LPSC140nn
HEPHY-PUB xxx/14

SModelS v1.0: a short user guide

Sabine Kraml[1], Suchita Kulkarni[1,2], Ursula Laa[1,2], Andre Lessa[3],
Veronika Magerl[2], Wolfgang Magerl[2], Doris Proschofsky-Spindler[2*],
Michael Traub[2], Wolfgang Waltenberger[2]

[1] *Laboratoire de Physique Subatomique et de Cosmologie, Université Grenoble-Alpes, CNRS/IN2P3, 53 Avenue des Martyrs, F-38026 Grenoble, France*

[2] *Institut für Hochenergiephysik, Österreichische Akademie der Wissenschaften, Nikolsdorfer Gasse 18, 1050 Wien, Austria*

[3] *Instituto de Física, Universidade de São Paulo, São Paulo - SP, Brazil*

Email: smodels-support@oeaw.ac.at

Abstract

SModelS is an automatised tool for interpreting simplified-model results from the LHC. Version 1.0 of the code is now publicly available. This note provides a quick user guide for installing and running SModelS v1.0.

*Present address: ....

1 Introduction

SModelS [1] is an automatic tool for interpreting simplified-model results from the LHC. It is based on a general procedure to decompose Beyond the Standard Model (BSM) collider signatures presenting a Z_2 symmetry into Simplified Model Spectrum (SMS) topologies. Our method provides a way to cast BSM predictions for the LHC in a model independent framework, which can be directly confronted with the relevant experimental constraints. Our concrete implementation currently focusses on supersymmetry searches with missing transverse energy (MET), for which a large variety of SMS results from ATLAS and CMS are available. The main ingredients of SModelS are

- the decomposition of the BSM spectrum into SMS topologies,
- the database of experimental SMS results,
- the interface between decomposition and results database.

The working principle is illustrated schematically in Fig. 1.

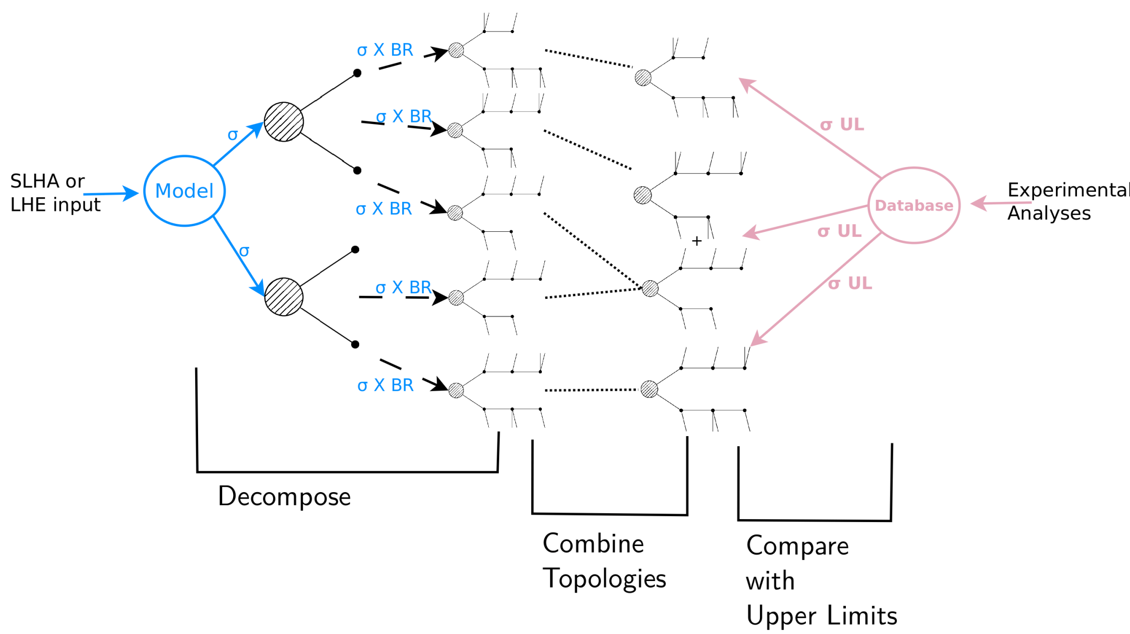

Figure 1: Schematic view of the working principle of `SModelS`.

SModelS v1.0 was publicly released on **[FIXME: date]** . This note provides a short user guide for the installation and for running SModelS v1.0. More details can be found in the primary physics publication [1] and in the documentation shipped with the code (also available online at [2]). In case of troubles using SModelS v1.0, we ask the user to kindly contact the authors at smodels-support@oeaw.ac.at.

2 Download and Installation

SModelS is a Python [?] library that requires Python version 2.6 or later (but not version 3). Internally, SModelS v1.0 uses Pythia 6.4.27 [3], NLL-fast versions 1.2 and 2.1 [4] (see also [5–11]) and a modified version of PySLHA [12]. It has been tested on Ubuntu 14.X, Scientific Linux 6, Scientific Linux CERN 6, as well as on Mac Os X 10.9 and 10.10.

The SModelS v1.0 package can be downloaded from [2]. Unpacking the tar ball with

```
tar -zxfv smodles-v1.0.tgz
```

creates a directory `smodels-v1.0` where the code (subdirectory `smodels`) and the results database (subdirectory `smodels-database`) are located.

For installation, SModelS makes use of Python's setuptools. On most machines (apart from SL6/SLC6)

```
python setup.py install
```

inside the `smodels-v1.0` directory should install the entire project, resolving automatically the external Python dependencies and compiling the internal Pythia 6 and NLL-fast versions using `gfortran`. Note that depending on the path specified in `setup.py` you may need to do the installation as superuser.

More details on the installation procedure, external dependencies and in particular instructions for installation on SL6 and SLC6 can be found in the INSTALL file in `smodels-v1.0`.

3 Citation

When you use results obtained with SModelS v1.0 in a publication, please cite this document as well as the original SModelS paper [1], Pythia 6.4 [3], NLL-fast [5–11] and PySLHA [12]. For convenience, these citations are provided in bibtex format in the SModelS v1.0 distribution.

4 Using SModelS

4.1 Basic input

SModelS v1.0 can be used with two forms of inputs

- SLHA (SUSY Les Houches Accord) [13] files containing masses, branching ratios and cross-sections, or
- LHE (Les Houches Event) [14] files containing parton-level events.

The SLHA format is usually more compact and best suited for supersymmetric models. On the other hand, a LHE file can always be generated for any BSM model (through the use of your favorite Monte Carlo generator). In this case, however, the precision of the results is limited to the MC statistics used to generate the file.

In the case of **SLHA input**, the production cross sections for the BSM states also have to be included as SLHA blocks, according to the SLHA cross-section format []. For the MSSM and some of its extensions, the cross sections can be conveniently calculated and added to the SLHA input file by means of the internal SModelS' `xseccomputer` described in the tools section of the online manual.

In the case of **LHE input**, the total production cross-section as well as the center-of-mass energy should be listed in the <init></init> block, according to the standard LHE format. Moreover, all the Z_2-even BSM particles (*e.g.*, additional Higgs states) should be set as stable, since in SModelS v1.0 they are effectively considered as final states.

Besides information about the masses and branching ratios, the user must also **define** which of the particles are Z_2**-odd states** and which are Z_2-even. This is done in the `particles.py` file, where some default values (for SM and MSSM particles) are already loaded. Finally, if the user wants to check the SLHA input file for possible issues using SModelS' SLHAchecker, it is also necessary to define the BSM particle quantum numbers in `particles.py`.

4.2 runSModelS.py

For the first-time user, SModelS ships with a command-line tool, `runSModelS.py`, which covers several different applications of the SModelS functionalities. These functionalities include detailed checks of input SLHA or LHE files, running the SMS decomposition, evaluating the theory predictions for the input model and comparing them to the experimental limits available in the database, determining the most important missing topologies and printing a summary text file.

The usage is

```
runSModelS.py -f INPUTFILE [-p PARAMETERFILE] [-o OUTPUTFILE]
```

Some comments are in order.

- The INPUTFILE can be an SLHA or LHE file as explained in Section 4.1.
- The PARAMETERFILE controls the basic options and parameters used by `runSModelS.py`. An example including all available parameters together with a short description, is provided as `parameters.ini`. If no parameter file is specified, the default parameters stored in `parameters_default.ini` are used. Note that the input type (SLHA or LHE) needs to be properly specified in the parameter file.
- If no OUTPUTFILE is specified, the file output will be printed to `summary.txt`.

4.3 Default output

...

...

...

...

...

4.4 Using SModelS v1.0 as Python library

Although `runSModelS.py` provides the main SModelS features with a command line interface, users familiar with Python and the SModelS language may prefer write their own main program, using SModelS v1.0 as a Python library. A simple example code for this purpose is provided as `Example.py` in the SModelS v1.0 distribution.

5 List of analyses included in the SModelS v1.0 release

The SModelS v1.0 database comprises a large number of SMS results from ATLAS and CMS for $\sqrt{s} = 8$ TeV. Most of them are for full luminosity. (The 7 TeV results will be included in a later release together with more 8 TeV results). Concretely, in the default run mode, the following results are considered:

ID	short description	L [fb^{-1}]	Ref.	Tx names
ATLAS-SUSY-2013-05	0 leptons + 2 b-jets + Etmiss	20.1	[]	T2bb

The **Tx** names are explained in the SMS dictionary on http://smodels.hephy.at/wiki/SmsDictionary.

The database contains moreover the entries of several preliminary results (ATLAS CONF notes or CMS PAS) which were superseded by a publication. These results are not listed here, and are not used when running `runSModelS.py`. They can, however, be activated by

```
smsAnalysisFactory.load(useSuperseded=True)
```

when using SModelS v1.0 as a Python library. For more details, see "How to load the database" in the Examples section of the manual.

Last but not least we note that for topologies with more than one step in the decay chain, *e.g.* charginos decaying through intermediate leptons, or stops decaying into bottom plus chargino followed by the chargino decay into the lightest neutralino, we need several (more than one) mass planes in order to interpolate between them. *Whenever only one mass plane is provided, the result is not useful for our purpose and thus not included in the database.*

6 Conclusions

Acknowledgements

We thank the ATLAS and CMS SUSY groups for helpful discussions on their results, and in particular for providing (most of) the SMS cross section upper limits used here in digital format.

This work was supported in part by the French ANR project DMAstroLHC. U.L. is supported by the “Investissements d’avenir, Labex ENIGMASS”. V.M. is grateful for financial support by the FEMtech initiative of the BMVIT of Austria. A.L. gratefully acknowledges the hospitality of LPSC Grenoble and of HEPHY Vienna.

References

[1] S. Kraml, S. Kulkarni, U. Laa, A. Lessa, W. Magerl, *et al.*, "SModelS: a tool for interpreting simplified-model results from the LHC and its application to supersymmetry," *Eur.Phys.J.* **C74** (2014) 2868, 1312.4175.

[2] http://smodels.hephy.at.

[3] T. Sjostrand, S. Mrenna, and P. Z. Skands, "PYTHIA 6.4 Physics and Manual," *JHEP* **0605** (2006) 026, hep-ph/0603175.

[4] http://pauli.uni-muenster.de/ãkule_01/nllwiki/index.php/NLL-fast.

[5] W. Beenakker, R. Hopker, M. Spira, and P. Zerwas, "Squark and gluino production at hadron colliders," *Nucl.Phys.* **B492** (1997) 51–103, hep-ph/9610490.

[6] W. Beenakker, M. Kramer, T. Plehn, M. Spira, and P. Zerwas, "Stop production at hadron colliders," *Nucl.Phys.* **B515** (1998) 3–14, hep-ph/9710451.

[7] A. Kulesza and L. Motyka, "Threshold resummation for squark-antisquark and gluino-pair production at the LHC," *Phys.Rev.Lett.* **102** (2009) 111802, 0807.2405.

[8] A. Kulesza and L. Motyka, "Soft gluon resummation for the production of gluino-gluino and squark-antisquark pairs at the LHC," *Phys.Rev.* **D80** (2009) 095004, 0905.4749.

[9] W. Beenakker, S. Brensing, M. Kramer, A. Kulesza, E. Laenen, *et al.*, "Soft-gluon resummation for squark and gluino hadroproduction," *JHEP* **0912** (2009) 041, 0909.4418.

[10] W. Beenakker, S. Brensing, M. Kramer, A. Kulesza, E. Laenen, *et al.*, "Supersymmetric top and bottom squark production at hadron colliders," *JHEP* **1008** (2010) 098, 1006.4771.

[11] W. Beenakker, S. Brensing, M. Kramer, A. Kulesza, E. Laenen, *et al.*, "Squark and Gluino Hadroproduction," *Int.J.Mod.Phys.* **A26** (2011) 2637–2664, 1105.1110.

[12] A. Buckley, "PySLHA: a Pythonic interface to SUSY Les Houches Accord data," 1305.4194.

[13] P. Z. Skands, B. Allanach, H. Baer, C. Balazs, G. Belanger, *et al.*, "SUSY Les Houches accord: Interfacing SUSY spectrum calculators, decay packages, and event generators," *JHEP* **0407** (2004) 036, hep-ph/0311123.

[14] J. Alwall, A. Ballestrero, P. Bartalini, S. Belov, E. Boos, *et al.*, "A Standard format for Les Houches event files," *Comput.Phys.Commun.* **176** (2007) 300–304, hep-ph/0609017.