

ColliderBit: A GAMBIT module for the calculation of high energy collider observables

ColliderBit author list

August 5, 2015

Abstract

We present a new code for the calculation of high energy collider observables, given a generic theory of beyond the Standard Model physics. Describe novel features of ColliderBit (parallelised MC generation, fast SUSY cross-section calculations, generic interface to BSM models, lots of LHC analyses). Mention link to GAMBIT framework, but emphasise that the package presents a standalone solution to the problem of applying LHC constraints to new physics theories.

1 Introduction

Despite decades of collider searches for physics beyond the Standard Model (BSM), it remains the case that we lack an unambiguous discovery of such physics. The many null results from the Large Hadron Collider (LHC) and other experiments allow us to constrain, to various degrees, the parameter spaces of any extension to the SM, including “bottom-up” effective theories of dark matter, supersymmetric theories, theories with extra space dimensions and composite Higgs or Technicolour models. Since a generic theory of BSM physics will have observable consequences in a large range of experiments, it is particularly important to combine collider exclusions with other experiments in a statistically rigorous way if one is to draw sound conclusions on the viability of a candidate theory.

It remains difficult to apply collider results to a generic theory of BSM physics. Searches for new particles at the LHC are typically presented either in particular planes of a GUT scale physics hypothesis (e.g the constrained minimal supersymmetric model), or in simplified models that only strictly apply to a very small volume of the total allowed space of particle masses and branching ratios. The computational expense of simulating signal processes for hundreds of thousands of points in a candidate model prevents an extended treatment. In addition, some LEP results remain state of the art,

and are not always rigorously applied in the literature. Finally, the discovery and subsequent observations of a SM-like Higgs boson by the ATLAS and CMS experiments in 2012 provides tight constraints on variations in the Higgs branching ratios that must be included in any thorough exploration of a BSM physics model [1, 2].

Partial solutions to each of these issues exist, but there is as yet no comprehensive tool that tackles all of them. The package **Smodels** applies constraints to supersymmetric models based on a combination of simplified model results [3]. **Fastlim** provides similar functionality for SUSY models, but is extendable (in principle) to non-SUSY models through the use of user-supplied efficiency tables [4]. Both of these tools will provide limits that are much more conservative than a more rigorous calculation, due to the limitations of simplified models. **CheckMATE** provides a customised version of the **DELPHES** detector simulation, an event analysis framework and a list of ATLAS and CMS analyses which can be used to apply LHC limits to candidate BSM models provided that the user supplies production cross-sections and a sample of Monte Carlo events [5, 6, 7]. To the best of our knowledge, no general purpose tool exists to apply LEP BSM search limits, although many theorists have implemented their own local codes over the years. On the Higgs side, the community is well served by the packages **HiggsBounds** and **HiggsSignals** [8, 9, 10].

We present a **ColliderBit**, a new software code for the application of high energy collider constraints to BSM physics theories. The code is designed within the **GAMBIT** framework, and thus offers seamless integration with other packages that provide a statistical fitting framework, and also the ability to impose constraints from electroweak precision data, flavour physics and a large range of astrophysical observations. Furthermore, the code is modular in design, allowing the user to easily swap components, add new collider analyses or provide interfaces to standard particle physics tools. For LHC physics, we use a combination of parallelised Monte Carlo simulation and fast detector simulation to recast LHC limits without the approximations of the simplified model approach. The package is supplied with a selection of the most important ATLAS and CMS analyses (focussing initially on supersymmetric and dark matter searches), and contains interfaces to the **Pythia 8** MC event generator and **DELPHES** detector simulation, in addition to a customised detector simulation based on four-vector smearing that is found to give comparable results to **DELPHES** with a reduced CPU overhead. We supply custom routines for re-evaluating LEP limits on supersymmetric particle production, and include an interface to **HiggsBounds** for the calculation of Higgs observables.

This paper serves as both a description of the physics and design strategy of **ColliderBit**, and a user manual for the first code release. In Section 2, we

provide a quick start guide for users keen to compile and use the software out of the box. Section 3 describes the physics and implementation of the ColliderBit software. The ColliderBit user interface is outlined in Section 4. Finally, Section 5 contains a detailed description of two use cases; the application of collider constraints to a) the minimal supersymmetric standard model (MSSM), and b) a toy model which is not currently included in the ColliderBit code. The second of these examples demonstrates the flexibility of ColliderBit in tackling generic theories supplied by the user (and we note that we here rely heavily on existing interfaces for automatic matrix element generation).

2 Quick Start Guide

Stuff.

3 Physics and implementation

For each part of the calculation outlined above: describe the code that exists within GAMBIT and/or the ready-made interface to a standard HEP code. Show validation plots (with possible links to appendices)..

Assuming that ColliderBit is supplied a parameter point of a physics model (either standalone via an SLHA file, or from a full GAMBIT run), the basic output is a series of likelihood terms derived from the LEP and LHC BSM searches, and the LEP, Tevatron and LHC Higgs searches. The terms may then be combined to form a composite likelihood as required by the user. We here describe the strategy for calculating each individual term, along with the code implementation.

3.1 LHC likelihood calculation

3.1.1 Physics background

A parameter point of a specified BSM model (hereafter referred to as a “model”) can in principle be expected to show up in a variety of LHC BSM searches. Assuming that these are “cut and count” style analyses, the relevant data to model are the number of events that pass the kinematic cuts imposed by each analysis. If a model predicts that s signal events pass the cuts for a given signal region, with b expected SM background events and o

observed events, the likelihood of the model is given by the standard Poisson formula:

$$\mathcal{L} = \frac{e^{-(s+b)}(s+b)^o}{o!} \quad (1)$$

Note that this does not include the effects of systematic uncertainties in the signal and background yields, σ_s and σ_b respectively. Assuming a distribution for the errors, one may treat them as nuisance parameters and marginalise over them as detailed in Section 3.1.6.

LHC BSM search papers provide details of b and o for each signal region, along with the background uncertainty (and some estimate of the signal uncertainty for representative models). Calculating the likelihood for a given model thus requires an accurate estimate of s which is given by:

$$s = \sigma \times A \times \epsilon \times L \quad (2)$$

where σ is the production cross-section for processes for which the detector has an acceptance of A and an efficiency of ϵ . L is the integrated luminosity of data used in the search (typically 20.1 fb^{-1} for the searches considered here).

The only fool proof way to calculate s is to perform a cross-section calculation at the desired accuracy before evaluating the acceptance and efficiency via a Monte Carlo simulation of the LHC collisions followed by a simulation of the ATLAS and/or CMS detectors. One can then apply the analysis cuts for a given LHC search to the results of the detector simulation. An approach using efficiency look up tables and extrapolation from simplified models removes the need for simulation, but gives very conservative results as a consequence, leading to an underestimate of the LHC reach. Furthermore, the generation of look up tables must be repeated for every physics model of interest, making it hard to produce a generic code for the application of LHC constraints.

The overriding strategy of the `ColliderBit` LHC module is instead to make each step of the simulation chain faster using a combination of custom speed increases and parallel computing. The package thus performs a cross-section calculation, generates Monte Carlo events, performs an LHC detector simulation and then applies the analysis cuts for a range of LHC analyses using a custom event analysis framework. The user can then utilise the `GAMBIT` statistical routines to return LHC likelihoods. The basic processing chain is illustrated in Figure 1, and the code is designed so that the user can

freely choose which software performs each step of this process. Nevertheless, `ColliderBit` has a default chain implemented, and the first release contains the elements summarised in the following subsections.

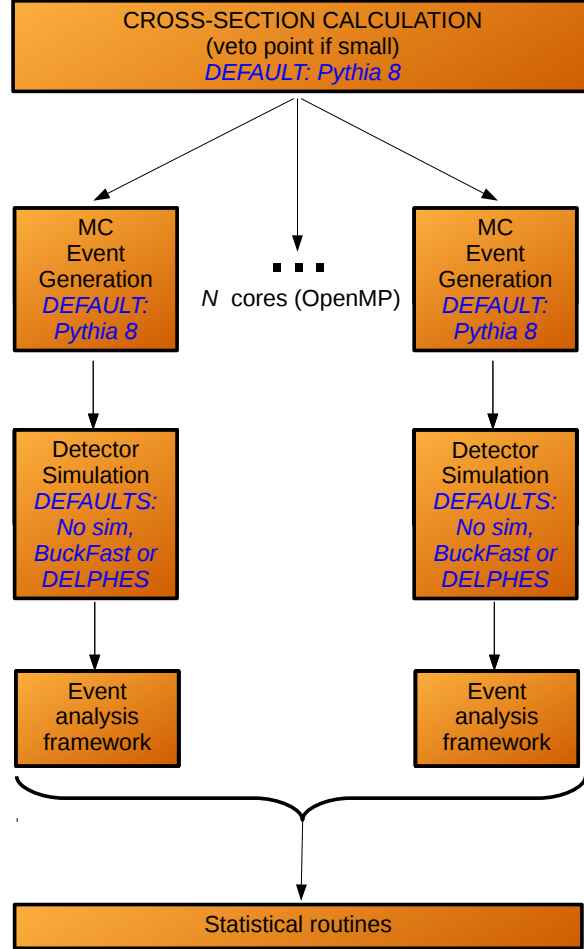


Figure 1: Schematic diagram of the `ColliderBit` processing chain for LHC likelihoods.

3.1.2 Cross-section calculations

We use the LO+LL cross-sections calculated numerically by the popular `Pythia 8` event generator. For many models, these are state of the art. For models where a NLO (or better) calculation is available (e.g. SUSY) our

results are conservative, but considerably quicker to evaluate than the full NLO results obtained using the standard `Prospino` tool [11].

For the first release, we should also provide an easy method of having a user-supplied cross-section, perhaps by adding a module function that reads the cross-section from a text file.

3.1.3 Monte Carlo event generation

We supply an interface to the `Pythia 8` event generator, alongside custom code that parallelises the main event loop of `Pythia` using `OpenMP` (for a similar approach, see [12]). This allows generation of $\approx 10,000$ SUSY events on an Intel Core i7 processor in less than 10s (**Should we note the necessary compile options here?**). In addition, in a GAMBIT-driven global fit, points are vetoed if the production cross-section is already too low to lead to observable consequences at the LHC, giving an implicit speed increase. Taken together, these routines make the running of a full Monte Carlo simulation in a global fit computationally tractable. The choice of the `Pythia` generator is an acceptable compromise between generality and ease of use for the first `ColliderBit` release. It is sufficient for many BSM models, and is easily extendable with matrix elements for new models via the existing `Madgraph 5` interface (for an example, see Section ADD). `Pythia` will prove insufficient, however, where NLO diagrams are significant (for example, in the accurate treatment of some effective field theories of dark matter in LHC monojet searches, where top quark loops become important [13]). These deficiencies can be fixed in the current release via a user-supplied interface to an appropriate Monte Carlo tool, and such interfaces will be supplied as standard in future `ColliderBit` releases.

Should we provide instructions for how to interface to a new MC tool for the first release? Perhaps we should make sure that `ColliderBit` doesn't fall over.

3.1.4 Detector simulation

`ColliderBit` is structured so that the detector simulation is run during the parallelised main event loop, thus giving an implicit speed increase of the simulation step. The user has several options for this step, and in all cases jets are reconstructed using the `Fastjet` package [14].

No detector simulation

The user can choose to run no detector simulation, in which case the truth level MC events are passed directly to the event analysis framework. Jets

may be defined directly at the parton level, or at the hadron level. The former is only really sufficient for analyses in which leptons are the main species of interest, in which case turning off hadronisation can lead to a large speed increase.

Delphes

We provide an interface to the DELPHES detector simulation (**Which version?**) [6, 7] which provides respected simulations of the ATLAS and CMS detectors. DELPHES includes a simulation of track propagation in the magnetic field of an LHC detector, along with a simulation of the electron and hadron calorimeters, and the muon chambers. The user can configure the parameters of the simulation using the normal DELPHES card, but it should be noted that b tagging and the ATLAS lepton ID selection efficiencies (“medium”, “tight”), etc, are applied within the event analysis framework in order to allow different analyses to use different working points.

BuckFast

For many purposes, a more approximate (and much faster) approach based on four vector smearing is sufficient, and we therefore also supply an internal ColliderBit detector simulation, **BuckFast**, that uses a mixture of the DELPHES momentum and energy resolution functions and new parameterisations of the ATLAS and CMS identification efficiencies, as follows: **(Need to add CMS functions for first release. Also should we update the DELPHES functions used for the initial efficiency and resolution?)**.

- **Electrons:** We apply, to the truth level electron four vectors, the DELPHES functions for electron tracking efficiency, electron energy resolution and electron reconstruction efficiency (in that order). In the analysis step, we apply parameterisations of the ATLAS medium and tight identification efficiencies (as appropriate), taken from [15].
- **Muons:** We apply, to the truth level muon four vectors, the DELPHES functions for the muon tracking efficiency, the muon momentum resolution and the muon reconstruction efficiency (in that order).
- **Taus:** Hadronic taus are identified at truth level, and selected using a flat 40% efficiency. The energy is smeared by a 3% jet energy resolution.
- **Jets:** Jets are reconstructed at hadron level using the anti- k_T algorithm of the **Fastjet** package **Need to add configurable jet radius**. The energy is smeared by a 3% jet energy resolution.
- **Missing energy:** **Andy? I can’t remember exactly how we do this.**

Andy to add a selection of DELPHES vs BuckFast validation plots for a representative signal process. Should also add some cutflow comparisons (MJW)?

3.1.5 Event analysis framework

- Describe event analysis framework
- Emphasise applicability to simulated and/or reconstructed events
- Describe the LHC analyses included in the module and include some validation plots showing some nicely reproduced ATLAS and CMS limits.
- Explain how to add new analyses

3.1.6 Statistics

- **Pat to describe the likelihoods implemented in ColliderBit.cpp (i.e. quote formulae for log normal and Gaussian error distributions).**

Having obtained a likelihood for each CMS and ATLAS signal region, we calculate the overall LHC likelihood by using the observed likelihood obtained using the signal region that gives the best expected limit. This is conservative, but is the appropriate treatment when one lacks sufficient information to handle correlated systematic uncertainties.

3.2 LEP likelihood calculation

Are to write short summary.

3.3 Higgs likelihood calculation

Chris Rogan to write short summary.

4 User interface

Abram to write the following:

- Describe interface to GAMBIT framework.

- Describe standalone interface (how to configure and run the package, what the defaults are).
- Describe the interface to the models that we include and how to change the input parameters of those models.
- Explain how to run code in single core or multicore form (i.e using the standard OpenMP commands).
- Describe parameters that can be varied in yaml file.

5 Examples

Give some tutorial-style examples for common use cases.

5.1 MSSM example

- Reproduction of an LHC CMSSM exclusion limit?
- How to scan over pMSSM parameters and return a likelihood.

5.2 Generic Pythia model example

Give an example of Madgraph matrix element code to Pythia 8, show how to get it working and how to scan over the parameters and return a likelihood. Might be ambitious for the first paper, but it wouldn't take too long and would be a superb feature for increasing the user base.

6 Conclusions

7 Acknowledgements

8 Appendix: Validation (optional)

References

- [1] Georges Aad et al. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC.

- Phys. Lett.*, B716:1–29, 2012.
- [2] Serguei Chatrchyan et al. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Phys. Lett.*, B716:30–61, 2012.
 - [3] Sabine Kraml, Suchita Kulkarni, Ursula Laa, Andre Lessa, Wolfgang Magerl, Doris Proschofsky-Spindler, and Wolfgang Waltenberger. SModelS: a tool for interpreting simplified-model results from the LHC and its application to supersymmetry. *Eur. Phys. J.*, C74:2868, 2014.
 - [4] Michele Papucci, Kazuki Sakurai, Andreas Weiler, and Lisa Zeune. Fastlim: a fast LHC limit calculator. *Eur. Phys. J.*, C74(11):3163, 2014.
 - [5] Manuel Drees, Herbi Dreiner, Daniel Schmeier, Jamie Tattersall, and Jong Soo Kim. CheckMATE: Confronting your Favourite New Physics Model with LHC Data. *Comput. Phys. Commun.*, 187:227–265, 2014.
 - [6] J. de Favereau et al. DELPHES 3, A modular framework for fast simulation of a generic collider experiment. *JHEP*, 1402:057, 2014.
 - [7] S. Ovin, X. Rouby, and V. Lemaitre. DELPHES, a framework for fast simulation of a generic collider experiment. 2009.
 - [8] Philip Bechtle, Oliver Brein, Sven Heinemeyer, Georg Weiglein, and Karina E. Williams. HiggsBounds: Confronting Arbitrary Higgs Sectors with Exclusion Bounds from LEP and the Tevatron. *Comput. Phys. Commun.*, 181:138–167, 2010.
 - [9] Philip Bechtle, Oliver Brein, Sven Heinemeyer, Georg Weiglein, and Karina E. Williams. HiggsBounds 2.0.0: Confronting Neutral and Charged Higgs Sector Predictions with Exclusion Bounds from LEP and the Tevatron. *Comput. Phys. Commun.*, 182:2605–2631, 2011.
 - [10] Philip Bechtle, Oliver Brein, Sven Heinemeyer, Oscar Stl, Tim Stefaniak, Georg Weiglein, and Karina E. Williams. HiggsBounds – 4: Improved Tests of Extended Higgs Sectors against Exclusion Bounds from LEP, the Tevatron and the LHC. *Eur. Phys. J.*, C74(3):2693, 2014.
 - [11] W. Beenakker, R. Hopker, and M. Spira. PROSPINO: A Program for the production of supersymmetric particles in next-to-leading order QCD. 1996.
 - [12] Christopher G. Lester, Michael Andrew Parker, and Martin J. White. Determining SUSY model parameters and masses at the LHC using cross-sections, kinematic edges and other observables. *JHEP*, 01:080, 2006.

- [13] Matthew R. Buckley, David Feld, and Dorival Goncalves. Scalar Simplified Models for Dark Matter. *Phys. Rev.*, D91(1):015017, 2015.
- [14] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. FastJet User Manual. *Eur. Phys. J.*, C72:1896, 2012.
- [15] Electron efficiency measurements with the ATLAS detector using the 2012 LHC proton-proton collision data. Technical Report ATLAS-CONF-2014-032, CERN, Geneva, Jun 2014.