

SModelS v1.2: long-lived particles, combination of signal regions, and other novelties

Federico Ambrogia^a, Juhi Dutta^b, Jan Heisig^c, Sabine Kraml^d, Suchita Kulkarni^e, Ursula Laa^{f,g}, Andre Lessa^h, Philipp Neuhuber^e, Humberto Reyes-González^d, Wolfgang Waltenberger^e, Matthias Wolf^e

^a*University of Vienna, Faculty of Physics, Boltzmanngasse 5, A-1090 Wien, Austria*

^b*Regional Centre for Accelerator-based Particle Physics, Harish-Chandra Research Institute, HBNI, Chhatnag Road, Jhusi, Allahabad-211019, India*

^c*Centre for Cosmology, Particle Physics and Phenomenology (CP3), Université catholique de Louvain, Chemin du Cyclotron 2, B-1348 Louvain-la-Neuve, Belgium*

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

^e*Institut für Hochenergiephysik, Österreichische Akademie der Wissenschaften, Nikolsdorfer Gasse 18, 1050 Wien, Austria*

^f*School of Physics and Astronomy, Monash University, Clayton, VIC 3800, Australia*

^g*Department of Econometrics and Business Statistics, Monash University, Clayton, VIC 3800, Australia*

^h*Centro de Ciências Naturais e Humanas, Universidade Federal do ABC, Santo André, 09210-580 SP, Brazil*

Abstract

SModelS is an automatised tool enabling the fast interpretation of simplified model results from the LHC within any model of new physics respecting a $\mathbb{Z}_2$ symmetry. With the version 1.2 we announce several new features. First, previous versions were restricted to missing energy signatures and assumed prompt decays within each decay chain. SModelSv1.2 considers the lifetime of each $\mathbb{Z}_2$ -odd particle and appropriately takes into account missing energy, heavy stable charge particle and R-hadron signatures. Second, the current version allows for a combination of signal regions in efficiency map results whenever a covariance matrix is available from the experiment. This is an important step towards fully exploiting the constraining power of efficiency map results. Several other improvements increase the user-friendliness, such as the use of wildcards in the selection of experimental results, and a faster database which can be given as a URL. Finally, `smodelsTools` provides an interactive plots maker to conveniently visualize the results of a model scan.

Keywords: LHC; supersymmetry; simplified models; physics beyond the standard model; reinterpretation

NEW VERSION PROGRAM SUMMARY

Program Title: SModelS

Licensing provisions: GPLv3

Programming language: Python3

Journal reference of previous version: Comput. Phys. Commun. **227** (2018) 72

Does the new version supersede the previous version?: Yes

Reasons for the new version: Addition of new features

Summary of revisions:

The most important new features in v1.2 are the combination of signal regions in efficiency map results whenever a covariance matrix is available from the experiment, and the implementation of heavy stable charge particle and R-hadron signatures. Moreover, the database of experimental results can now be given as a URL, and the pickling has been improved to make the database faster. Other improvements include that wildcards are allowed when selecting analyses, datasets or topologies, and that the path to the model file, formerly required to be `smodels/sparticles.py`, can be specified in the parameters card. For the convenience of the user, we also provide a tool to make interactive plots to visualize results of a model scan. Finally, the whole code now also runs with Python3, which has become the recommended default, and it can now be installed in its source directory.

Nature of problem:

The results for searches for new physics beyond the Standard Model (BSM) at the Large Hadron Collider are often communicated by the experimental collaborations in terms of constraints on so-called simplified models spectra (SMS). Understanding how SMS constraints impact a realistic new physics model, where possibly a multitude of production channels and decay modes are relevant, is a non-trivial task.

Solution method:

We exploit the notion of simplified models to constrain full models by “decomposing” them into their SMS components. A database of SMS results obtained from the official results of the ATLAS and CMS collaborations, but in part also from ‘recasting’ the experimental analyses, can be matched against the decomposed model, resulting in a statement to what extent the model at hand is in agreement or contradiction with the experimental results. Further useful information on, e.g., the coverage of the model’s signatures is also provided.

Additional comments including Restrictions and Unusual features:

At present, only models with a $\mathbb{Z}_2$ -like symmetry can be tested. Each SMS is defined purely by the vertex structure and the final-state particles; initial and intermediate BSM particles are described only by their masses, production cross sections, branching ratios and total width. Possible differences in signal selection efficiencies arising, e.g., from different production mechanisms or from the spin of the BSM particles, are ignored in this approach. Since only part of the full model can be constrained by SMS results, SModelS will always remain more conservative (though orders of magnitude faster) than “full recasting” approaches.

- [1] F. Ambrogio *et al.*, “SModelS v1.1 user manual: Improving simplified model constraints with efficiency maps,” *Comput. Phys. Commun.* **227** (2018) 72 [arXiv:1701.06586 [hep-ph]].

1. Introduction

S MODELS is an automatic, public 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 respecting a $\mathbb{Z}_2$ 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.

In Ref. [1], we introduced the use of efficiency maps¹ in SModelS. The advantage of efficiency maps (EMs) over cross section upper limit (UL) maps is that they allow for combining contributions from different simplified-model topologies to the same signal region (SR) of a given experimental analysis. This can significantly enhance the coverage and constraining power of the simplified model results. However, a drawback compared to UL-type results so far was that only the most sensitive SR could be used in EM-type results.

The next step further is to statistically combine the contributions to the different SRs within an analysis. This is possible if a) the SRs are exclusive, i.e. non-overlapping, and b) a likelihood model describing the background correlations is available. To this end, the CMS collaboration has recently started to provide covariance matrices within the simplified likelihood framework [2] for their supersymmetry (SUSY) results. The support for these covariance matrices and hence the combination of signal regions is one of the main new features presented here.

The other major extension presented in this paper concerns the kind of topologies which can be treated by S MODELS: so far, in previous versions, only BSM searches in final states with missing transverse momentum (MET) could be taken into account. Furthermore, all BSM states were required to have prompt decays or be stable (at collider scales). However, well-motivated BSM theories also often feature meta-stable states or long-lived particles (LLPs) which are electrically or color charged. S MODELS v1.2 now supports two important classes of LLP signatures: heavy stable charged particles (HSCPs) and R-hadron signatures. To this end, the decomposition procedure was extended to include a **final state** property, which corresponds to the distinct types of signatures: MET, HSCP or R-hadron. The inclusion of LLP signatures in S MODELS and its physics impact were first

¹By efficiency maps we mean maps of acceptance $\times$ efficiency ($\text{eff} = A \times \epsilon$).

presented in [3]; here give some more technical details of the implementation and how to use it.

Other novelties in this new version include:

- the database of experimental results can now be given as a URL;
- new experimental results were added to the database;
- the pickling has been improved to make the database faster;
- wildcards are allowed in the selection of analyses, topologies or datasets;
- the path to the model file, formerly required to be `smodels/particles.py`, can be specified in the parameters file (`parameters.ini`);
- more detailed information is available in the default output formats;
- a tool to make interactive plots was added to `smodelsTools`;
- the default python version changed from Python 2 to Python 3;
- a makefile is provided to install SMODELS in the source folder, as an alternative to the previous installation in the default system or user directory.

The concrete version we are presenting in the following is SMODELS v1.2.2. In section 2 we first explain the new features accessible via the parameters file before discussing in more detail the combination of signal regions in section 3. The implementation of LLP signatures is presented in section 4. The module in `smodelTools` to make interactive plots is presented in section 5. Further updates of the code and the database are documented in section 6, and the new installation procedure in section 7. We conclude in section 8.

2. New features accessible via the parameters file

The basic options and parameters used by `runSMODELS.py` are defined in the parameter file. An example parameter file, including all available parameters together with a short description, is given in `parameters.ini` in the SMODELS distribution. If no parameter file is specified, the default parameters stored in `smodels/etc/parameters_default.ini` are used. Below we describe the parts which are new with respect to SMODELS v1.1 [1].

Section “options”: main options for turning SModelS features on or off.

- **combineSRs** (True/False): Set True to combine signal regions when a covariance matrix is available. Note that this might take a few seconds per point. False uses only the most sensitive (best expected) SR.

The combination of SRs is described in more detail in section 3 below.

Section “particles”: allows to define the BSM particle content.

- **model** (path): path to the Python file which describes the $\mathbb{Z}_2$ -even and $\mathbb{Z}_2$ -odd particle content of the BSM model to be tested.² This can be given either in Unix file notation, e.g.

`model=share/models/mymodel.py`

or as a Python module path, e.g.

`model=share.models.mymodel`

If `model=mymodel`, the file `mymodel.py` is searched for in the current working directory as well as in `smodels/share/models/`. The latter directory contains a few sample model files. If no model is specified, the standard MSSM particle content is taken as the default.

Note that from v1.2 onwards the quantum numbers of each BSM particle must also be specified in order to classify long lived particles as missing energy, HSCP or R-hadron final states. The mapping of quantum numbers to final states is defined in `smodels/theory/particleNames.py`.

²Formerly the particle content had to be specified in the hard-coded file path `smodels/particles.py`.

Section “*database*”: allows choice of database version and/or selection of a subset of experimental results.

- **path:** the absolute or relative path to the database. The user can supply either the directory name of the database, or the path to the pickle file. From v1.1.3 onwards, it is also possible to provide the URL to the official database release, e.g.:

```
path = http://smodels.hephy.at/database/official122.
```

The `official122.pkl` file has about 400 MB; downloading it is often faster than parsing the text database oneself. Moreover, it has the advantage of avoiding any machine dependence in the pickling. The list of database releases is available at [4]. Note that separate URLs are provided for database versions which do not or do include the 8 TeV efficiency maps from Fastlim-1.0 [5]. Furthermore, some pickle files can only be used with the corresponding SModelS version.

Note: from v1.2.2 onwards, the shorthand notation `path=official` (or `path=official.fastlim`) points to the official database without (with) Fastlim of the same version as the SModelS code that is being used.

- **analyses** (list of results): set to *all* to use all available results. If a list of experimental analyses is given, only these will be used.

Wildcards (*, ?, [⟨list-of-ored-letters⟩]) are expanded in the same way a Unix shell does wildcard expansion for file names. For example, `ATLAS*` will use all ATLAS results, while `CMS-SUS-16*` will use only published CMS SUSY results with an ID from 2016.

To filter by centre-of-mass energy, use a suffix beginning with a colon, like `:13*TeV` (note that the asterisk in the suffix is a multiplication operator and not a wildcard). This can be combined with wildcards, e.g., `CMS*:13*TeV` will use only CMS Run 2 results.

- **txnames** (list of topologies): set to *all* to use all available simplified model topologies. The topologies are labeled according to the TxName convention [6]. If a list of TxNames is given, only the corresponding topologies will be considered. Wildcards (*, ?, [⟨list-of-ored-letters⟩]) can now also be used. As for analyses above, they are expanded in the same way the shell does wildcard expansion for file names. For example, `txnames = T[12]*bb*` picks all txnames beginning with T1 or

T2 and containing bb. As of the time of writing these were: T1bbbb, T1bbbt, T1bbqq, T1bbtt, T2bb, T2bbWW, T2bbWWoff.

- **dataselector** (list of datasets): set to *all* to use all available data sets. If dataselector = upperLimit (efficiencyMap), only UL-type results (EM-type results) will be used. Furthermore, if a list of signal regions (data sets) is given, only the experimental results containing these datasets will be used. Wildcards can be used in the same way as for analyses and txnames.

Section “*python/xml printers*”: options for printing the results.

- **addTxWeights** (True/False): set to True to show individual contribution from each topology to the total theory prediction for EM-type results.

3. Combination of signal regions

3.1. Method

If the experiment provides a covariance matrix together with efficiency maps for SMS topologies, the signal contributions in different signal regions can be combined [2, 7].³ This is implemented in SModelS v1.1.3 onwards following the (symmetric) simplified likelihood approach of [2], and easily extendible to the more general treatment described in [7] once experimental results are available in that format.

SModelS allows for a marginalisation as well as a profiling of the nuisances, with profiling being the default. As CPU performance is a concern in SModelS, we try to aggregate the official results, which can comprise a very large number of signal regions, to an acceptable number of aggregate regions. Here “acceptable” means as few aggregate regions as possible without significant loss in precision or constraining power. The CPU time required scales roughly linearly with the number of signal regions, so aggregating e.g. from 80 to 20 signal regions means gaining a factor of four in computing time.

³We stress here that, with or without covariances, EMs are needed *for all signal regions*. Since the best SR can change depending on the model being tested, a single map containing the efficiencies for the best SR in each bin cannot be used in a general way.

For the computation of the 95% confidence level from the likelihoods, a $CL_s(= CL_{sb}/CL_b)$ limit [8] is computed from the test statistic q_μ , as described in section 2.4, Eq. (14), in [9]. We then search for $CL_s = 0.95$ using Brent’s bracketing technique through the SciPy optimize library, see [10].

When using `runSMODELS.py`, the combination of signal regions is turned on or off with the parameter `combineSRs` described in the previous section. Per default, `combineSRs=False`, in which case only the result from the best expected signal region (best SR) is reported. If `combineSRs=True`, both the combined result and the result from the best SR are quoted. If the user writes his/her own python code, the combination of SRs is invoked by setting `combinedResults=True` in `theoryPredictionsFor()`. In the same instance, one can switch between profiling and marginalizing with `marginalize=True/False`, `False` being the default. An explicit example is discussed in the “How To’s” in the online manual [11].

3.2. Available results in the database

The CMS SUSY group is providing covariance matrices for most of their analyses, so far under the assumption of Gaussian errors. However, only two CMS analyses also provide simplified-model efficiency maps for each SR: CMS-SUS-16-050 [12] and CMS-PAS-SUS-16-052 [13].⁴ Since the former analysis has significantly non-Gaussian background uncertainties, as we have checked, the covariance matrix provided for it is not a good approximation [7]. For this reason, at present only the covariance matrix for CMS-PAS-SUS-16-052 is included in the SMODELS database.

Concretely, CMS-PAS-SUS-16-052 is a search for supersymmetry with compressed mass spectra in events with at least one soft lepton, moderate to high values of missing transverse momentum p_T^{miss} , and one or two hard jets, compatible with the emission of initial-state radiation. It targets scenarios of stop-pair production, $pp \rightarrow \tilde{t}_1 \tilde{t}_1^*$, where the mass difference to the $\tilde{\chi}_1^0$ is smaller than the mass of the W boson. The analysis has 44 SRs. The simplified model interpretations assume either four-body stop decays, $\tilde{t} \rightarrow b f \tilde{f}' \tilde{\chi}_1^0$ (T2bbWWoff) or decays via an intermediate chargino, $\tilde{t} \rightarrow b \tilde{\chi}_1^+$, $\tilde{\chi}_1^+ \rightarrow f \tilde{f}' \tilde{\chi}_1^0$ (T6bbWWoff). Efficiency maps for all 44 SRs are provided for both simplified models.

⁴Other CMS Run 2 SUSY analyses provide covariance matrices but no efficiency maps. ATLAS analyses typically give efficiencies for the best SR only; moreover they do not provide covariances.

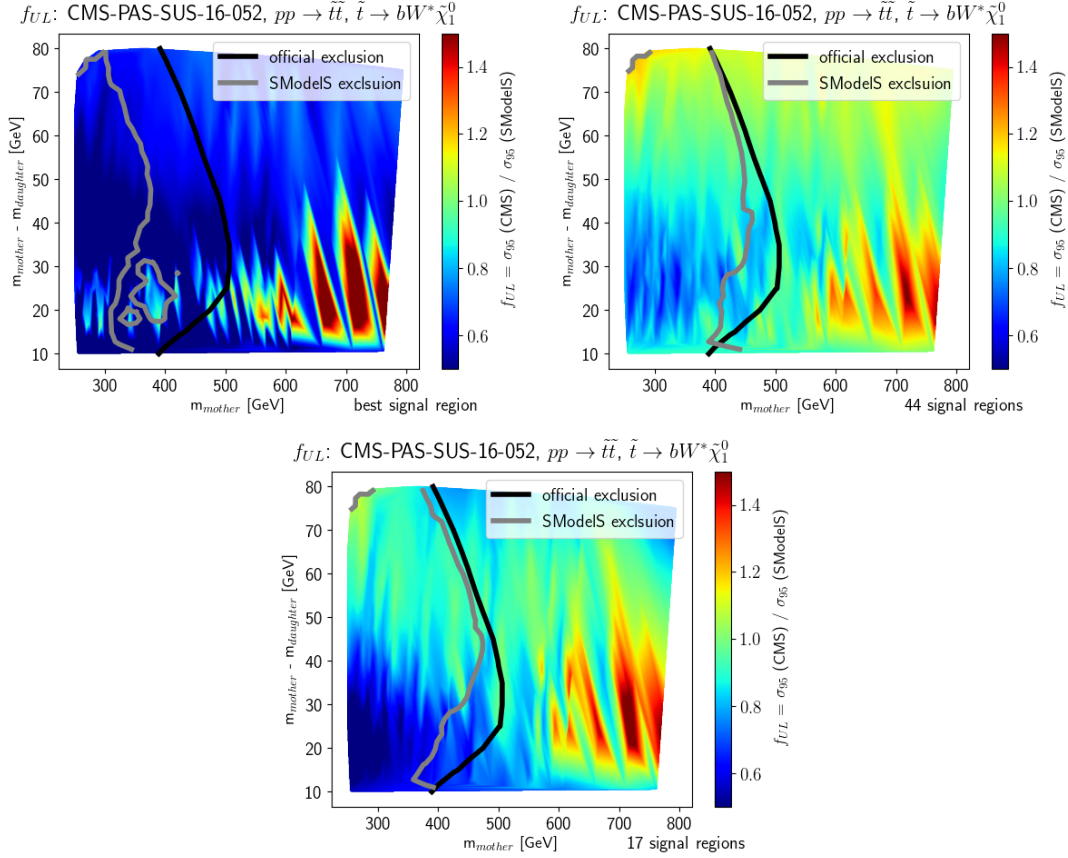


Figure 1: Comparison of exclusion curves from CMS-PAS-SUS-16-052 efficiency maps for the T2bbWWoff simplified model, using only the best signal region (top left), the combination of all 44 signal regions (top right) and the combination of 17 aggregate signal regions (bottom). The colour map shows the f_{UL} -value obtained from dividing the official limit by the one obtained with SModelS. The grey lines indicate the SModelS exclusion $r = 1$. For comparison, the black line shows the official CMS exclusion curve.

Figure 1 shows validation plots for the T2bbWWoff simplified model, comparing the 95% CL cross section upper limit derived by SModelS to the official UL from CMS across the SMS mass plane for three cases: using only the best SR (top left), combining all 44 SRs (top right), and aggregating to 17 SRs before combination (bottom). The x and y axes are $m_{\tilde{t}}$ and $m_{\tilde{t}} - m_{\tilde{\chi}_1^0} \equiv \Delta m$, respectively, and the colour code shows $f_{UL} \equiv \sigma_{95}^{ul}(\text{CMS}) / \sigma_{95}^{ul}(\text{SModelS})$. A value of $f_{UL} = 0.8$ means that our SR combination gives a 20% weaker limit than the official CMS result, a value of $f_{UL} = 1.2$ means that it is 20%

too aggressive. Also shown are the official exclusion line (black) and the exclusion line derived from SModelS (grey).

We see that using only the best SR leads to too weak a limit: the mean f_{UL} is about 0.6 and the mass limit is up to about 150 GeV too low—compared to the maximum reach of $m_{\tilde{t}} \approx 500$ GeV this is significant. In contrast, the combination of all 44 SR performs much better for reproducing the official CMS result. Aggregating the 44 original SRs to 17 speeds up the calculation by more than a factor 2 while still giving a result very close to the full combination. The same holds true for the T6bbWWoff topology. We therefore included the efficiency maps for 17 aggregated SRs as CMS-PAS-SUS-16-052-agg in the SModelS database. The covariance matrix is given in the `globalInfo.txt` file in the same folder. Combining 17 SRs takes less than 2 sec on an average 4-core Intel i5 desktop computer.

4. Implementation of heavy stable charge particle and R-hadron signatures

The implementation of HSCP and R-hadron signatures required several important changes in the SModelS code as detailed below. For the user this is primarily noticeable in the “missing topologies” output. The usage in particular of `runSModelS.py` remains the same as before.

4.1. Extension of decomposition procedure

Taking the BSM particle masses, quantum numbers, total decay widths, branching ratios (BRs) and total production cross-sections as input, SModelS performs a decomposition of the collider signature of any $\mathbb{Z}_2$ symmetric BSM model into a coherent sum of simplified-model topologies. The resulting topologies exhibit a two-branch structure emerging from the production of two $\mathbb{Z}_2$ -odd BSM states and their subsequent cascade decays. In the v1.0 and v1.1 releases, it was assumed that all BSM decays are prompt and that all cascade decays end in a stable neutral BSM particle leading to MET final states (otherwise, if `checkInput=True`, an error was issued). From v1.2 onwards this has been generalized in two ways.

First, instead of always assuming a MET final state, we allow each cascade decay to terminate in any BSM particle. Using the quantum numbers defined for the BSM particle (see section 2), the final state is then classified as a MET signature, an HSCP or an R-hadron. If it does not fall in any of these categories, an error message is displayed.

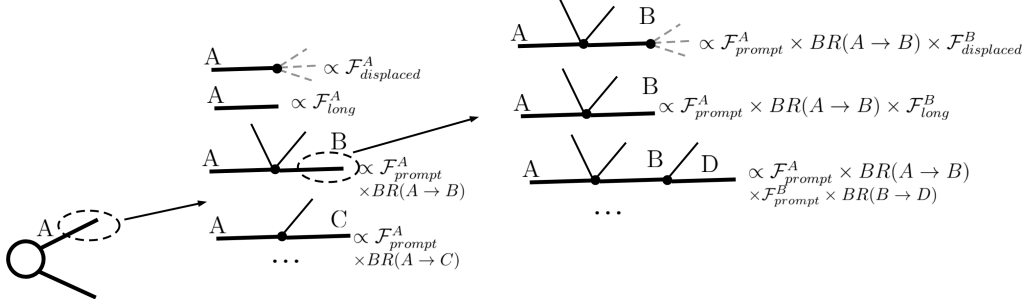


Figure 2: Decomposition into simplified model topologies and computation of weights in SMODELS v1.2.

Second, during the decomposition into simplified models, at each step in the cascade we compute the probabilities for the respective BSM particle to decay promptly ($\mathcal{F}_{\text{prompt}}$) and to decay outside the detector ($\mathcal{F}_{\text{long}}$) [14, 3]. The procedure is illustrated schematically in Fig. 2. The fraction of decays which take place inside the detector (labeled simply as “displaced decays”) is approximated as $\mathcal{F}_{\text{displaced}} = 1 - \mathcal{F}_{\text{long}} - \mathcal{F}_{\text{prompt}}$. Note that the final states appearing in the displaced vertex are not stored during decomposition, since SMODELS is currently not able constrain displaced decay signatures; this is left for future work. As a result, for the time being, all elements with displaced decays will be identified as missing topologies.

Concretely, $\mathcal{F}_{\text{long}}$ and $\mathcal{F}_{\text{prompt}}$ are computed from the respective particle proper lifetime, τ , using the approximation

$$\mathcal{F}_{\text{prompt}} = 1 - \exp\left(-\frac{1}{c\tau} \left\langle \frac{\ell_{\text{inner}}}{\gamma\beta} \right\rangle_{\text{eff}}\right) \quad (1)$$

and

$$\mathcal{F}_{\text{long}} = \exp\left(-\frac{1}{c\tau} \left\langle \frac{\ell_{\text{outer}}}{\gamma\beta} \right\rangle_{\text{eff}}\right). \quad (2)$$

We choose $\langle \ell_{\text{inner}}/\gamma\beta \rangle_{\text{eff}} = 1 \text{ mm}$ and $\langle \ell_{\text{outer}}/\gamma\beta \rangle_{\text{eff}} = 7 \text{ m}$,⁵ which provides a good approximation to the result of a full simulation as shown in Appendix B of [3].

After decomposition the respective weight $\tilde{\sigma}$ of each simplified-model

⁵As we currently include CMS results only, ℓ_{outer} corresponds to the CMS detector size. For the inclusion of ATLAS results $\langle \ell_{\text{outer}}/\gamma\beta \rangle_{\text{eff}}$ would have to be adjusted accordingly.

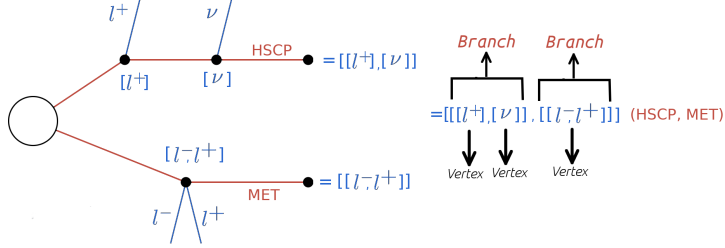


Figure 3: Extension of the SModelS bracket notation to include the $\mathbb{Z}_2$ -odd final state description, in this example (HSCP, MET).

topology is hence given by

$$\tilde{\sigma} = \sigma_{\text{prod}} \left(\prod_i \text{BR}_i \times \mathcal{F}_{\text{prompt}}^i \right) \mathcal{F}_{\text{long/displaced}}^X \mathcal{F}_{\text{long/displaced}}^Y, \quad (3)$$

where σ_{prod} is the production cross section of the mother particles and X, Y are the $\mathbb{Z}_2$ -odd final states of the two cascades. The index i runs over all intermediate $\mathbb{Z}_2$ -odd particles. For each chain, the last probability factor is given by $\mathcal{F}_{\text{long}}$ for decays which take place outside the detector, or by $\mathcal{F}_{\text{displaced}}$ for displaced decays.

4.2. Extension of SModelS bracket notation

As explained in detail in [1], inside SModelS the structure and final states of elements are represented in textual form using a nested brackets notation. In order to fully specify all the information of a given SMS topology (an *element*), we must also include the list of masses for the $\mathbb{Z}_2$ -odd states, the list of $\mathbb{Z}_2$ -odd final states and the element weight. The masses for the $\mathbb{Z}_2$ -odd BSM states are represented by a mass array for each branch. An example is given in Fig. 3.

The quantum numbers of this BSM final state are essential for defining which type of signature this element represents. In an element the $\mathbb{Z}_2$ -odd final state quantum numbers are mapped to final state signatures, as defined in the `particleNames` module. Currently the defined types of final states are: MET, HSCP, RHadronG, RHadronQ, the latter two being color-octet and color-singlet R-hadrons, respectively. New final state types can easily be added in this module.

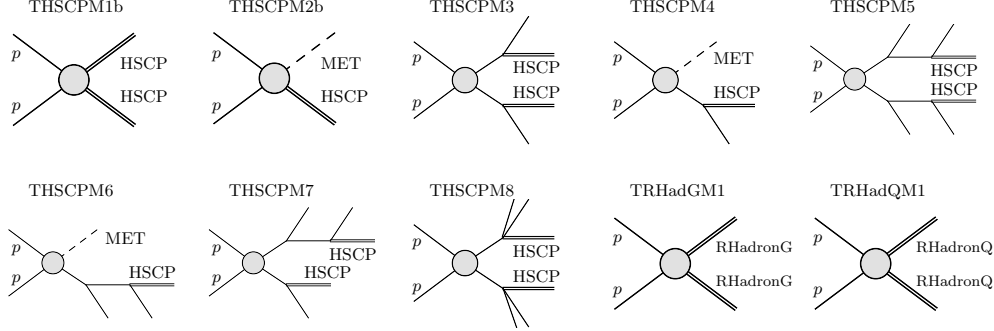


Figure 4: Simplified model topologies containing HSCP and R-hadron final states included in the SMODELS v1.2.2 database. A double line represents an HSCP or R-hadron, while a dashed line represents an arbitrary decay chain terminating in a neutral LLP (providing a MET signature). The labels of the topologies correspond to the naming convention used in the database.

4.3. HSCP and R-hadron results in the database

The v1.2.2 database includes results from the CMS searches for lepton-like HSCPs (carrying one unit of elementary charge), color-octet (gluino-like) R-hadrons and color-triplet (squark-like) R-hadrons at 8 TeV [15] and 13 TeV [16] centre-of-mass energies. The relevant topologies and their shorthand notation (“txnames”) in the SMODELS database are summarized in Fig. 4.

We computed efficiency maps for the eight HSCP simplified-model topologies shown in Fig. 4. To this end, for the 8 TeV analysis we utilized the recasting provided in Ref. [17]. For the 13 TeV analysis we performed a dedicated recasting (see Appendix A of [3] for details). These efficiency maps are included in the SMODELS v1.2.2 database.

For the R-hadron searches we consider the direct production topologies (TRHadGM1, TRHadQM1) only. As R-hadrons are strongly produced, their production via cascade decays is assumed to be less relevant and hence not considered here. We include the respective cross-section upper limits from Refs. [15, 16] considering the cloud hadronization model (assuming a 50% probability for gluino-gluon bound state formation).

5. Interactive plots maker in `smodelsTools`

For a simple and quick visualisation of results from a scan over input files, we now provide an “interactive plots maker” as part of `smodelsTools`. This tool allows to easily produce interactive plots which relate the SModelS output (*in python output format*) with information on the users model stored in the SLHA files. It gives 2d plots in the parameter space defined by the user, with additional user-defined information appearing in hover boxes. The output is in html format for viewing in a web browser. We stress that the aim is not to make publication-ready plots but simply to facilitate the user’s analysis of, e.g., the properties of points in a scan. The usage of the interactive plots tool is:

```
smodelsTools.py interactive-plots [-h] [-p PARAMETERS]
                                -f SMODELSFOLDER -s SLHAFOLDER [-o OUTPUTFOLDER]
                                [-N NPOINTS] [-v VERBOSITY]
```

arguments:

- h, --help** show this help message and exit.
- p PARAMETERS, --parameters PARAMETERS** path to the parameters file [./iplots_parameters.py].
- f SMODELSFOLDER, --smodelsFolder SMODELSFOLDER** path to the smodels folder with the SModelS python output files.
- s SLHAFOLDER, --slhaFolder SLHAFOLDER** path to the SLHA folder with the SLHA input files.
- o OUTPUTFOLDER, --outputFolder OUTPUTFOLDER** path to the output folder, where the plots will be stored [./plots].
- N NPOINTS, --npoints NPOINTS** How many (randomly selected) points will be included in the plot. If -1, all points will be read and included [-1].
- v VERBOSITY, --verbosity VERBOSITY** Verbosity (debug, info, warning, error) [info].

The default values are given in []. A typical usage example is:

```
./smodelsTools.py interactive-plots \
  -f inputFiles/scanExample/smodels-output/ \
  -s inputFiles/scanExample/slha -p iplots_parameters.py \
  -o results/iplots
```

The above command will read the SModelS output files (in python format) from the folder `inputFiles/scanExample/smodels-output`, the corresponding SLHA input files from `inputFiles/scanExample/slha` and generate a set of HTML files with the interactive plots, which can be visualized with a regular web browser.

The settings in `iplots_parameters.py` include:

plot_title: main overall title for your plots, e.g. the model name.

x and y axes:: axis label, SLHA block and PDG code number of the variables you want to plot, in a python dictionary form. Example:⁶

```
variable_x = {'m<sub>gluino</sub>': ['MASS', 1000021]}
variable_y = {'m<sub>LSP</sub>': ['MASS', 1000022]}
```

spectrum hover information: defines which information from the input SLHA file will appear in the hover box. The syntax is again a python dictionary.

- ***slha_hover_information***: information from the input SLHA file, e.g. model parameters or masses. Example:

```
slha_hover_information = {'m(gluino)': ['MASS', 1000021],
  'm(chi10)': ['MASS', 1000022]}.
```

- ***BR_hover_information***: defines for which particle(s) to display decay channels and branching ratios. Example:

```
BR_hover_information = {'BR(gluino)': 1000021}.
```

The output is written in the form `.25[1000022, 1, -1]`, where the first number (0.25) is the branching ratio, and the numbers in `[]` are the PDG codes of the decay products.

⁶Html notation like `<sub>NAME</sub>` for subscript and `<sup>NAME</sup>` for superscript works for axes labels and in the spectrum hover information, but not in the SModelS hover information.

WARNING: Lists of branching ratios can be very long, so they may not fit in the hover box. One can define the number of entries with `BR_get_top` (default: all); e.g. `BR_get_top = 3` will display only the three largest branching ratios.

- `ctau_hover_information`: displays the mean decay length in meter for the listed particle(s). Example:
`ctau_hover_information = {'ctau(chi1+): 1000024'}.`

SModelS hover information: defines, as a list of keywords, which information to display from the SModelS output. Example:

```
smodels_hover_information = ['SModelS_excluded', 'r_max',
                             'Tx', 'Analysis', 'file'].
```

The options are:

- `SModelS_status`: prints whether the point is excluded or not by SModelS;
- `r_max`: shows the highest r-value for each parameter point (False if no experimental result applies);
- `chi2`: shows, if available, the χ^2 value associated to the highest r-value (if not, the output is False);
- `Tx`: shows the topology/ies which give the highest r-value;
- `Analysis`: shows the experimental analysis from which the strongest constraint (`r_max`) comes from;
- `MT_max`: shows the missing topology with the largest cross section (in SModelS bracket notation);
- `MT_max_xsec`: shows the cross section of `MT_max` in fb;
- `MT_total_xsec`: shows the total missing cross section in fb, i.e. the sum of all missing topologies cross sections;
- `MT_long_xsec`: shows the total missing cross section (in fb) in long cascade decays;
- `MT_asym_xsec`: shows the total missing cross section (in fb) in decays with asymmetric branches;

- `MT_outgrid_xsec`: shows the total missing cross section (in fb) outside the mass grids of the experimental results;
- `file`: shows the name of the input spectrum file.

Choice of plots to make :

- `plot_data`: choice of which points to plot; the options are: all, excluded, non-excluded points. Example:
`plot_data = ['all', 'excluded', 'non-excluded']`.
- `plot_list`: which quantities to plot in the x, y plane; the same options as for SModelS hover information apply. Example:
`plot_list = ['r_max', 'chi2', 'Tx', 'Analysis', 'MT_max', 'MT_max_xsec', 'MT_total_xsec']`.

The plotted quantities (`r_max`, `chi2`, `Tx`, `Analysis`, etc.) and kind of points (all, excluded, non-excluded by SModelS) are reflected in the interactive plots' filenames. Moreover, a file `index.html` is created for a convenient access to all plots.

6. Other updates in SModelS

6.1. Database update

In addition to the aggregated EMs and covariance matrix for CMS-PAS-SUSY-16-052 explained in section 3, and the HSCP and R-hadron results explained in section 4, the database was extended in several more ways:

- CMS Run 2 results: 84 new UL maps from 19 different CMS SUSY analyses with 36 fb^{-1} were added in v1.1.2. The impact of this update was shown in [18].
- ATLAS Run 2 results: 12 new UL maps from 6 ATLAS SUSY analyses with 36 fb^{-1} were added in v1.2.2, and one existing 3.2 fb^{-1} analysis was augmented with an additional EM result.
- Finally, EMs relevant for constraining gluino-squark production (see [19]) were produced by us by recasting ATLAS and CMS 8 TeV multi-jet + MET analyses with MadAnalysis 5 [20]. This resulted in 51 new EMs (for 8 TeV) added to the v1.2.2 database. Their impact will be analysed in detail in a separate publication [21].

The evolution of the database can also be traced via the wiki pages on the SMOBELS server: <http://smodels.hephy.at/wiki/ListOfAnalyses<VERSION>> and <http://smodels.hephy.at/wiki/SmsDictionary<VERSION>> document the state of the database in version <VERSION>, with the dots removed (i.e. 122 for version 1.2.2). Leaving <VERSION> empty returns the latest official version.

6.2. Bug fixes

- `smodelsTools.py` has fixpermissions for system-wide installations via `pip3`; this allows to install programs used by `smodelsTools` (Pythia6, Pythia8, NLLfast) into system-wide directories (if run with `sudo`);
- fixed small issue with pair production of $\mathbb{Z}_2$ -even particles;
- fixed `lastUpdate` bug: the date of the last update of a database entry is now printed correctly;
- fixed a problem in `particleNames.py` for non-MSSM models;
- fixed small issue when clustering elements with asymmetric masses.

7. Installation

SMODELS is a Python library that requires Python version 2.6 or later. It is fully compatible with Python 3, which is now the recommended default. The external Python dependencies are

```
unum>=4.0.0, numpy>=1.13.0, scipy>=1.0.0, argparse,
requests>=2.0.0, docutils>=0.3 and pyslha>=3.1.0.
```

These tools need not be installed separately, as the SMOBELS build system takes care of this. In addition, the cross section computer provided by `smodelsTools.py` requires a C++ compiler (for Pythia8 [22]) and a Fortran compiler (for NLLfast [23] and Pythia6 [24]). The current default is that both Pythia6 and Pythia8 are installed together with NLLfast. Finally, the database browser provided by `smodelsTools.py` requires IPython, while the interactive plotter requires plotly and pandas.

From v1.1.3 onwards, SMOBELS can be installed in its source directory. After downloading the code from

<https://github.com/SModelS/smodels/releases>

and extracting it, run

```
make smodels
```

in the distribution’s top-level directory. The installation will remove redundant folders, install the required dependencies (using `pip install`) and compile Pythia 8 and NLL-fast. If the installation fails with an error that no Fortran compiler was found, try

```
make FC=<path-to-fortran-compiler> smodels
```

If the cross section computer is not needed, one can replace `smodels` with `smodels.noexternaltools` in the `make` command. In case some Python libraries cannot be installed automatically, we recommend to install them separately with the user’s preferred method. Pythia and NLL-fast can also be compiled separately running `make externaltools`.

For more details, see the online documentation [11].

8. Conclusions

Version 1.2 of SModelS presented here features the combination of signal regions in efficiency map results whenever a covariance matrix is available from the experiment, and the implementation of HSCP and R-hadron signatures. The former is a crucial step towards fully exploiting the constraining power of efficiency map results, while the latter considerably extends the types of topologies which can be treated by SModelS. Several other improvements in the code increase the user-friendliness of the tool. These include, for instance, the use of wildcards in the selection of analyses, topologies or datasets, and a faster database which can be given as a URL. Moreover, an interactive plots maker was added to `smodelsTools` in order to conveniently visualize results of a model scan.

The v1.2.2 database includes 2733 individual results for 76 different SMS topologies from a total of 75 ATLAS and CMS analyses. This is not counting superseded results. In particular new 51 “home-grown” EMs for 8 TeV have been added, which were produced by us for better covering scenarios where gluino-squark associated production is important. For Run 2, the database now comprises 9 UL and 3 EM results from 10 ATLAS analyses, as well as

89 UL (of which 3 LLP) and 10 EM (of which 8 LLP) results from 21 CMS analyses. The addition of more ATLAS/CMS Run 2 results is ongoing.

Understanding how the plethora of LHC searches for new physics, conducted in specific final states, impact realistic new physics models, where possibly a multitude of production channels and decay modes are relevant, is a non-trivial task, but important for guiding future theoretical and experimental investigations. We will continue to further extend and improve SMOBELS to contribute to this aim.

Acknowledgements

We thank all SMOBELS users for their interest, feedback and encouragement.

J.D. is partially supported by funding available from the Department of Atomic Energy, Government of India, for the Regional Centre for Accelerator-based Particle Physics (RECAPP), Harish-Chandra Research Institute (HRI), and by an INFOSYS scholarship for senior students at the HRI. She thanks moreover the INFRE-HEPNET (IndoFrench Network on High Energy Physics) of CEFIPRA/IFCPAR (Indo-French Centre for the Promotion of Advanced Research) for support for research visits to LPSC Grenoble. J.H. acknowledges support from the F.R.S.-FNRS, of which he is a postdoctoral researcher, and S.K. from the IN2P3 project “Théorie – LHCiTools” and the CNRS-FAPESP collaboration grant PRC275431. Su.K. is supported by the New Frontiers program of the Austrian Academy of Sciences and the FWF under project number V592-N27. A.L. acknowledges support the Sao Paulo Research Foundation FAPESP projects 2015/20570-1 and 2016/50338-6 and H.R.G. from the Consejo Nacional de Ciencia y Tecnologia, CONACyT, scholarship no. 291169.

References

- [1] F. Ambrogio, S. Kraml, S. Kulkarni, U. Laa, A. Lessa, V. Magerl, J. Sonn-eveld, M. Traub, W. Waltenberger, SModelS v1.1 user manual: Improv-ing simplified model constraints with efficiency maps, *Comput. Phys. Commun.* 227 (2018) 72–98. [arXiv:1701.06586](#), [doi:10.1016/j.cpc.2018.02.007](#).
- [2] CMS Collaboration, Simplified likelihood for the re-interpretation of public CMS results, *Tech. Rep. CMS-NOTE-2017-001*, CERN, Geneva, <https://cds.cern.ch/record/2242860> (Jan 2017).
- [3] J. Heisig, S. Kraml, A. Lessa, Constraining new physics with searches for long-lived particles: Implementation into SModelS, *Phys. Lett. B* (2018) in print. [arXiv:1808.05229](#).
- [4] <https://github.com/SModelS/smodels-database-release/releases>.
- [5] M. Papucci, K. Sakurai, A. Weiler, L. Zeune, Fastlim: a fast LHC limit calculator, *Eur. Phys. J. C* 74 (11) (2014) 3163. [arXiv:1402.0492](#), [doi:10.1140/epjc/s10052-014-3163-1](#).
- [6] <http://smodels.hephy.at/wiki/SmsDictionary>.
- [7] A. Buckley, M. Citron, S. Fichet, S. Kraml, W. Waltenberger, N. Wardle, The Simplified Likelihood Framework. [arXiv:1809.05548](#).
- [8] A. L. Read, Presentation of search results: The CL(s) technique, *J. Phys. G* 28 (2002) 2693–2704. [doi:10.1088/0954-3899/28/10/313](#).
- [9] G. Cowan, K. Cranmer, E. Gross, O. Vitells, Asymptotic formulae for likelihood-based tests of new physics, *Eur. Phys. J. C* 71 (2011) 1554, [Erratum: *Eur. Phys. J. C* 73, 2501 (2013), [doi:10.1140/epjc/s10052-013-2501-z](#)]. [arXiv:1007.1727](#), [doi:10.1140/epjc/s10052-011-1554-0](#).
- [10] <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.brentq.html>.
- [11] <https://smodels.readthedocs.io/en/stable/index.html>.

- [12] A. M. Sirunyan, et al., Search for supersymmetry in proton-proton collisions at 13 TeV using identified top quarks, *Phys. Rev. D* **97** (1) (2018) 012007. [arXiv:1710.11188](#), [doi:10.1103/PhysRevD.97.012007](#).
- [13] CMS Collaboration, Search for supersymmetry in events with at least one soft lepton, low jet multiplicity, and missing transverse momentum in proton-proton collisions at $\sqrt{s} = 13$ TeV, Tech. Rep. CMS-PAS-SUS-16-052, <https://cds.cern.ch/record/2273394> (2017).
- [14] J. Heisig, A. Lessa, L. Quertenmont, Simplified Models for Exotic BSM Searches, *JHEP* **12** (2015) 087. [arXiv:1509.00473](#), [doi:10.1007/JHEP12\(2015\)087](#).
- [15] S. Chatrchyan, et al., Searches for long-lived charged particles in pp collisions at $\sqrt{s}=7$ and 8 TeV, *JHEP* **07** (2013) 122. [arXiv:1305.0491](#), [doi:10.1007/JHEP07\(2013\)122](#).
- [16] CMS Collaboration, Search for heavy stable charged particles with 12.9 fb⁻¹ of 2016 data, Tech. Rep. CMS-PAS-EXO-16-036, CERN, Geneva, <https://cds.cern.ch/record/2205281> (2016).
- [17] V. Khachatryan, et al., Constraints on the pMSSM, AMSB model and on other models from the search for long-lived charged particles in proton-proton collisions at $\sqrt{s} = 8$ TeV, *Eur. Phys. J. C* **75** (7) (2015) 325. [arXiv:1502.02522](#), [doi:10.1140/epjc/s10052-015-3533-3](#).
- [18] J. Dutta, S. Kraml, A. Lessa, W. Waltenberger, SModelS extension with the CMS supersymmetry search results from Run 2, *LHEP* **1** (1) (2018) 5–12. [arXiv:1803.02204](#), [doi:10.31526/LHEP.1.2018.02](#).
- [19] F. Ambrogio, S. Kraml, S. Kulkarni, U. Laa, A. Lessa, W. Waltenberger, On the coverage of the pMSSM by simplified model results, *Eur. Phys. J. C* **78** (3) (2018) 215. [arXiv:1707.09036](#), [doi:10.1140/epjc/s10052-018-5660-0](#).
- [20] B. Dumont, B. Fuks, S. Kraml, S. Bein, G. Chalons, E. Conte, S. Kulkarni, D. Sengupta, C. Wymant, Toward a public analysis database for LHC new physics searches using MADANALYSIS 5, *Eur. Phys. J. C* **75** (2) (2015) 56. [arXiv:1407.3278](#), [doi:10.1140/epjc/s10052-014-3242-3](#).

- [21] F. Ambrogio, et al., *in preparation*.
- [22] T. Sjostrand, S. Mrenna, P. Z. Skands, A Brief Introduction to PYTHIA 8.1, Comput. Phys. Commun. 178 (2008) 852–867. [arXiv:0710.3820](#), [doi:10.1016/j.cpc.2008.01.036](#).
- [23] http://pauli.uni-muenster.de/~akule_01/nllwiki/index.php/NLL-fast.
- [24] T. Sjostrand, S. Mrenna, P. Z. Skands, PYTHIA 6.4 Physics and Manual, JHEP 05 (2006) 026. [arXiv:hep-ph/0603175](#), [doi:10.1088/1126-6708/2006/05/026](#).