RooStats validation results

Giovanni Petrucciani
(Univ. of California, San Diego)

Outline

- Our interface to RooStats
- Validation results
  - Profile Likelihood method
  - Bayesian methods
  - Hybrid methods
- Summary
- A full table with all the numbers is also attached to the agenda

CMS Higgs interface to RooStats

A binary package wrapping RooStats

- Takes as input either a simple text datacard for counting experiments (same format as L&S) or any RooStats HighLevelFactory file.
- Configures and runs RooStats methods, prints results and saves them to root files.
- Takes care of generating toys for expected limits, or averaging results of multiple runs.

Validation: code

- Validation done on ROOT 5.27.06 (64bit/gcc434)
- Another set of validation will be done on 5.28 once it's integrated with the CMS software. (but we already started to look at it)
- We tested mainly the following routines:
  - ProfileLikelihoodCalculator
  - BayesianCalculator, MCMCCalculator
  - HybridCalculatorOriginal
  - We had a quick look also at the new HybridCalculator, but not at the new features only in 5.28.

Validation: observables

The validation was twofold:

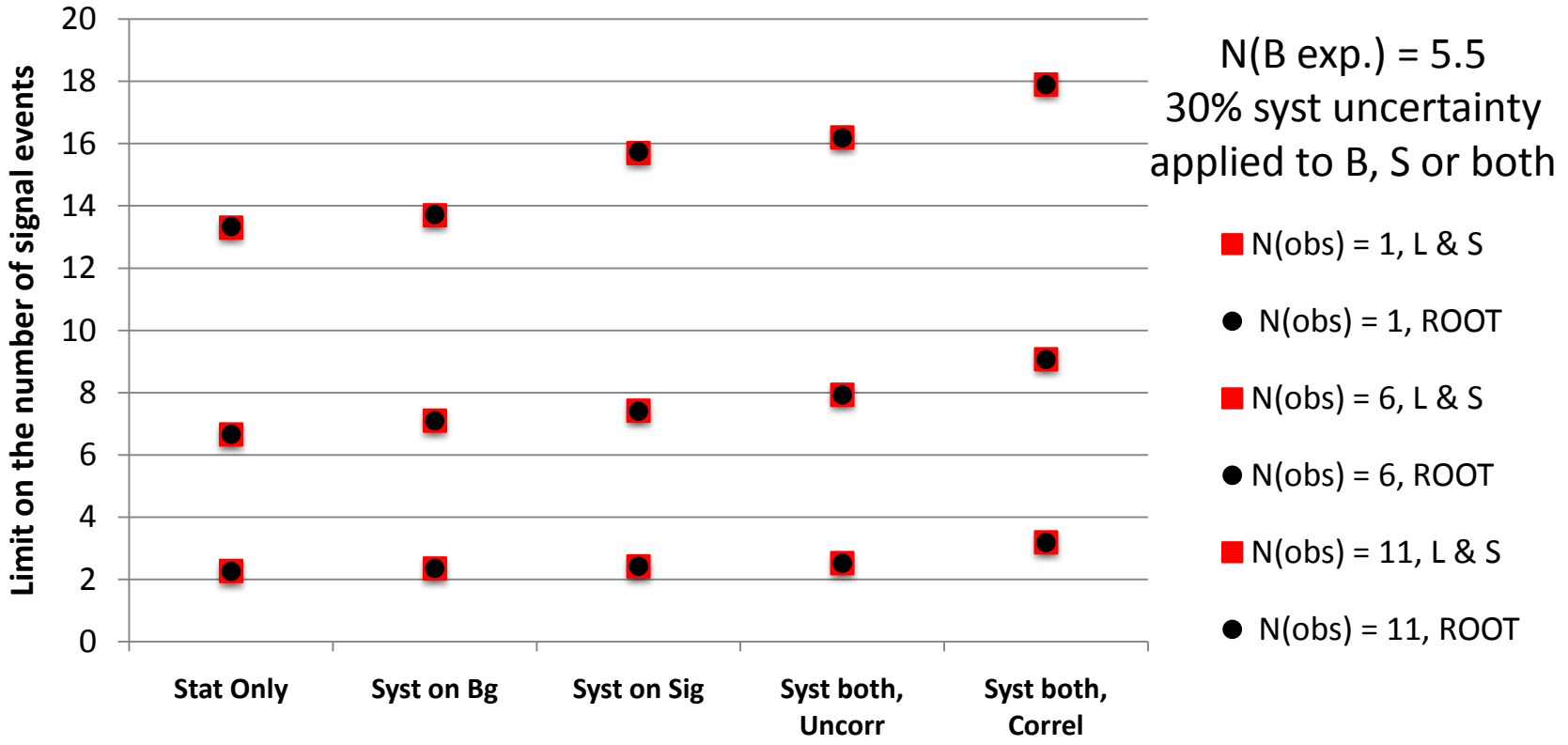
- Check that we get results in agreement with the reference ones from L&S, or from semi-analytical results for the simple cases.
- Check what can be computed with a reasonable amounts of resources per job.
- For the second part, we should probably follow up on the technicalities with roostas developers (not in this meeting).

Profile Likelihood method

- **The results agree to high precision with what computed with the independent tool L&S.**
- **It's fast, below 1s even for complex models.**
- On rare occasions we've found numerical issues that cause the computation to fail. Solved switching between Minuit2 and Minuit, or to making very tiny changes to the variable ranges or inputs.

Profile Likelihood method

Comparison for simple counting experiments with different n(obs) and syst. uncertainties



Bayesian Methods: BayesianCalculator

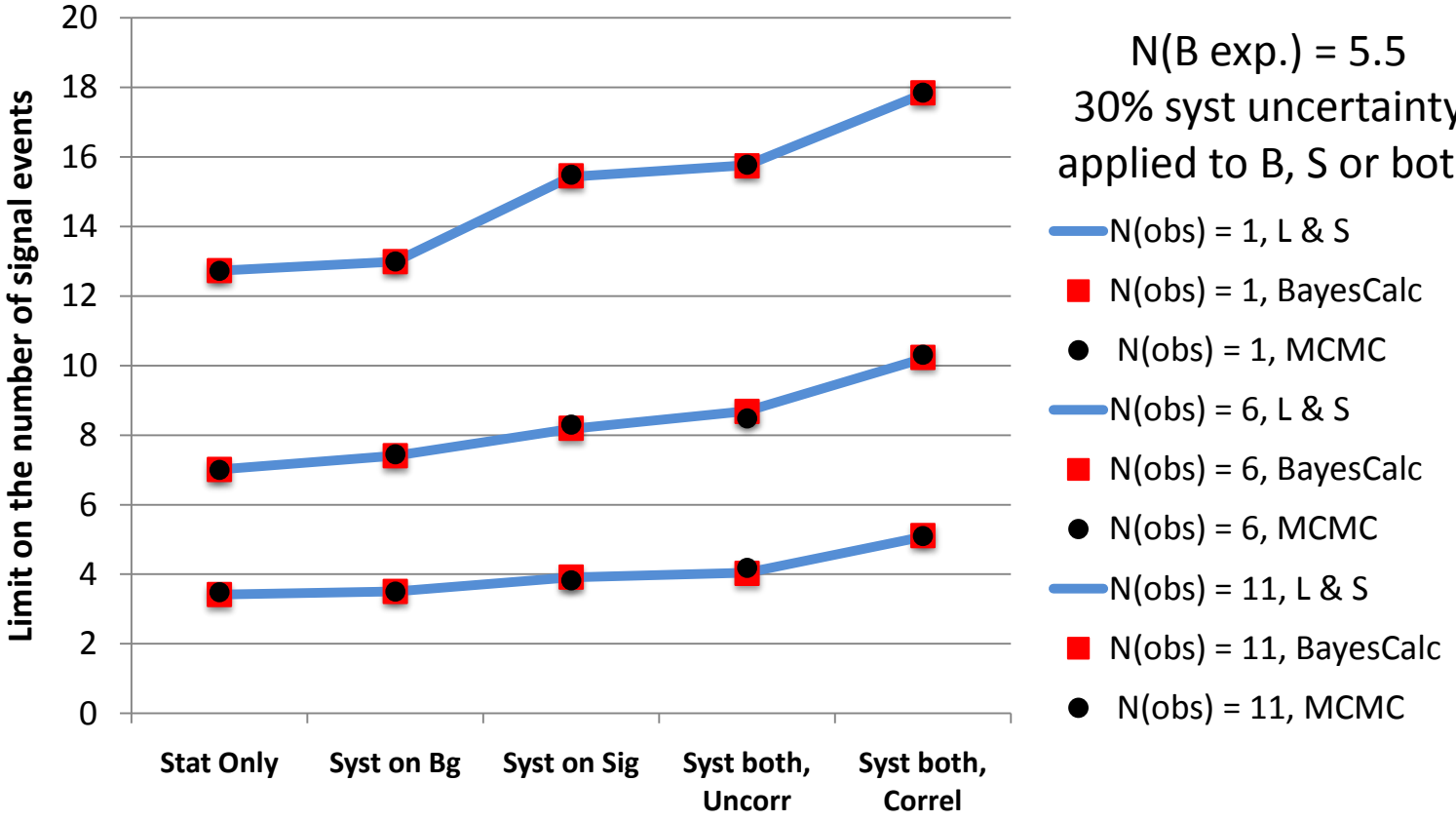
- **Allows only very few nuisance parameters.**
- **In the simple counting experiment tests, the results in good agreement with L&S, both for flat prior and prior prop. to 1/sqrt(σ)**
- Whenever it works, the performance is good.
- Anyway, due to the limitation, it won't be of much use for any real higgs combination that includes systematical uncertainties

Bayesian Methods: MCMC

- The MCMCCalculator is what we normally use for Bayesian limits for complex models.
- There are lots of configurable options; we're not experts of MCMC so our approach has been "try and see what seems to works best". **The choice of proposal function seems the more critical.**
- **Whenever MCMC works, the results are in good agreement with L&S for the two tested priors.**
- **One point also cross checked with BAT using the same Workspace. Good agreement found.**

Bayesian

On simple experiments, good agreement between L&S, BayesianCalculator and MCMCCalculator

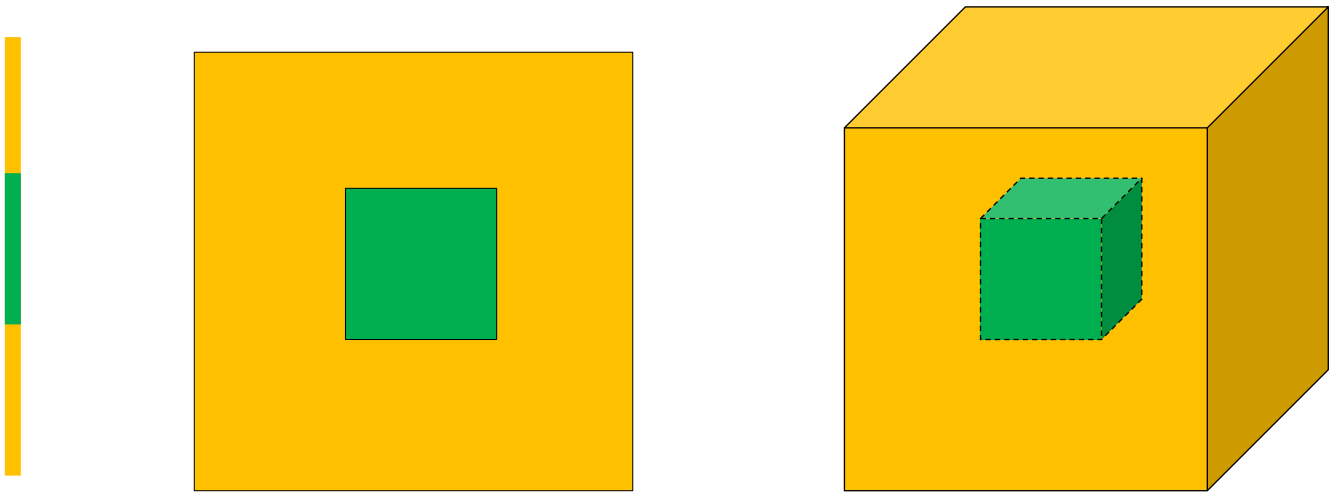


MCMC Proposals

- Uniform proposal: works very well for very simple models, but fails in complex ones (the acceptance of the chain drops)
- Multi-gaussian proposal from ProposalHelper: our current default, it works fine but for large number of nuisances (>20) the acceptance of the chain starts to drop, affecting the performance.
- We didn't test the proposal that starts from a fit covariance matrix; in past tests it worked like the Multi-gaussian but sometimes gave biased results

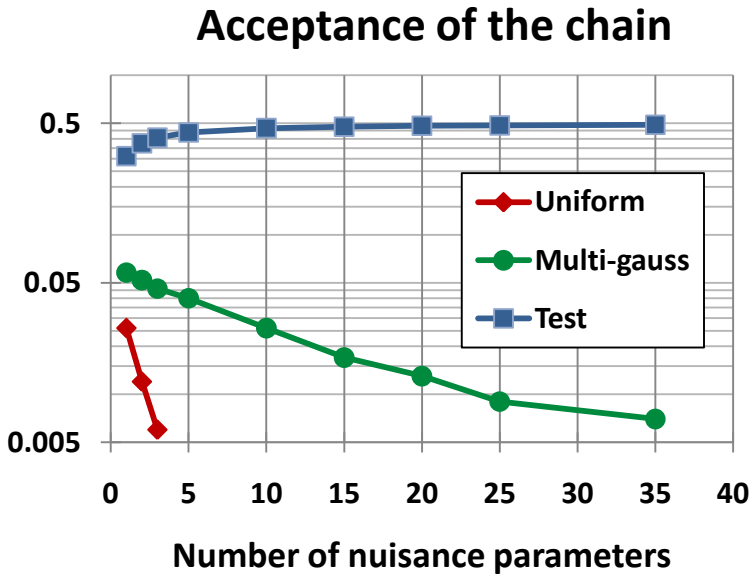
MCMC Proposals

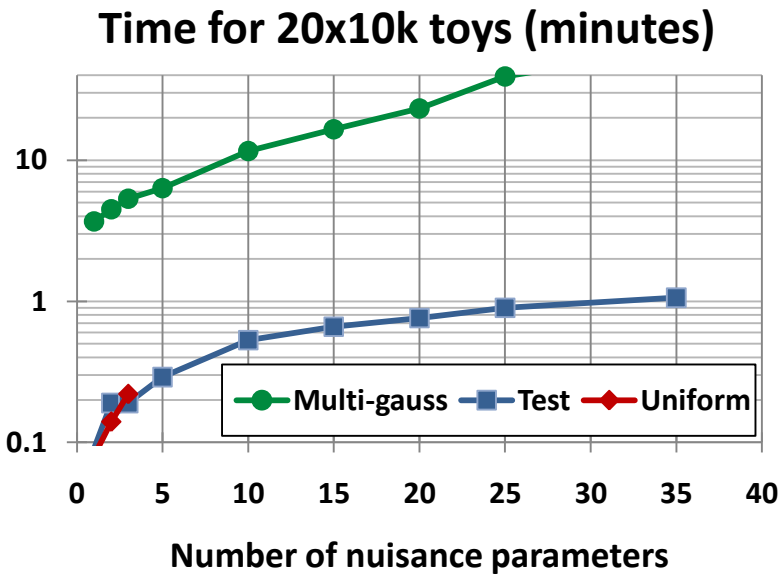
- The issue is that for large N most of the points of the $[\pm n\sigma]^N$ hypercube have vanishing probabilities even for small n (4-5).



MCMC Proposals

- Evaluated the performance of the various proposal with H->WW model but including only N nuisance parameters.





- Due to the higher correlation between points with the test proposal, the uncertainty increases with the number of nuisances (20x10k toys: 0.09%->1.4% for 1->35 nuisances)

Hybrid

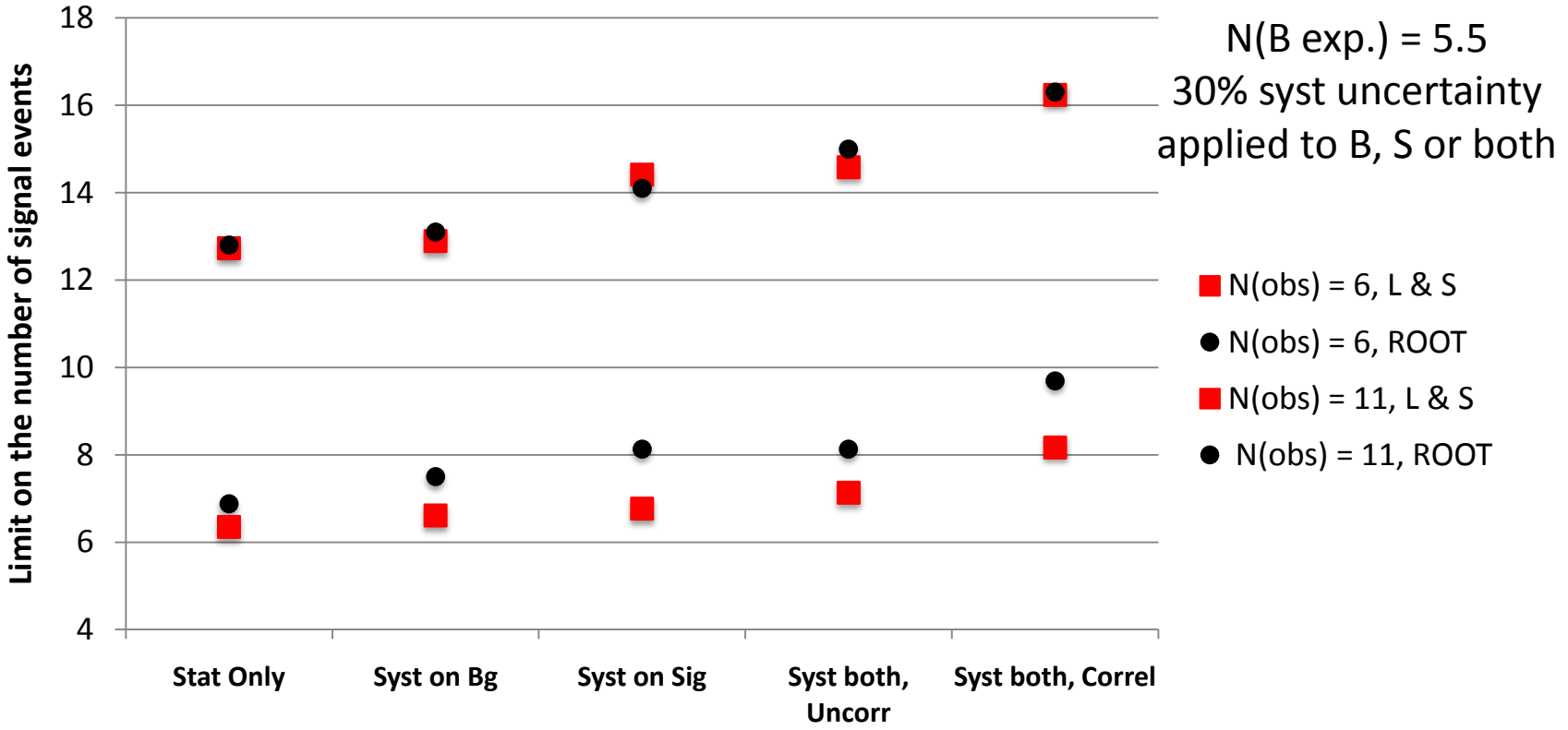
- **This code is very expensive in cpu and memory, limiting significantly the accuracy of what we can compute within "reasonable" bounds per job (e.g. 8hours & 8GB of RAM)**
- Multiple options were tested:
  - Two rules for limits: CL(s) vs CL(s+b)
  - Two test statistics: LEP (no profiling), Tevatron (profile nuisances). We had some problems trying to implement Atlas's one (profile all params.)

Hybrid for limits

- Comparison carried on only with limited accuracy (order 10% for complex models)
- **For the LEP test stat., results appear to be in agreement with expectations**, except perhaps when n(obs) is significantly below n(exp).
- **For the Tevatron test stat., found some issues**. Investigation ongoing to see if it's in our interface, but the poor performance make it painful (limits for simple benchmarks take several hours and sometimes exhaust the available memory)

Hybrid limits

Comparison for simple counting experiments with different n(obs) and syst. uncertainties.



Hybrid for significance

- Again, performance is a problem but in this case the task can be parallelized.
  - **Successfully computed a 3.5 sigma significance for a H->WW model for the LEP test statistics**
    Used 50 jobs of 10k toys, each taking 1.5h and 5GB.
    L&S on the same model does 1M toys in 2 minutes!
  - Other test statistics tested only in simpler jobs.
- Here the importance sampling should make a huge difference, but we didn't have time to test it yet.

Very first look at ROOT 5.28

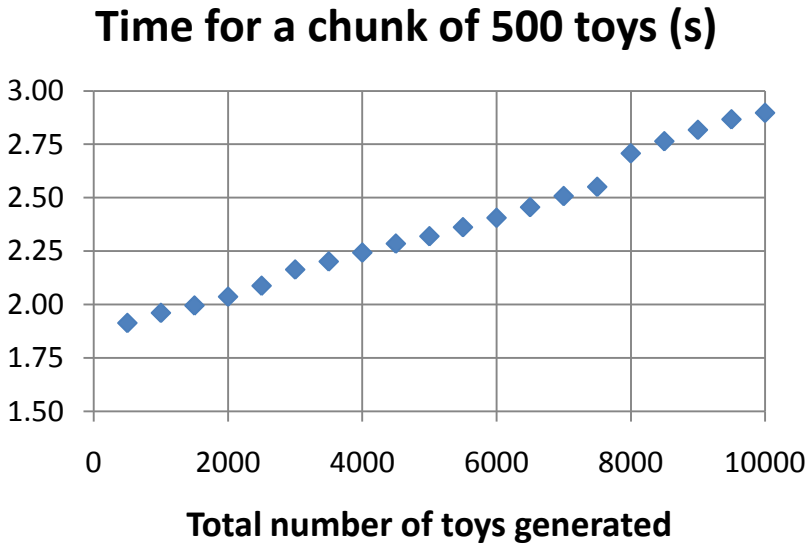
- **Profile-likelihood & Bayesian:** exactly identical results for the two releases (for MCMC the same seed was used)
- **HybridCalculatorOriginal:** same as 5.27, memory usage still increases with time.
- **New HybridCalculator**, but not using the new 5.28 features like importance sampling. Performance qualitatively similar to old one, and similar memory problem.

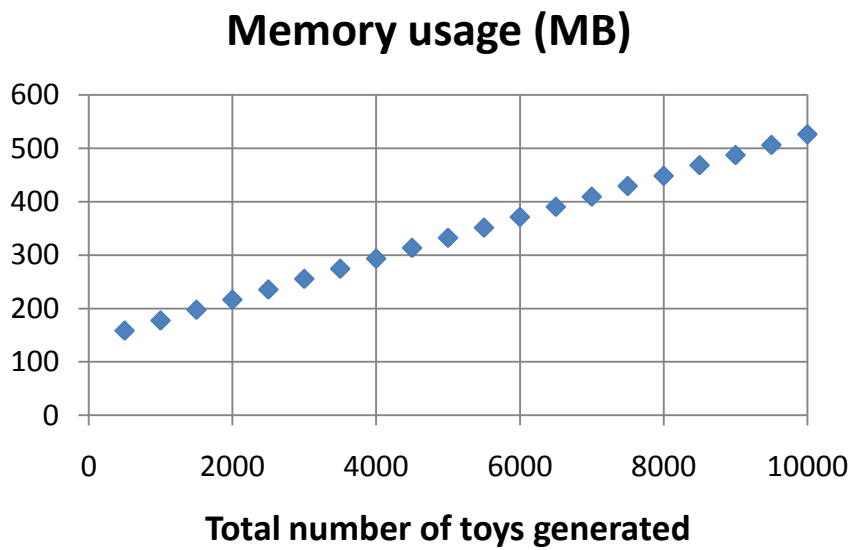
Summary

- Successful validation of the profile likelihood and the bayesian methods in ROOT 5.27.06.
- For MCMC, testing a new proposal function to improve the acceptance in complex models.
- For Hybrid, only partial validation done due to the performance issues. Will need to work further for the test statistics with profiling.
- First checks on ROOT 5.28 gave similar results as ROOT 5.27, and similar peformance issues, but we didn’t test yet the new features.

Getting Slower and Bigger

- HybridCalculatorOriginal toy generation becomes slower as the job goes on.





(simple counting experiment with a gaussian uncertainty on the background; LEP test stat.; **ROOT 5.28**)