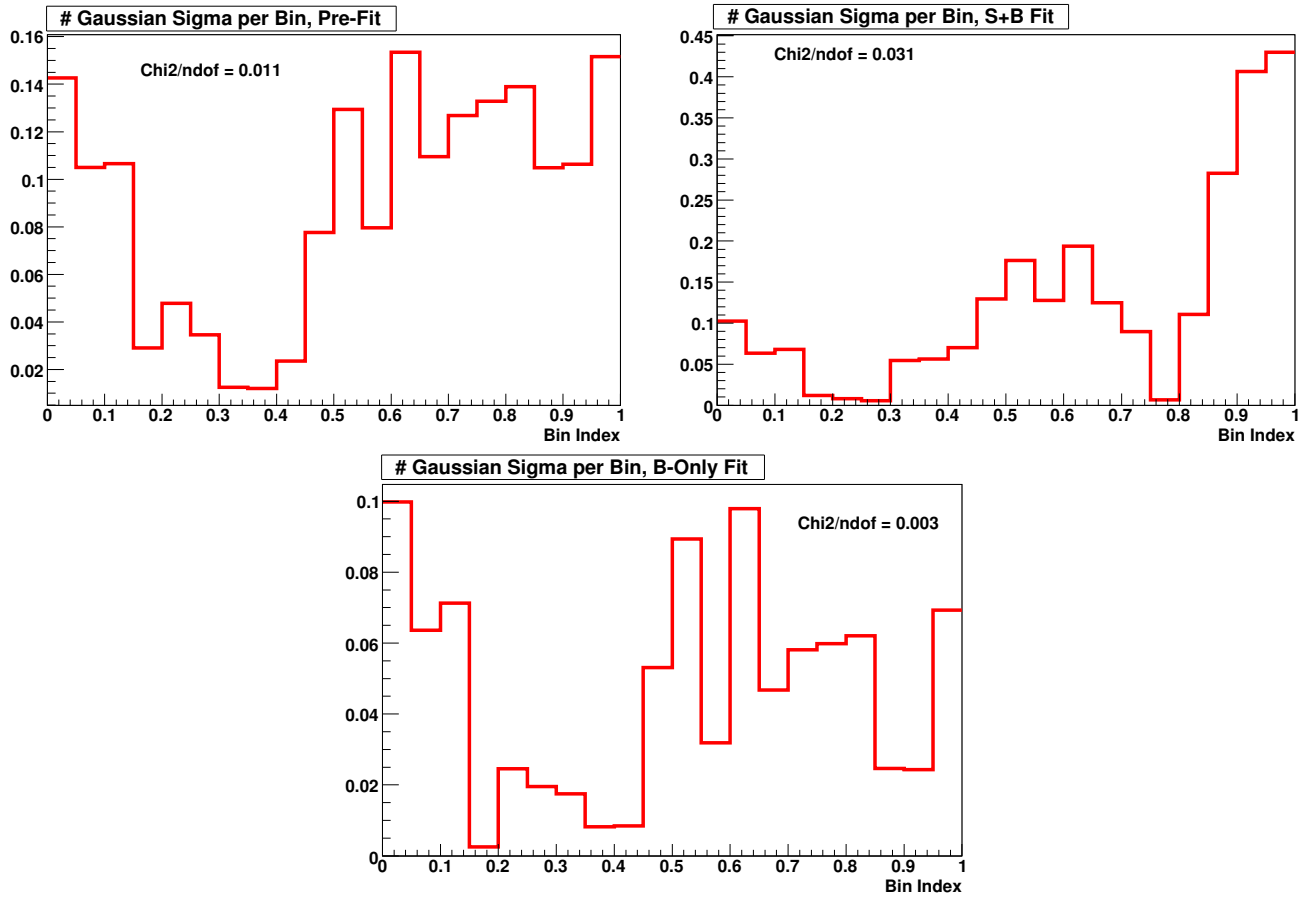


FIG. 23: Per-bin Gaussian χ values for the nominal predictions (top left), after the TEST hypothesis fit (top right) and after the NULL hypothesis fit for the example analysis.

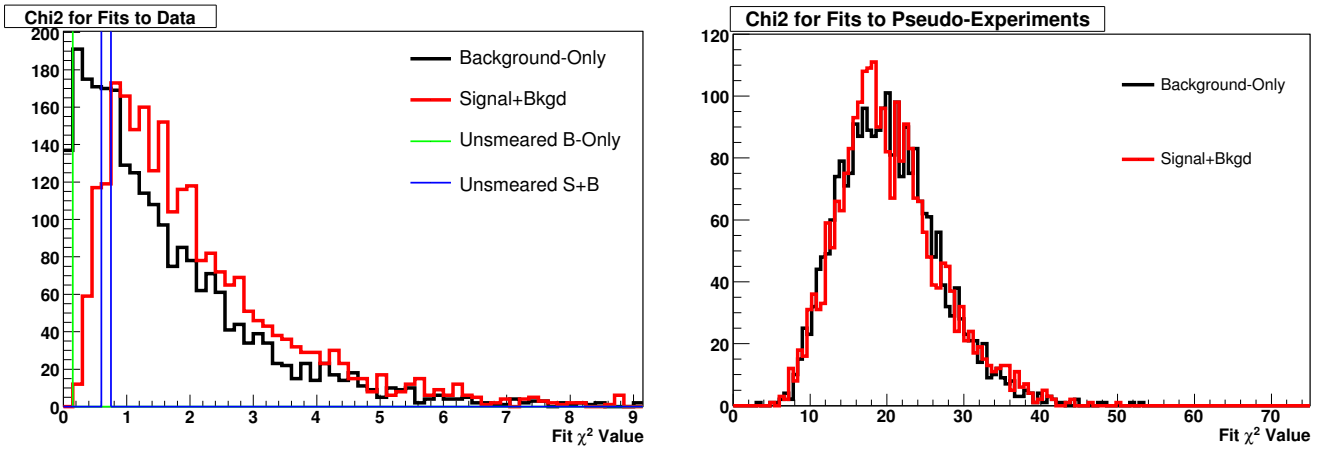


FIG. 24: χ^2 distributions for fits to data (left) and pseudo-data (right) in the TEST and NULL hypotheses for the example analysis.

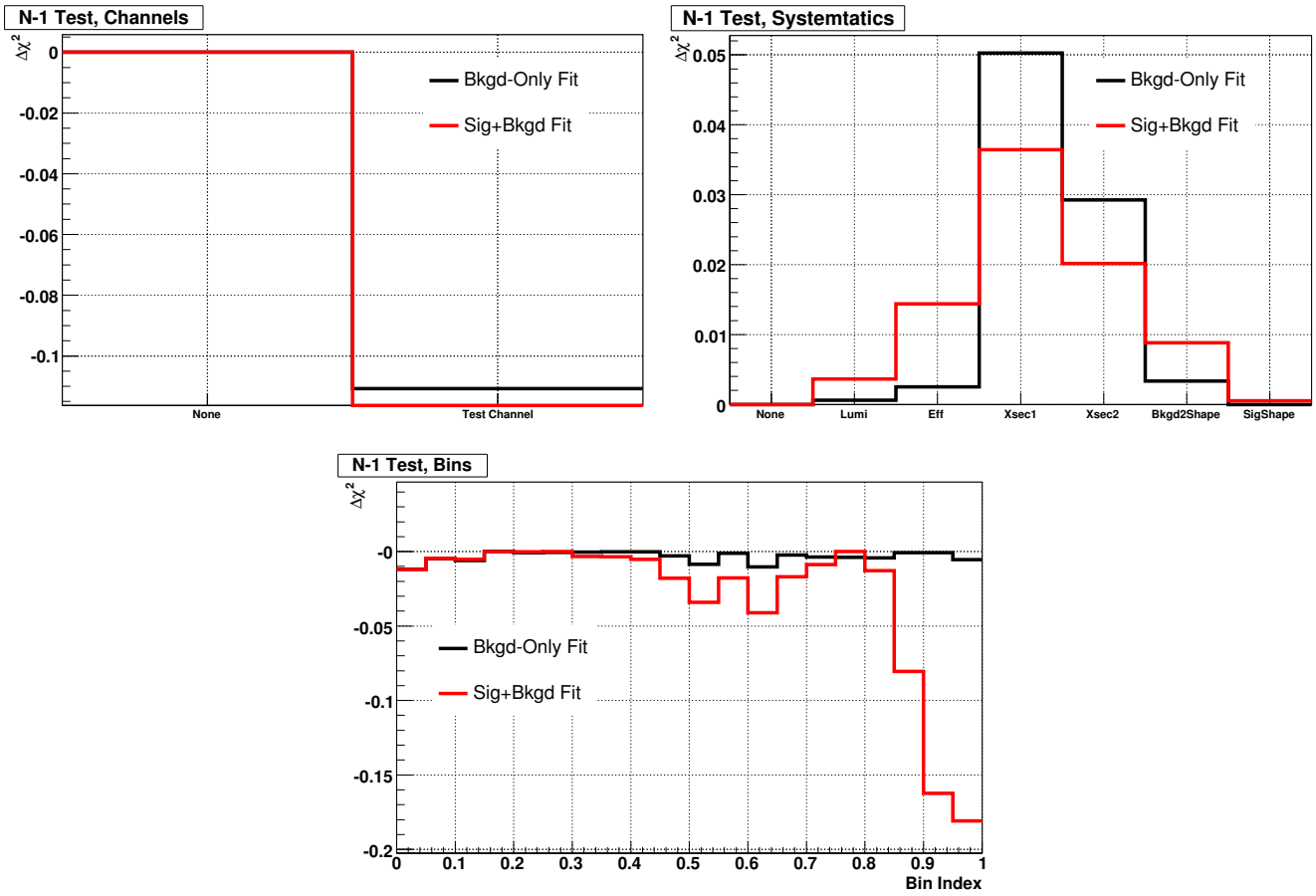


FIG. 25: "N-1" tests of the χ^2 values obtained after removing individual channels (top left), individual systematic uncertainties (top right), and individual bins (bottom) for the example analysis.

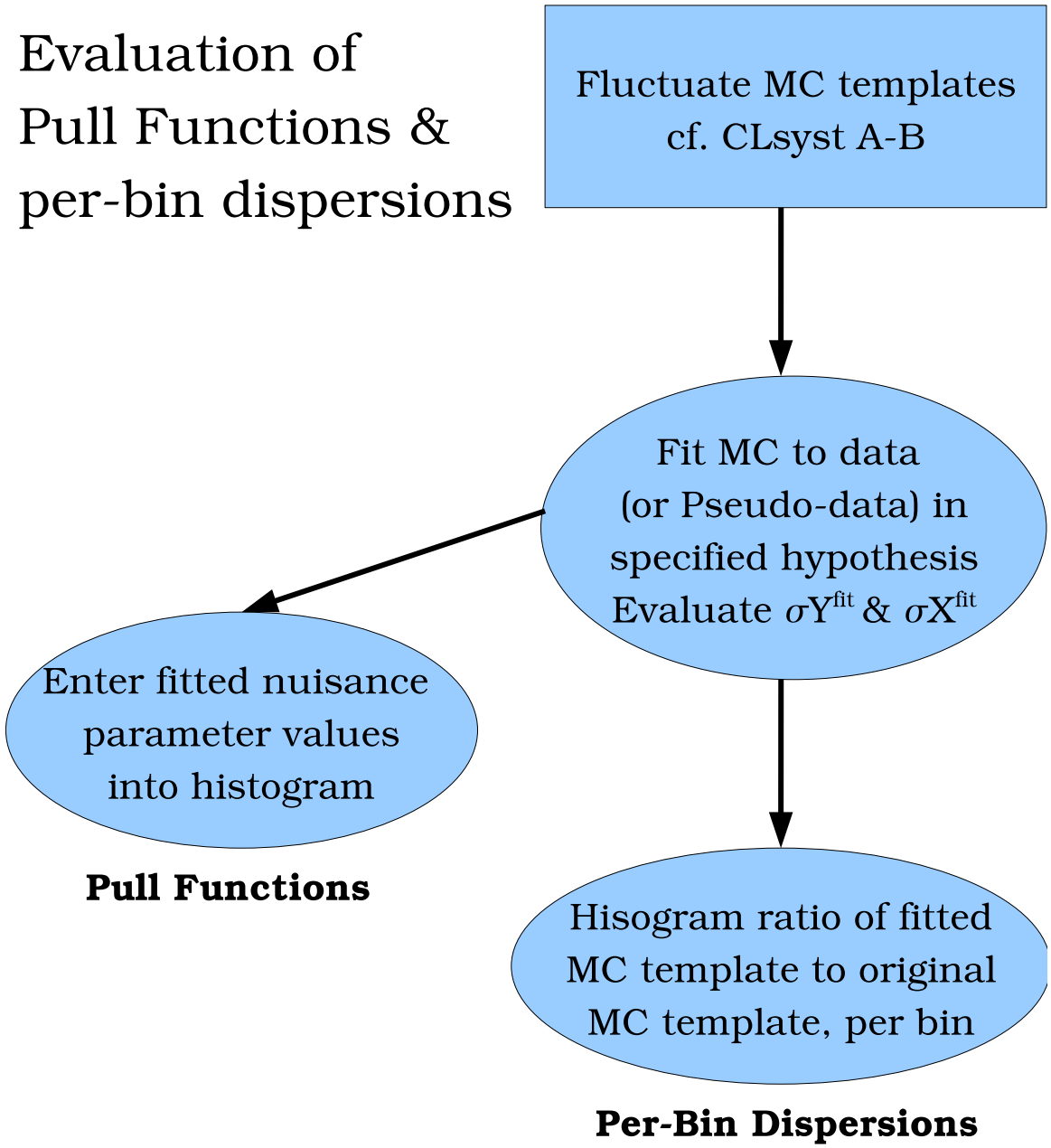


FIG. 26: Flow diagram for the evaluation of pull functions and per-bin dispersions.

IX. VOIDING YOUR WARRANTY

In order to obtain reliable results, COLLIE users should make their best effort to understand the details of all MC templates and systematics given to COLLIE. This section outlines several user-specific aspects that should be treated with care.

- Binning: All input distributions given as input to COLLIE calculations must have the same binning. Histogram binning should be chosen to reflect (1) the expected resolution of the variable and (2) the statistical precision of the MC and data histograms. Choosing a binning too fine for point (1) generally results in randomly-distributed entries in adjacent bins, which are subdividing the nominal resolution. This can increase or decrease search significance in a non-physical manner. By choosing a binning that is too fine for point (2), users can inadvertently create bins in which no background entries are present but with a non-zero signal. This forces COLLIE to limit the NLLR response to this bin as there is no estimate of background rate or background uncertainty. *In general, it is not safe to optimize a limit based on binning.*

- Systematic Uncertainties: The user-supplied systematic uncertainties must correspond to $\pm 1\sigma$ fluctuations of the nuisance parameter in question. It is the sole responsibility of COLLIE users to determine the accuracy of their uncertainty estimations. Overestimating uncertainties will degrade search significance and the degree of degradation can be lessened when using COLLIE's fitting algorithms. Nonetheless, overestimation of uncertainties is not a "safe" option when using the fitting algorithms. Such a choice can violate the $\pm 1\sigma$ assumptions made in the fit model and generate unreliable results. Furthermore, all pseudo-experiments are generated based on the user-input values of nuisance parameter uncertainties. If these are overestimated, then the width of the prior predictive ensemble will be larger than it otherwise would be. This will necessarily degrade the significance of the hypothesis test.

- Systematic Uncertainties: The fitting model specified within COLLIE expects systematic uncertainties to correspond to the change in final variable shape or normalization due to the effects of a given systematic uncertainty shifting by $\pm 1\sigma$. This is not the same as the systematic uncertainty on a given parameter. Due to the complicating nature of selection cuts (which may lie on sharply-falling distributions), it is unsafe to propagate uncertainties without re-evaluating selection efficiencies. In the case of absolute normalizations, this caveat does not apply (*e.g.*, luminosity or cross sections).

- Systematic Uncertainties: The fitting model specified within COLLIE expects systematic uncertainties to be mutually orthogonal for a given source of events (*e.g.*, a single background source). For example, if a background is assigned nuisance parameters associated with the uncertainty of the final variable shape and normalization due to the theoretical model for event production, the shape systematic should not include uncertainty on the absolute normalization.

- Statistical Uncertainties: If users wish to include the statistical uncertainties in their calculation, they must ensure that the per-bin histogram uncertainties as supplied to COLLIE are reliable.

- Shape Uncertainties: Uncertainties that impact the shape of the input variable must be handled with care. Users should inspect the implied change in shape due to such an uncertainty to ensure that it corresponds to a coherent shape change. Many uncertainties are modeled using a random removal or smearing of selection variables in analyses. Such uncertainties often generate a non-coherent, statistical fluctuation in per-bin values and should not be used as shape uncertainties. A more reliable prescription for these instances is to determine the change in overall rate (if any) and report as a flat uncertainty.

- Fit Diagnostics: COLLIE users who utilize the fitting algorithms must perform the fit diagnostics outlined above. These diagnostics are the only tool available to determine the quality of the user-constructed fit model. Failure to inspect these diagnostics may result in unreliable results.

- Fit Diagnostics: Upon inspection of COLLIE's fit diagnostic tests, the analysis of individual pull and χ^2 response functions may indicate that a systematic is being constrained below its nominal prediction by the user's data. This is can be determined from inspection of the pull functions of fits to data. One interpretation of this behavior is that the uncertainty is overestimated. However, this conclusion depends strongly upon the validity of the user-constructed fit model. To study this situation, users should consider the following questions:

  - Are all appropriate systematic uncertainties parameterized in the fit?
  - Are all systematic uncertainties assigned an appropriate size, including theoretical and estimated (unmeasured) values?

- Are all shape-dependent uncertainties appropriately described and do any "flat" systematics require a shape-dependent component?
- Are all correlations assigned correctly?

In cases where a systematic uncertainty is properly constrained below the nominal prediction, the newly constrained values are not generally applicable to other analyses. The results obtained from a well-modeled fit are specific to the system being studied. Thus, a constrained parameter does not *a priori* represent a measured quantity with 68% (1-σ) coverage for all instances, but may indicate a nuisance parameter that needs more study. However, instances can indeed be constructed in which the fit results are applicable outside the studied system.

Getting and Using Collie

This section contains the information necessary to obtain and utilize the COLLIE package.

X. GETTING COLLIE

X.A. Within the DØ Experiment

The source code of the COLLIE package is available in the DØ CVS repository [CVSROOT=d0cvs@cdcvs.fnal.gov:/cvsroot/d0cvs]. The DØ COLLIE release does not depend directly on the SoftRelTools (SRT) code management environment. Rather, it can be compiled and used outside a normal software release directory. Nonetheless, the code does make use of software available via the standard DØ builds in UPS. Users may obtain a copy of the COLLIE package by requesting a specific CVS tag (the most recent tag is V00-04-00):

```
> setup d0cvs
> cvs checkout -r V00-04-00 collie
```

This procedure will create and populate the COLLIE directory structure, including a file with more instructions (collie/README). Users can then perform a build of the software:

```
> cd collie
> source setup setup_Collie.tcsh
> make
```

The setup_Collie.tcsh (setup_Collie.bash) script assumes a TCSH (BASH) shell. Users with different shells will need to adapt the syntax accordingly.

X.B. Outside the DØ Experiment

...coming soon...

XI. USING COLLIE

After following the above instructions, a successful build will generate the following files:

```
collie/lib/libCollieIO.so
collie/lib/libCollieLimit.so
collie/examples/collieIOmaker.exe
collie/examples/collieLimitCalc.exe
collie/examples/collieXsecCalc.exe
```

The COLLIE package is separated into two unique aspects: (1) the generation and manipulation of data files that represent the input to calculations (**CollieIO** files), and (2) the calculation of different statistical products based on input files. Most of the detailed user interface is performed in the CollieIO step, while the calculation step consists of fixed routines. A detailed tutorial is available in .pdf format in the collie/examples/ directory.

XI.A. Generating CollieIO Files

The COLLIE data model expects the user to specify three aspects of the hypotheses to be tested: (1) the observed data, (2) the background-only (NULL) hypothesis, (3) the signal prediction. The TEST hypothesis is constructed by summing the background and signal predictions. These distributions are input via ROOT histograms. All data and background input histograms must be normalized to the absolute number of events predicted. Signal histograms may be normalized to the predicted number of events **or** the predicted number of events divided by the signal cross section. More details on signal normalization can be found below.

Example source code for the generation of CollieIO files can be found in the file "collie/examples/collieIOexample.cc". After compiling, this program can be executed as follows (along with the immediate output):

```
> cd collie/examples
> ./collieIOmaker.exe
==>Created mass point 100
Mass: 100
     Data: 451
     Signal: 7.504
     Bkgd: Bkgd1, 150.000
     Bkgd: Bkgd2, 299.876
     Allbkgd: 449.88
==>Saving inspection histos to fv_exampleCollieIOfile.root
==>Saving channel data to exampleCollieIOfile.root....
```

This example program generates sample data describing the results of a search analysis, which is the same input data used in the above analysis example (Sec. VII). The output of the program is two files: fv_exampleCollieIOfile.root and exampleCollieIOfile.root. The first file (fv_exampleCollieIOfile.root) includes in ROOT-browseable format all the input data available for calculations within COLLIE. The second file (exampleCollieIOfile.root) is the ***CollieIO*** file and is not ROOT-browseable to avoid file size overhead.

Upon inspection of the source code, the important aspects of file creation can be addressed. The first step is to instantiate the CollieIO file and specify the parameters of the histograms that will be provided by the user as input. All methods used in this code are specified in collie/io/include/ColleIOFile.hh.

```
////////////////////////////////////////
///Create IO file with input parameters
////////////////////////////////////////
CollieIOFile* cfile = new CollieIOFile();

//Specify output file and channel name
cfile->initFile("exampleCollieIOfile.root", "Test Channel");

//Define your input histograms
cfile->setInputHist(0.0,1.0,20);

//Option to rebin histograms to a coarser binning (1 = no rebinning)
cfile->setRebin(1);

//Option to smooth histograms
cfile->setSmooth(false);
```

The histogram parameters are pre-specified as a cross check to protect users from entering histograms with parameters not identical to what they intended. Methods are also available for two-dimensional input histograms. The next step is to specify the number of backgrounds and their names.

```
//Define backgrounds
vector<string> bkgdNames;
bkgdNames.push_back("Bkgd1");
bkgdNames.push_back("Bkgd2");

//Generate channel framework
cfile->createChannel(bkgdNames);
```

In this example, there are two backgrounds. This information is sufficient to generate the framework of a single input channel.

The next step for users is to obtain the histogrammed data, background, and signal distributions. In the example, these are filled by hand. However, users may extract the information from an external file or simply include the CollieIO step in their analysis code. The histograms are entered and associated with up to three model parameters:

```
//Backgrounds are passed in via vector
vector<TH1D*> vbkgd;
vbkgd.push_back(bkgd1);
vbkgd.push_back(bkgd2);

//Alpha parameters only matter when smoothing is utilized
// Input values do not matter if you're not smoothing.
// Don't smooth unless you know what you're doing.
vector<double> valpha;

//Each parameter point has a signal histo, data histo, and an array of backgrounds...
cfile->createMassPoint(100, data, sig, -1, vbkgd,valpha);
```

In this example, a single model parameter is specified (100). This parameter is completely arbitrary and can be specified as necessary by the user. Users may include multiple ***MassPoints*** in a single CollieIO file to allow many model parameters to be accessed from a single input file.

At this point, users may specify a set of nuisance parameters and their associated uncertainties (systematics). As outlined in Sec. II III.A, the default parameterization for nuisance parameter PDFs is Gaussian. Gaussian uncertainties are specified by the $\pm 1-\sigma$ deviations in units of the fractional change in bin value (*e.g.*, 5% is given as 0.05). Because positive and negative uncertainties are given separately, the input values take the same sign convention as detailed in Sec. II III.A II.A.2 (*e.g.*, ± 0.05 is given as $+0.05$ in the positive direction and $+0.05$ in the negative direction). Individual nuisance parameters can be correlated between signal and background (or amongst backgrounds) by assigning them identical names. Backgrounds are indexed by the order in which they were introduced to define the channel. Each uncertainty must be specified for each model parameter point individually. The following methods create "flat" systematic uncertainties that have the same value for every bin of the histogram:

```
//Add systematics...either flat or by shape (ie, function of final variable)
// if by shape, must supply a histogram of the values in percent(%) fluctuations...
// Signal requires no index, but backgrounds must be specifically indexed (0->N bkgds)
cfile->createFlatSigSystematic("Lumi",0.06,0.06,100);
cfile->createShapeSigSystematic("SigShape",sigSystP,sigSystN,100);

cfile->createFlatBkgdSystematic(0,"Lumi",0.06,0.06,100);
cfile->createFlatBkgdSystematic(1,"Lumi",0.06,0.06,100);

cfile->createFlatSigSystematic("Eff",0.10,0.10,100);
cfile->createFlatBkgdSystematic(0,"Eff",0.10,0.10,100);
cfile->createFlatBkgdSystematic(1,"Eff",0.10,0.10,100);

cfile->createFlatBkgdSystematic(0,"Xsec1",0.15,0.15,100);
cfile->createFlatBkgdSystematic(1,"Xsec2",0.15,0.15,100);
```

Users may also input uncertainties via histograms with non-constant values per bin. In this case, the same sign convention as above is maintained. A third method is to simply introduce the alternative final variable distributions obtained via the $\pm 1-\sigma$ changes for the nuisance parameter in question.

```
//Example of systematics input as histograms, can be flat or function of final variable
//==>Use this method if you're inputing fractional shape systematics
TH1D* systP = (TH1D*)infile.Get("signal_Systematic_positive");
TH1D* systN = (TH1D*)infile.Get("signal_Systematic_negative");
cfile->createSigSystematic("ShapeSyst",systP,systN,100);

//==>Use this method if you're inputing a different shape template
TH1D* systP = (TH1D*)infile.Get("bkgd_BkgdShape_positive");
TH1D* systN = (TH1D*)infile.Get("bkgd_BkgdShape_negative");
cfile->createShapeBkgdSystematic(0,"BkgdShape",systP,systN,100);
cfile->createShapeBkgdSystematic(1,"BkgdShape",systP,systN,100);
```

Users also have the option of using a log-normal PDF if their nuisance parameter uncertainties are large. This can be specified after a nuisance parameter and its uncertainty have been defined:

```
// Specify that this systematic should have a log-normal PDF (rather than Gaussian)
cfile->setLogNormalFlag("Lumi",true,100);
```

After specifying all input parameters for a channel, the file is written and closed.

```
//store and output channel information
cfile->storeFile();
```

During creation of a CollieIO file, error and warning messages are printed to the screen along with the event check-sum show above. Users should pay careful attention to any messages and make corrections if necessary.

XI.A.1. Statistical Uncertainties

The calculations available within the COLLIE package may be performed with or without including the effects of the uncorrelated per-bin statistical uncertainties of the input histograms. Because histograms are commonly filled with weighted entries, given a series of normalizations, or filled "by hand" with a single entry per bin with the appropriate number of events, COLLIE cannot assume *a priori* that input histograms have valid statistical uncertainties. For this reason, COLLIE places two layers of protection on the use of statistical uncertainties. First, during the CollieIO file generation step each histogram is checked for valid statistical uncertainties and warning messages are printed to the screen for the user. **Second, the use of statistical uncertainties in calculations is turned off by default**. After verifying that they have indeed provided valid statistical uncertainties, users may re-enable the calculation of statistical effects as described below in Sec. XI XI.C.

XI.B. Performing Calculations

Once valid CollieIO files have been generated, users may proceed to perform one or more of the statistical calculations available within the COLLIE package. Example source code for performing confidence level and limit calculations can be found in the file "collie/examples/exampleLimitCalculation.cc". Example source code for performing cross section and significance measurements can be found in the file "collie/examples/exampleXsecCalculation.cc". These examples use the CollieIO file produced by the previous file creation example as input. After compiling the COLLIE package, the example calculation programs can be executed. Each program (*e.g.*, collieLimitCalc.exe) requires two inputs: an output file in which the results will be stored and the model parameter for which the calculations should be performed. An insufficient number of arguments will evoke a correct usage reminder:

```
Using collieLimitCalc.exe:
 collieLimitCalc.exe [ Output ROOT File ] [ Test Point ]
 The test point input is the test variable you wish to look at.
 If you leave off the test point, the code will loop over all
 available points in succession.
```

The code begins by loading the user's input file, checking for formatting errors, and creating an output file for the results:

```
////Open the CollieIOFile created in I/O example step...
CollieLoader loader;
char options[1024]; bool ok = true;

////Specify the filename ...and...
const char* filename = "exampleCollieIOfile.root";
////...the channel name given in the I/O step
sprintf(options,"name='%s'","Test Channel");
```

```
////Pass this information to the loader to extract the channel info
if (!loader.open(filename,option)) {
    cout << "Failed to open " << filename << " using " << options << "!" << endl;
    ok = false;
}
if(!ok) return;

///Create an output container for the results you're about to calculate
TFile f(outfile,"CREATE");
TTree t("SCAN","SCAN");
CLpoint clresults;
clresults.branch(&t);
```

The program will terminate if an error occurs when loading a CollieIO input file.

XI.B.1. Selecting a Calculation Class

Next, the user is allowed to select a calculation class to be used in all statistical calculations. All of the classes derive from the CLcompute class and have a common interface. The CLcompute classes are designed to calculate confidence levels based on a specified set of input distributions. By default the CLfast class is chosen, however this class should only be used for quick tests of input files. The algorithms used in each calculation class are described in Sec. V.

```
// Choose a systematics treatment...
// The CLfast computation uses no systematics.  This class should only be used for
testing purposes.
CLfast clcompute;

// The CLsyst computation applies all systematics via Gaussian distribution
CLsyst clcompute;

// Use CLfit2 for profileLH fitting of systematics-smeared distributions using two fits
per pseudoexperiement
CLfit2 clcompute;

//Use CLfit for profileLH fitting of systematics-smeared distributions using just one
fit per pseudoexperiment
CLfit clcompute;
// If you choose the CLfit option (faster but less powerful than CLfit2), you must
// specify whether the fit will include signal contributions.  If
// not, you must specify at which level to exclude signal bins.
// The cutoff is calculated in terms of log(1+s/b) and the default
// value is 0.005 (ie, remove bins if log(1+s/b)>0.005.
clcompute.fitSignal(false);
clcompute.logSigExclusion(0.005);
```

XI.B.2. Selecting Model Parameter Distributions

This example step demonstrates how to loop over the model parameter points in a file and extract the set of input distributions associated with a specific parameter point. This loop is common to all calculations.

```
//extract the total number of masspoints in the file
int len=loader.getNMasspoints();
```

```
if (len<=0) {
  cout << "Cannot handle loader with " << len << "masspoints" << endl;
  return;
}

//create list of mass point indices
int *v1; v1=new int[len];
int *v2; v2=new int[len];
int *v3; v3=new int[len];
loader.getMasspointList(len,v1,v2,v3);

//loop over all masspoints and perform calculations
for (int i=0; i<len; i++) {
  if(v1[i]==mass || mass==-1){
//tell the container what point you're working on
    clresults.reset(v1[i],v2[i],v3[i]);

//Extract the signal & background distributions associated with this point
    SigBkgdDist* sbd=loader.get(v1[i],v2[i],v3[i]);

//....include your calculation here....//

    t.Fill();
    delete sbd;
  }
}
```

The SigBkgdDist class is an interface container that holds the information for all the signal, background, data, systematic uncertainty, and PDF specifications for a given model parameter point. The instance of this class created for the model parameter point being tested will be passed to the calculators performing the calculations.

XI.B.3. Calculating Confidence Levels

The next step is to perform a calculation. This example step demonstrates how to calculate confidence levels (p-values). However, this calculation is not required for the calculation of confidence limits. Users may remove the confidence level calculation to save time if they are not interested in the results. At this point, users have already chosen a CLcompute class and may perform a CL calculation. Insert this code into the parameter point loop.

!!! Inside the model parameter loop:

```
//calculate Confidence Levels
clcompute.calculateCLs(*sbd,clresults,CLcompute::LEVEL_VERYFAST);

//report your results for interested observers
clresults.print();
```

The calculation performs a fixed number of pseudo-experiments, as specified in the "calculateCLs(...)" method. The number of pseudo-experiments varies from class to class and is outlined in Table IV. The results can be printed in a summary format as well.

Histograms of the test statistic (NLLR) can be extracted and viewed in Root format by adding the following lines of code after the calculation has been performed:

```
//Add the following lines...
int bins = 500; double min = -50; double max = 50;
TH1D* sigLLR = clcompute.getLLRdist_sb("NLLR_SB",bins,min,max);
```

Specifier	CLfast	CLsyst	CLfit	CLfit2
CLcompute::LEVEL_VERYVERYFAST	5.0×10^3	5.0×10^3	5.0×10^3	5.0×10^3
CLcompute::LEVEL_VERYFAST	1.5×10^4	1.5×10^4	1.5×10^4	1.5×10^4
CLcompute::LEVEL_FAST	2.5×10^4	5.0×10^5	5.0×10^4	2.5×10^4
CLcompute::LEVEL_STANDARD	5.0×10^4	1.0×10^5	1.0×10^5	5.0×10^4
CLcompute::LEVEL_FINE	1.0×10^5	1.0×10^6	2.0×10^5	1.0×10^5
CLcompute::LEVEL_VERYFINE	2.0×10^5	5.0×10^6	5.0×10^5	2.0×10^5
CLcompute::LEVEL_VERYVERYFINE	5.0×10^5	2.5×10^7	1.0×10^6	5.0×10^5

TABLE IV: The number of pseudo-experiments generated for each specifier and for each calculation class.

```
TH1D* bkgLLR = clcompute.getLLRdist_b("NLLR_B",bins,min,max);
TH1D* LLRd = new TH1D("NLLR_D","NLLR_D",bins,min,max);
TH1D* LLRsigma1 = new TH1D("NLLR_B_1sigmas","NLLR_B_1sigmas",bins,min,max);
TH1D* LLRsigma2 = new TH1D("NLLR_B_2sigmas","NLLR_B_2sigmas",bins,min,max);

LLRd->Fill(clresults.llrobs);
LLRsigma2->Fill(clresults.llrb_m2s);
LLRsigma1->Fill(clresults.llrb_m1s);
LLRsigma1->Fill(clresults.llrb_p1s);
LLRsigma2->Fill(clresults.llrb_p2s);
```

Users must provide the desired number of bins and range for the histograms.

XI.B.4. Calculating Confidence Limits

Confidence limits are calculated using the CrossSectionLimit class. This class requires the user to specify a computation class. Users may also specify the following parameters:

- The confidence level (CL) for which the limit will be calculated: *default = 95%.*
- The CL accuracy of the limit calculation. This value specifies the requirement the search algorithm must satisfy: $-CL^{accuracy} < \left(CL^{measured} - CL^{required}\right) < CL^{accuracy}$, *default = 0.001, or 0.1%.*
- The precision of the limit calculation. This value determines the relative number of pseudo-experiments used in the limit finding algorithm. A larger number of pseudo-experiments achieves a higher precision: *default = 0, maximum=4.*
- The number of standard deviations (Nsigma) away from the background prediction for which the expected limit will be calculated. The expected limit can be calculated using different reference points for the integration: *default = 0-sigma.*
- The cross section scaling factor search seed. The search algorithm determines the value of the limit by moving a multiplicative factor for the signal cross section. The starting point can be set at any value. Setting a value near the true value will increase the speed of the algorithm: *default = 1.0.*

The algorithm calculates both the expected and observed limits by default. Users may choose to calculate only one at a time to increase the speed of the calculation. The following instantiation and parameter setup should be performed outside the model parameter loop.

!!! Outside the model parameter loop:

```
/// This is the class for computing cross section limits
CrossSectionLimit csLim;
csLim.setup(\&clcompute);
```

```
//Verbosity switch
csLim.setVerbose(false);

//95% CL is the default value
csLim.setCLlevel(0.95);

//The range of CL values that will satisfy the algorithm:  -0.001 < (CL-0.95) < 0.001
csLim.setAccuracy(0.001);

//Toggle the number of pseudo-experiments used to find the limit:  0 is
lowest(fastest), 4 is highest(slowest)
csLim.setPrecision(0);

//Toggle expected/observed to speed things up if you wish
csLim.calculateExpected(true);
csLim.calculateObserved(true);

//Calculate the expected limit in the case of -2,-1,0,1, or 2-sigma variations of the
data relative to bkgd
csLim.setNSigma(0);

//Start the cross section limit search at a cross section of 1.0 times the nominal
input value
// Use this to shorten your calculation if you know roughly where the limit will be.
csLim.setSearchSeed(1.0);
```

After setting up the CrossSectionLimit class, users can calculate a cross section limit by specifying a SigBkgdDist instance obtained in the model parameter selection loop. Following the calculation, a summary of results and settings can be printed. Informational and warning messages will be printed to the screen as well.

!!! Inside the model parameter loop:

```
//Calculate a cross section limit...
//These results are reported in the factor by which you must
//multiply your nominal signal cross section to obtain a 95% CL
//upper limit for this model...  IE, multiply this factor by
//your model xsec to get your limit in barns
csLim.calculate(*sbd,clresults);

//report your results for interested observers
csLim.print();
```

XI.B.5. Performing a Cross Section Calculation

Cross section measurements are performed using the CrossSectionCalc class and are described in Sec. VI VI.A. Begin by creating an instance of the class outside the model parameter loop:

!!! Outside the model parameter loop:

```
/// This is the class for computing cross section limits
CrossSectionCalc csCalc;
csCalc.setup(&clcompute);
csCalc.setVerbose(false);
```

Inside the parameter loop, the calculation is performed using the selected SigBkgdDist instance. Printing the result will obtain details of the fit results including the fitted signal cross section, its error, and the best fit values of all nuisance parameters along with their errors in units of the input $1-\sigma$ systematics.

!!! Inside the model parameter loop:

```
//Calculate a cross section
// The signal rate is floated as a free parameter
// The resulting fit gives you the fitted xsec in units of
// the input cross section.
csCalc.calculate(*sbd,clresults);
//report your results for interested observers
csCalc.print();
```

XI.B.6. Cross Section Significance Calculations

To perform cross section significance calculations, the CrossSectionCalc class is used again. Users specify the SigBkgdDist instance for calculation, the signal cross section scaling factor to be used for generating pseudo-experiments, and the number of pseudo-experiments to generate. Specifying a signal cross section scaling factor of 0.0 will result in no signal contribution in the pseudo-experiments and will correspond to the Null hypothesis significance. Alternatively, the Test hypothesis significance can be calculated by using a signal scaling factor of 1.0. Any other floating point number can also be specified.

The results are stored in the output file in histograms with a range of 0-8 times the nominal cross section value and 2000 bins. These histograms can be integrated to determine p-values for the generated scenario. The interpretation of these results is described in Sec. VI VI.B.

!!! Inside the model parameter loop:

```
//Perform a significe test for your signal cross section
// The second parameter determines the signal cross section size
// to be used in the generation of pseudo-experiments (0.0 for NULL hypothesis, 1.0 for
TEST hypothesis)
// The third parameter determines how many pseudo-experiments will
// be generated for the test.  The results are stored in the output file.
csCalc.testFitPE(*sbd, 0.0, 10000);
```

XI.B.7. Performing a Fit Test

Users who wish to use the fitting classes in Collie are advised to perform a fit test to generate diagnostic information on the quality of their fit model. The details of the fit test are described in Sec. VIII. Users may perform a test as follows. An example of the FitTest class instantiation can be seen in the file "collie/examples/exampleLimitCalculation.cc":

!!! Outside the model parameter loop:

```
// This class is used to test the fit used by the CLfit and CLfit2 classes
// Use this to determine the quality of your fit model.
FitTest fitTest;
// Set the number of pseudo-experiments to fit
fitTest.setIterations(2000);
// Determine if you want fitted pseudo-experiments in the tests
fitTest.testPE(true);
```

Users must choose how many fit iterations they want to perform, which will determine the statistics of the fit test pull functions. Users may also choose to perform fits to pseudo-experiments. The default is to perform fits both of smeared MC to data and of MC to pseudo-experiments. See Sec. VIII for more details. Next, the fit can be performed by inserting the following lines into the model parameter loop:

!!! Inside the model parameter loop:

```
fitTest.runTest(sbd,0.005);
continue;
```

Here, the value 0.005 determines the which bins will be removed in the fits based on the values of $log_{10}(1 + s/b)$ for each bin. Setting this number to a large value (*e.g.*, 1×10^6) will ensure all bins are included in the fit. In this example, the loop exits following the fit test in order to ensure the test is performed for only one model parameter point. The fit test diagnostics are not designed to properly include multiple model parameter points.

After performing the fit test, the output file will contain a number of histograms summarizing the results of the fitting diagnostics. These results can be viewed using the ROOT macro "makePlots(...)" in the file "collie/limit/macro/fitResults.C". Details on these results and their interpretation can be found in Sec. VIII.

XI.B.8. Combining Channels

Users may combine multiple channels in the calculation step. To do so, users must open a COLLIELOADER for each input file. In the model parameter loop, the input channels may be appended if both loaders have a valid set of distributions for the model parameter point in question. The following example combines two channels:

!!! Outside the model parameter loop:

```
CollieLoader loader1;
CollieLoader loader2;
char options[1024]; bool ok = true;

const char* filename = "collieIOfile1.root";
sprintf(options,"name='%s'","Test Channel 1");
if (!loader1.open(filename,option)) {
    cout << "Failed to open " << filename << " using " << options << "!" << endl;
    ok = false;
}

filename = "collieIOfile2.root";
sprintf(options,"name='%s'","Test Channel 2");
if (!loader2.open(filename,option)) {
    cout << "Failed to open " << filename << " using " << options << "!" << endl;
    ok = false;
}
if(!ok) return;
```

!!! Inside the model parameter loop:

```
//Extract the signal & background distributions associated with this point
SigBkgdDist* sbd1=loader1.get(v1[i],v2[i],v3[i]);
SigBkgdDist* sbd2=loader2.get(v1[i],v2[i],v3[i]);
if(sbd1 && sbd2) sbd1->append(*sbd2);
else cout << "Failed to append a channel" << endl;
```

Once channels have been appended, all remaining calculations can be performed with no other modifications. All channels being appended must have unique channel names, as no channel may be appended to itself. Upon performing the append operation, COLLIE will perform an inspection of the systematic uncertainties associated with each channel. Uncertainties with identical names will be correlated at this point. Any specifications for floating systematics or log-Normal systematics will be maintained. However, a systematic common to both channels must either be fully floated or not floated at all. If any channel requests a floating nuisance parameter, that parameter will be floated for all channels.

XI.B.9. Viewing Results

The results of all calculations are stored in the output ROOT file specified by the user. The results can be accessed as follows:

- Confidence level calculations: The results of the confidence level calculations are stored in ROOT TTREE format. The default name given to the TTREE is "SCAN" and is specified in the example limit calculation program. If calculations are performed for several model parameter points, the confidence level variables can be plotted as function of the model variable. Examples for plotting various quantities are given in the following ROOT macros:
  - CL_b: collie/limit/macro/plotCLb.C
  - CL_s: collie/limit/macro/plotCL.C
  - NLLR values: collie/limit/macro/plotLLR.C
- Confidence limit calculations: The results of confidence limit calculations are stored in the same manner as confidence level results. An example macro for plotting limit results can be found in collie/limit/macro/plotFactor.C
- Fit Results: As noted above, the fit test results are stored as ROOT-browseable histograms in the output file. The results may be viewed using the ROOT macro collie/limit/macro/fitResults.C.

XI.C. The Collie Novice Flag

By default, the COLLIE package makes certain methods and techniques off limits to users. The assumption made is that novice users must gain familiarity and experience with the code interface before proceeding with these methods. This novice flag applies to both the CollieIO file creation step and the calculation steps. By turning off the novice flag, users are making the explicit statement that they are comfortable with their understanding of the methods in question. The methods that are off limits and the required understanding for each are listed here:

- Interpolation: Interpolation of input histograms to generate non-simulated parameter points is an advanced feature not described in this manual. Users should communicate directly with the author if they wish to use this feature. The COLLIE package implements an algorithm for the linear interpolation of histograms as outlined in [15].
- Smoothing: Smoothing input histograms to generate non-simulated parameter points is an advanced feature not described in this manual. Users should communicate directly with the author if they wish to use this feature. The COLLIE package provides a histogram smoothing algorithm based on Gaussian kernel estimation as described in [16]. This algorithm is a better alternative to that provided in the ROOT histogramming software package.
- Statistical Uncertainties: By default, all statistical uncertainties are ignored. Users must ensure that the per-bin statistical uncertainties specified in their input histograms are correct. If the novice flag is turned off, the COLLIE package will perform tests of the histogram statistical uncertainties and warn users who may be violating proper usage.
- Fitting Calculations: By default, all fitting calculations cannot be used (*i.e.*, CLFIT and CLFIT2 classes). Users must be certain of the accuracy of their nuisance parameter model and its associated uncertainties before proceeding. Careful inspection of the fit test results is a requirement for using the fitting calculations.

When users are comfortable that they are prepared to utilize these methods, the novice flag can be unset via the following methods. Users must also turn on the histogram statistical uncertainties as shown:

```
//In the CollieIO step:
cfile->setNoviceFlag(false);

//In the calculation step:
//All calculators derive from the CLcompute class (eg, CLfast)
clcompute->setNoviceFlag(false);

//Use of histogram statistical uncertainties can be enabled as follows:
clcompute->useHistoStats(true);
```

Acknowledgments

The COLLIE software package was designed as a tool for calculating confidence levels during the end of LEP II collider operations. The earliest form was written primarily by Jeremy Mans at Princeton University. Much of the original design philosophy remains. COLLIE was adapted for use at the Tevatron by Wade Fisher via the inclusion of a treatment for systematic uncertainties. This version was later updated to include the aspects of fitting.

The current version of COLLIE has been reviewed by the DØ statistics committee, during which time many insights and design aspects were brought to being. In particular, Jim Linnemann and Harrison Prosper have provided essential feedback and suggestions to make COLLIE a robust and well-documented tool for general use.

---

[1] C. Amsler et al., Phys. Lett. B **667**, 1 (2008).
[2] G.J. Feldman and R.D. Cousins, Phys. ReV. D **57**, 3873 (1998).
[3] J. Linnemann, M. Paterno and H. B. Prosper, "Calculating Confidence Limits", DØ Note #4491.
[4] R.T. Cox, Am. J. Phys. **14**, 1 (1946).
[5] K.S. Cranmer, "Frequentist hypothesis testing with background uncertainty," 2003. PhyStat2003, SLAC physics/0310108.
[6] R.D. Cousins and V.L. Highland, "Incorporating systematic uncertainties into an upper limit," Nucl. Instrum. Meth., **A32**, 331-335 (1992).
[7] J. Linnemann, "Matching Lognormal and Gaussian Shapes for Systematic Uncertainties", DØ Note #5648.
[8] G.E.P. Box, M.E. Muller, "A note on the generation of random normal deviates," Annals Math. Stat. **29** 610-611 (1958).
[9] R. Barlow, eConf **C030908** WEMT002 (2003) [arXiv:physics/0401042].
[10] T. Junk, Nucl. Intrum. Meth. A **434**, 435 (1999).
[11] A. L. Read, J. Phys. G **28**, 435 (2002).
[12] W. Fisher, "Systematics and Limit Calculations", DØ Note #5309.
[13] James, F. 1998, MINUIT, Reference Manual, Version 94.1, CERN, Geneva, Switzerland.
[14] Ridders, C. F. J. "A New Algorithm for Computing a Single Root of a Real Continuous Function." IEEE Trans. Circuits Systems **26**, 979-980 (1979).
[15] A. Read, Nucl. Instrum. Meth. A **425**, 357 (1999).
[16] K. S. Cranmer, Comput. Phys. Commun., **136**, 198 (2001) [arXiv:hep-ph/0011057].
[17] Because the signal cross section is the parameter of interest in nominal confidence level calculations, there is no uncertainty associated with it. Including a signal cross section uncertainty violates the interpretation of the result. However, one may include an efficiency uncertainty that is unique to the signal.