SModelS v2.0 tutorial

(Re)interpreting the results of new physics searches at the LHC
February 16, 2021

https://smodels.github.io/

Gaël Alguero, Jan Heisig, Charanjit K. Khosa, Sabine Kraml, Suchita Kulkarni, Andre Lessa,
Philipp Neuhuber, Humberto Reyes-Gonzalez, Wolfgang Waltenberger, Alicia Wongel

Everything you need is explained on **smodels.github.io**

see also talk “SModelS v2.0: new features and developments” by Andre Lessa.

- A detailed documentation is available in the **online manual**
- For instructions on how to install SModelS, check the **installation** section in the manual.
- You may also want to check the release notes and known issues
- Here are the list of analyses in the latest database version, the respective validation plots and an SMS dictionary explaining the Tx names used by SModelS.

Mailing lists:

- For questions and comments, send an e-mail to: smodels-users@lists.oeaw.ac.at.
- To receive updates and announcements, subscribe to smodels-info.

… and lots more useful infos ….

Requirements

SModelS is a Python package; v2.0 has been developed and tested with **Python 3**
It depends on the following ***external* Python libraries**:

- unum>=4.0.0
- numpy>=1.13.0
- argparse
- requests>=2.0.0
- docutils>=0.3
- scipy>=1.0.0
- pyslha>=3.1.0
- pyhf>=0.4.3 (>=0.5.2 recommended!)
- jsonpatch>=1.25
- jsonschema>=3.2.0

(+ recommended for pyhf: pytorch)

The cross section computer provided by smodelsTools.py requires:

- Pythia 8.2 (requires a C++ compiler) or Pythia 6.4.27 (requires fortran)
- NLL-fast 1.2, 2.1, and 3.1 (requires a fortran compiler)

These tools need not be installed separately, as the SModelS build system takes care of that.

The database browser provided by smodelsTools.py requires IPython, while the interactive plotter requires plotly and pandas.

https://smodels.readthedocs.io/en/latest/Installation.html

Standard installation

Download the v2.0.0 (beta) from https://github.com/SModelS/smodels/releases and extract it in a source directory, e.g.:

```
> tar -zxvf smodels-2.0.0-beta.tar.gz
> cd smodels-2.0.0-beta
```

then run

```
> make smodels   (or: make FC=<path_to_fortran> smodels)
```

in the top-level directory. This will install the required dependencies (using pip install) and compile Pythia and NLL-fast.

If the (MSSM) cross section computer is not needed, run instead

```
> make smodels_noexternaltools
```

In case the Python libraries cannot be successfully installed, the user can install them separately using his/her preferred method. Pythia and NLL-fast can also be compiled separately running `make externaltools`.

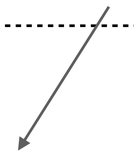
Alternatively:

- using python setuptools:

```
setup.py install [--user]
```

- using pip:

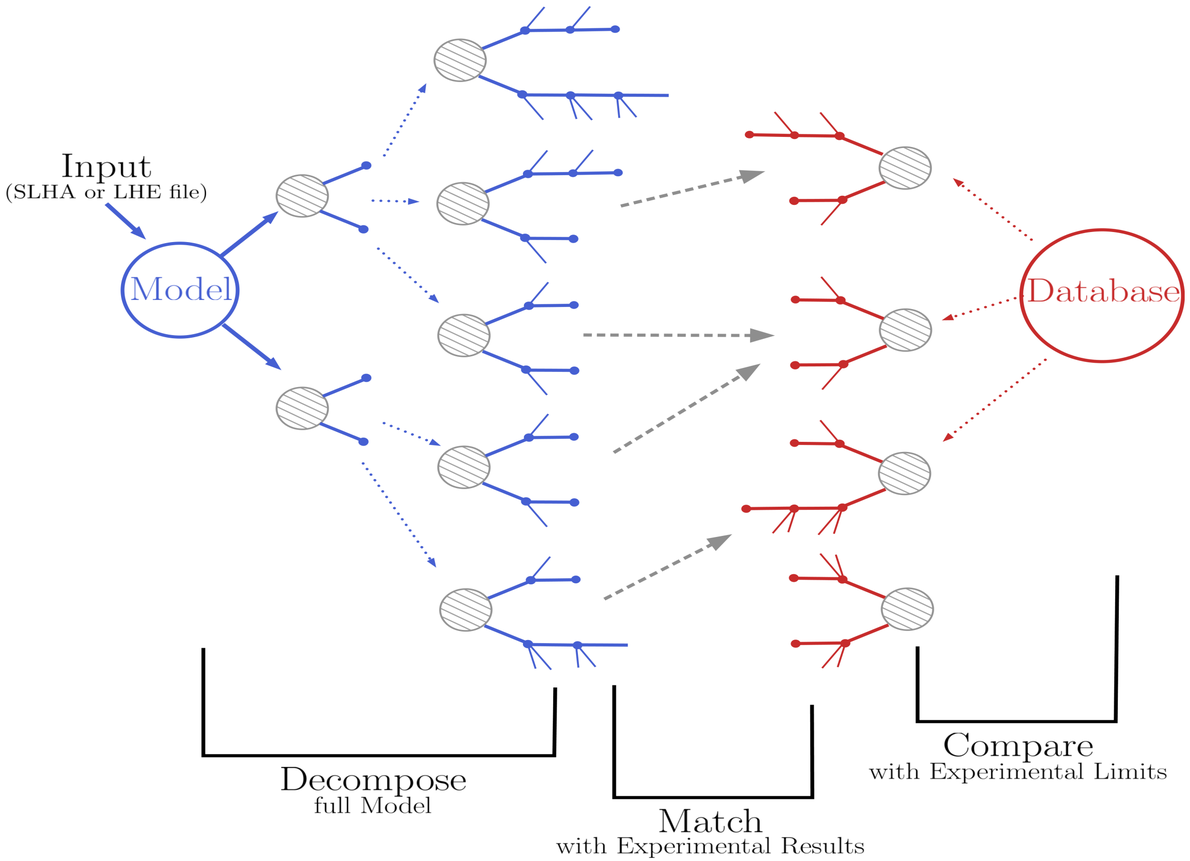
```
pip3 install [--user] smodels==2.0.0b0
```

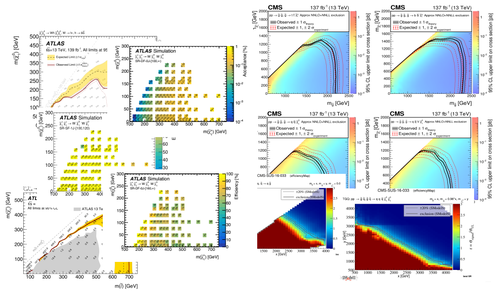


https://smodels.readthedocs.io/en/latest/Installation.html

Basic principle

$$\mathcal{L} = \mathcal{L}_{SM} + \mathcal{L}_{BSM}$$





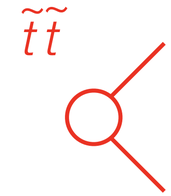
Basic principle

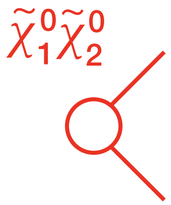
spectrum.slha:

```
BLOCK MASS  # Mass Spectrum
# PDG code           mass       particle
[…]
   1000006     6.48255292E+02   # ~t_1
   1000022     3.00681405E+02   # ~chi_10
   1000023     3.06894404E+02   # ~chi_20
[…]

#         PDG            Width
DECAY   1000006     1.03408969E+01   # stop1 decays
#          BR         NDA      ID1       ID2
     2.30885744E-01    2     1000022       6   # BR(~t_1 -> ~chi_10 t )
     2.11305478E-01    2     1000023       6   # BR(~t_1 -> ~chi_20 t )
     5.57808778E-01    2     1000024       5   # BR(~t_1 -> ~chi_1+ b )
[…]
#         PDG            Width
DECAY   1000022     0.00000000E+00   # neutralino1 decays
#         PDG            Width
DECAY   1000023     7.99513486E-09   # neutralino2 decays
#          BR         NDA      ID1       ID2
     2.51986629E-02    2     1000022      22   # BR(~chi_20 -> ~chi_10 gam)
[…]

XSECTION  1.30E+04  2212 2212 2 -1000006 1000006 # 10000 events, [pb], pythia8 for LO
  0  2  0  0  0  0    1.06548924E-01 SModelSv1.1.3rc3
XSECTION  1.30E+04  2212 2212 2 1000022 1000023 # 10000 events, [pb], pythia8 for LO
  0  0  0  0  0  0    3.68210665E-02 SModelSv1.1.3rc3
[…]
```





…

For production mode:

σ_{prod}

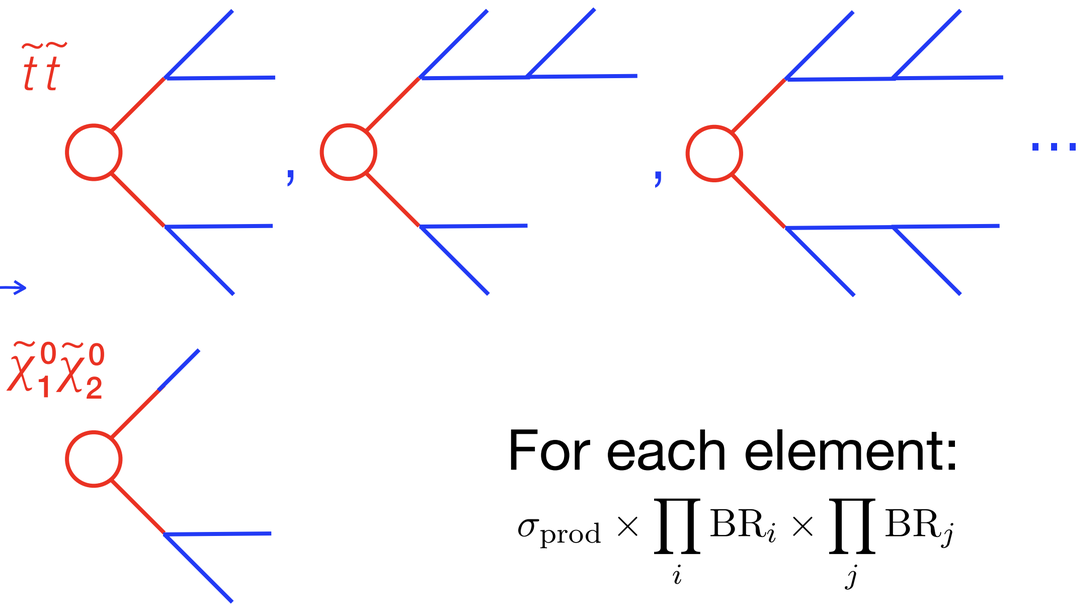
Basic principle

spectrum.slha:

```
BLOCK MASS  # Mass Spectrum
# PDG code           mass       particle
[…]
   1000006     6.48255292E+02   # ~t_1
   1000022     3.00681405E+02   # ~chi_10
   1000023     3.06894404E+02   # ~chi_20
[…]

#         PDG            Width
DECAY   1000006     1.03408969E+01   # stop1 decays
#          BR         NDA      ID1       ID2
     2.30885744E-01    2     1000022      6   # BR(~t_1 -> ~chi_10 t )
     2.11305478E-01    2     1000023      6   # BR(~t_1 -> ~chi_20 t )
     5.57808778E-01    2     1000024      5   # BR(~t_1 -> ~chi_1+ b )
[…]
#         PDG            Width
DECAY   1000022     0.00000000E+00   # neutralino1 decays
#         PDG            Width
DECAY   1000023     7.99513486E-09   # neutralino2 decays
#          BR         NDA      ID1       ID2
     2.51986629E-02    2     1000022     22   # BR(~chi_20 -> ~chi_10 gam)
[…]

XSECTION   1.30E+04  2212 2212 2 -1000006 1000006 # 10000 events, [pb], pythia8 for LO
  0   2   0   0   0   0     1.06548924E-01 SModelSv1.1.3rc3
XSECTION   1.30E+04  2212 2212 2 1000022 1000023 # 10000 events, [pb], pythia8 for LO
  0   0   0   0   0   0     3.68210665E-02 SModelSv1.1.3rc3
[…]
```



For each element:

$$\sigma_{\text{prod}} \times \prod_i \text{BR}_i \times \prod_j \text{BR}_j$$

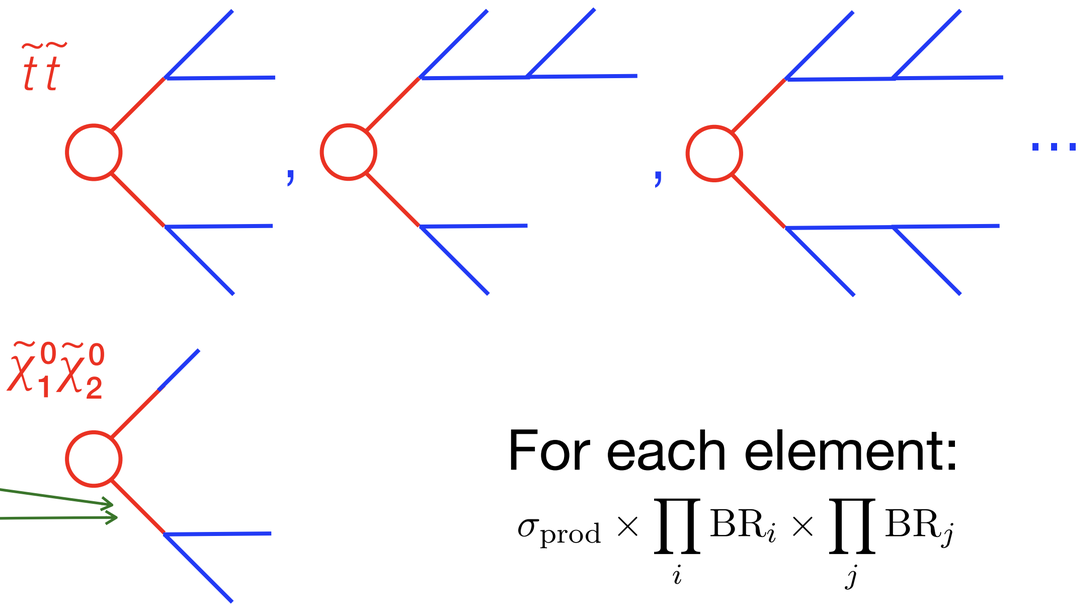
Basic principle

spectrum.slha:

```
BLOCK MASS  # Mass Spectrum
# PDG code           mass       particle
[…]
   1000006     6.48255292E+02   # ~t_1
   1000022     3.00681405E+02   # ~chi_10
   1000023     3.06894404E+02   # ~chi_20
[…]

#         PDG            Width
DECAY   1000006     1.03408969E+01   # stop1 decays
#          BR         NDA      ID1       ID2
     2.30885744E-01    2     1000022       6   # BR(~t_1 -> ~chi_10 t )
     2.11305478E-01    2     1000023       6   # BR(~t_1 -> ~chi_20 t )
     5.57808778E-01    2     1000024       5   # BR(~t_1 -> ~chi_1+ b )
[…]
#         PDG            Width
DECAY   1000022     0.00000000E+00   # neutralino1 decays
#         PDG            Width
DECAY   1000023     7.99513486E-09   # neutralino2 decays
#          BR         NDA      ID1       ID2
     2.51986629E-02    2     1000022      22   # BR(~chi_20 -> ~chi_10 gam)
[…]

XSECTION   1.30E+04  2212 2212 2 -1000006 1000006 # 10000 events, [pb], pythia8 for LO
  0  2  0  0  0  0    1.06548924E-01 SModelSv1.1.3rc3
XSECTION   1.30E+04  2212 2212 2 1000022 1000023 # 10000 events, [pb], pythia8 for LO
  0  0  0  0  0  0    3.68210665E-02 SModelSv1.1.3rc3
[…]
```



For each element:

$$\sigma_{\text{prod}} \times \prod_i \text{BR}_i \times \prod_j \text{BR}_j$$

mass, width* & further qnumbers* stored for each BSM particle

* new in v2.0!

Basic principle

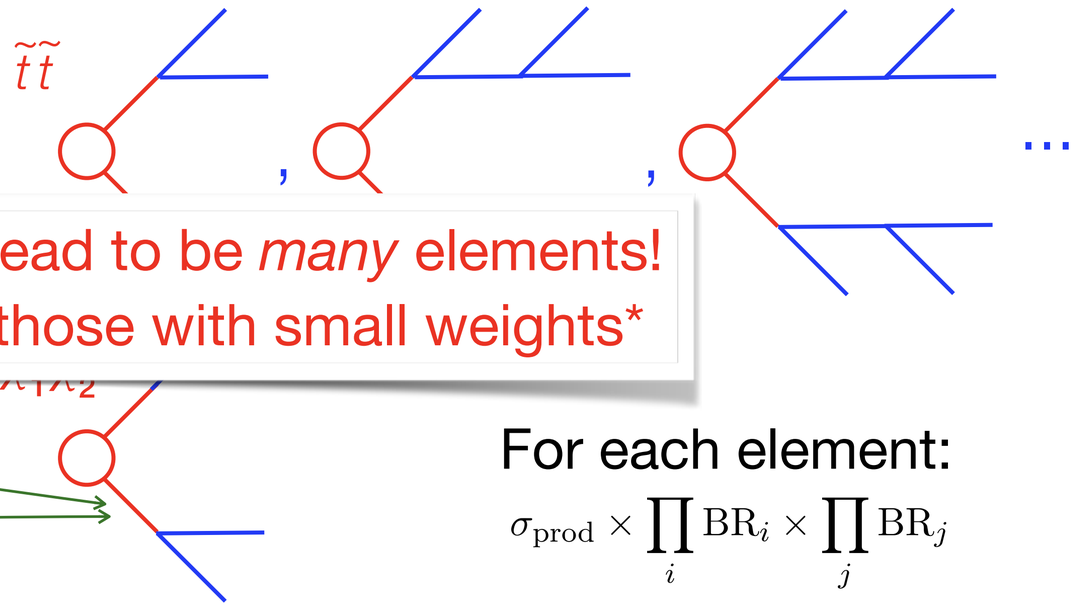
spectrum.slha:

```
BLOCK MASS  # Mass Spectrum
# PDG code           mass       particle
[…]
   1000006     6.48255292E+02   # ~t_1
   1000022     3.00681405E+02   #
   1000023     3.06894404E+02   #
[…]

#         PDG            Width
DECAY   1000006     1.03408969E+01
#          BR         NDA      ID1       ID2
     2.30885744E-01    2     1000022       6   # BR(~t_1 -> ~chi_10 t )
     2.11305478E-01    2     1000023       6   # BR(~t_1 -> ~chi_20 t )
     5.57808778E-01    2     1000024       5   # BR(~t_1 -> ~chi_1+ b )
[…]
#         PDG            Width
DECAY   1000022     0.00000000E+00   # neutralino1 decays
#         PDG            Width
DECAY   1000023     7.99513486E-09   # neutralino2 decays
#          BR         NDA      ID1       ID2
     2.51986629E-02    2     1000022      22   # BR(~chi_20 -> ~chi_10 gam)
[…]

XSECTION   1.30E+04  2212 2212 2 -1000006 1000006 # 10000 events, [pb], pythia8 for LO
  0  2  0  0  0  0    1.06548924E-01 SModelSv1.1.3rc3
XSECTION   1.30E+04  2212 2212 2 1000022 1000023 # 10000 events, [pb], pythia8 for LO
  0  0  0  0  0  0    3.68210665E-02 SModelSv1.1.3rc3
[…]
```

Note: the decomposition may lead to be *many* elements!
Use `sigmacut` (in fb) to omit those with small weights*



For each element:

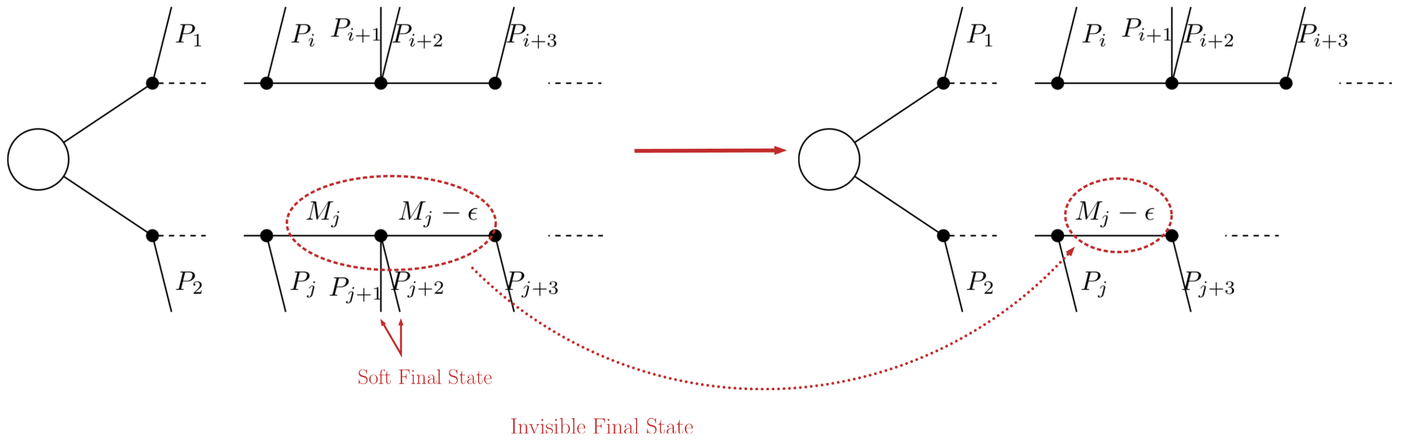
$$\sigma_{\text{prod}} \times \prod_i \text{BR}_i \times \prod_j \text{BR}_j$$

* Rule of thumb: Adjust to one order of magnitude below the minimum signal cross sections the experimental data can constrain.

Mass and invisible compression

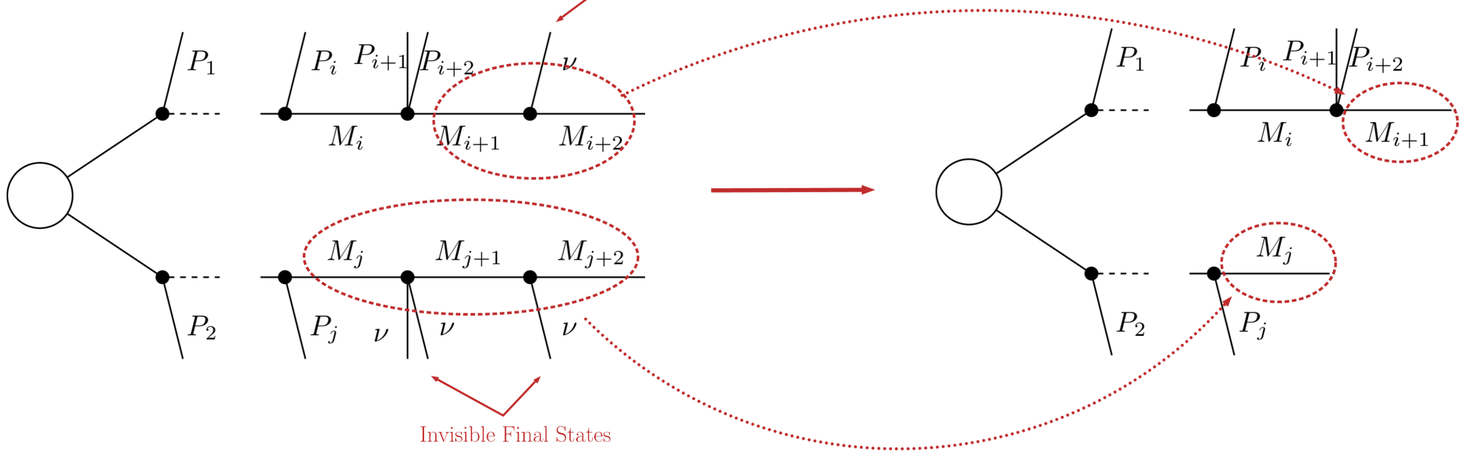
Mass compression:

- enabled by `doCompress`
- maximum gap for application set by `minmassgap` (in GeV)

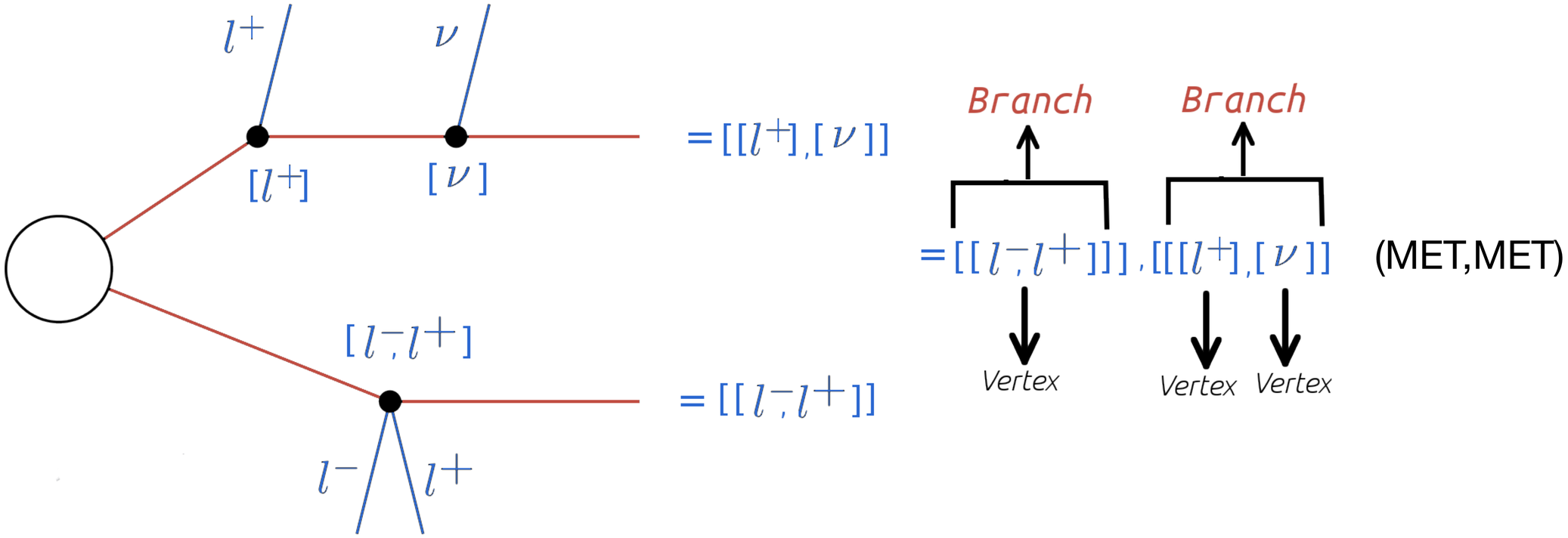


Invisible compression:

- enabled by `doInvisible`

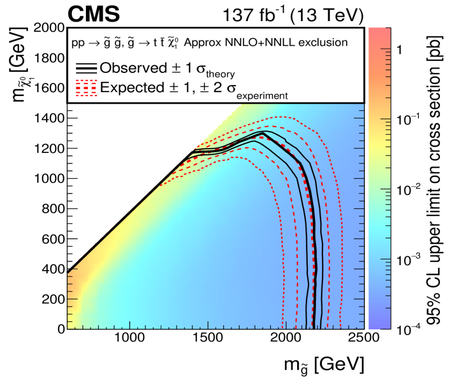


SModelS bracket notation



Database result types

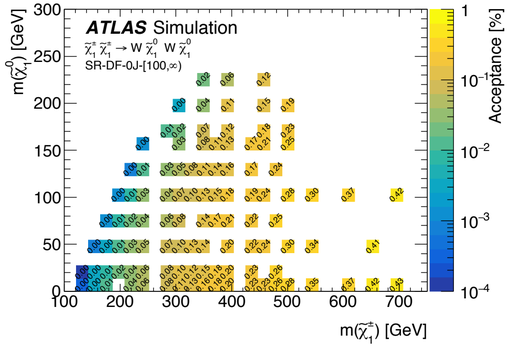
Upper limits (UL)



$$\sigma_{\text{prod}} \times \prod_i \text{BR}_i \times \prod_j \text{BR}_j \leqslant \sigma^{\text{UL}}_{\text{prod}}$$

- Signal regions usually combined
- Not restricted to cut-and-count
- No combinations of topologies

Efficiency maps (EM)

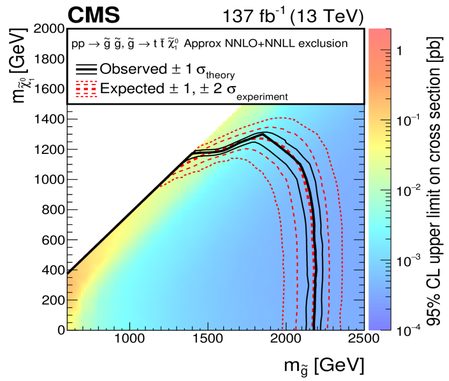


$$\sigma_{\text{prod}} \times \prod_i \text{BR}_i \times \prod_j \text{BR}_j \times \epsilon_s \leqslant \sigma^{\text{UL}}_{\text{fid},\, s}$$

- Different topologies can easily be combined (within the same signal region)
- Likelihoods can be computed
- Combination of signal region if covariance/ full likelihood supplied

Database result types

Upper limits (UL)



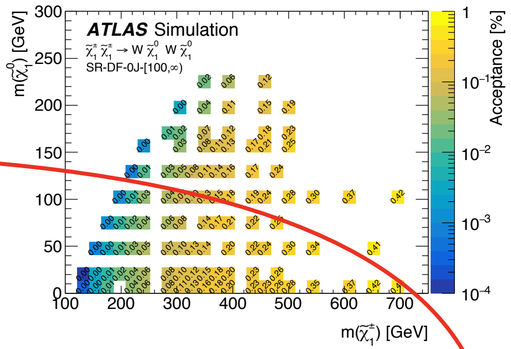
In general, depends on:

- masses
- width
- q-numbers

$$\sigma_{\text{prod}} \times \prod_i \text{BR}_i \times \prod_j \text{BR}_j \leqslant \sigma^{\text{UL}}_{\text{prod}}$$

- Signal regions usually combined
- Not restricted to cut-and-count
- No combinations of topologies

Efficiency maps (EM)



$$\sigma_{\text{prod}} \times \prod_i \text{BR}_i \times \prod_j \text{BR}_j \times \epsilon_s \leqslant \sigma^{\text{UL}}_{\text{fid},\, s}$$

- Different topologies can easily be combined (within the same signal region)
- Likelihoods can be computed
- Combination of signal region if covariance/ full likelihood supplied

Using SModelS: runSModelS.py

SModelS can take **SLHA files** (with masses, decay tables and *cross sections* !) **or LHE files** as input.
It ships with a command-line tool `runSModelS.py`, which reports on the SMS decomposition and theory predictions in several output formats.

The usage of runSModelS is:

```
> ./runSModelS.py [-h] -f FILENAME [-p PARAMETERFILE] [-o OUTPUTDIR]
```

- -h : show help message and exit
- -f : SLHA or LHE input file(s); can be a directory (needs Decay tables and XSs!)
- -p : name of parameters file (default is ./smodels/etc/parameters_default.ini)
- -o : output directory
- other optional arguments: verbosity level, colored output, timeout, etc.

https://smodels.readthedocs.io/en/latest/RunningSModelS.html

The `parameters.ini` file

The basic options and parameters used by `runSModelS.py` are defined in the parameters file. A commented example, including all available parameters together with short descriptions, is given in `parameters.ini` in the top-level directory.

```
[options]
checkInput = True ;Set True to check the input file for possible errors
doInvisible = True ;Set True if invisible compression should be performed, False elsewise
doCompress = True ;Set True if mass compression should be performed, False elsewise
computeStatistics = True ;Set True to compute likelihood and chi2 for the most sensitive EM result,
False elsewise
testCoverage = True ;Set True if topologies not covered by experiments (missing topologies) should be
identified, False otherwise
combineSRs = False ;Set True to combine signal regions when covariance matrix is available. Might take
a few secs per point. False uses only best SR.
```

https://smodels.readthedocs.io/en/latest/RunningSModelS.html#the-parameters-file

The `parameters.ini` file (cont.)

The basic options and parameters used by `runSModelS.py` are defined in the parameters file. A commented example, including all available parameters together with short descriptions, is given in `parameters.ini` in the top-level directory.

```
[particles]
model=share.models.mssm ; path to the BSM model file. It can be a python module with definition of BSM
particles or a SLHA file with QNUMBERS blocks. If omitted, we search in the current working  directory
as well as "smodels/share/models". MSSM is the default.
promptWidth = 1e-8 ; All particles with widths (in GeV) above this value are considered prompt
stableWidth = 1e-25 ; All particles with widths (in GeV) below this value are considered stable

[parameters]
sigmacut = 0.01 ; Give minimum cross section value [fb] considered in SLHA decomposition (relevant for
SLHA decomposition and detection of missing topologies)
minmassgap = 5. ; Give minimum mass gap [GeV] for mass compression
maxcond = 0.2 ; Maximum relative violation of conditions for valid results
ncpus = 1 ; Give number of cores used when running in parallel (integer, -1 means all available CPUs
are used). Warning: do not change unless you know what you are doing!
```

https://smodels.readthedocs.io/en/latest/RunningSModelS.html#the-parameters-file

The `parameters.ini` file (cont.)

The basic options and parameters used by `runSModelS.py` are defined in the parameters file. A commented example, including all available parameters together with short descriptions, is given in `parameters.ini` in the top-level directory.

```
[particles]
model=inputFiles/slha/idm_example.slha ; path to the BSM model file. It can be a python module with
definition of BSM particles or a SLHA file with QNUMBERS blocks. If omitted, we search in the current
working  directory as well as "smodels/share/models". MSSM is the default.
promptWidth = 1e-8 ; All particles with widths (in GeV) above this value are considered prompt
stableWidth = 1e-25 ; All particles with widths (in GeV) below this value are considered stable
```

https://smodels.readthedocs.io/en/latest/RunningSModelS.html#the-parameters-file

The `parameters.ini` file (cont.)

The basic options and parameters used by `runSModelS.py` are defined in the parameters file. A commented example, including all available parameters together with short descriptions, is given in `parameters.ini` in the top-level directory.

```
[particles]
model=share.models.mssm ; path to the BSM model file. It can be a python module with definition of BSM
particles or a SLHA file with QNUMBERS blocks. If omitted, we search in the current working  directory
as well as "smodels/share/models". MSSM is the default.
promptWidth = 1e-8 ; All particles with widths (in GeV) above this value are considered prompt
stableWidth = 1e-25 ; All particles with widths (in GeV) below this value are considered stable

[parameters]
sigmacut = 0.01 ; Give minimum cross section value [fb] considered in SLHA decomposition (relevant for
SLHA decomposition and detection of missing topologies)
minmassgap = 5. ; Give minimum mass gap [GeV] for mass compression
maxcond = 0.2 ; Maximum relative violation of conditions for valid results
ncpus = 1 ; Give number of cores used when running in parallel (integer, -1 means all available CPUs
are used). Warning: do not change unless you know what you are doing!
```

https://smodels.readthedocs.io/en/latest/RunningSModelS.html#the-parameters-file

The `parameters.ini` file (cont.)

The basic options and parameters used by `runSModelS.py` are defined in the parameters file. A commented example, including all available parameters together with short descriptions, is given in `parameters.ini` in the top-level directory.

```
[database]
path = official ; URL to the database pickle file (it will be downloaded)

analyses = all ; Set all to use all analyses included in the database
#to use only specific analyses, give a list of the names separated by comma
# analyses = CMS-PAS-SUS-13-008, CMS-SUS-13-013,ATLAS-CONF-2013-024,ATLAS-SUSY-2013-04
# Wildcards are understood as in shell-expansion of file names: * ? [<list of letters>]
# Filter centre-of-mass energy with suffix beginning with a colon, in unum-style, like :13*TeV
# Note that the asterisk in the suffix is not a wildcard.

txnames= all ; Set all to use all constraints included in the database
#to use only specific constraints, give a list of the names separated by comma
#txnames = T2,T1,TChiWZ
# Wildcards are understood as in shell-expansion of file names: * ? [<list of letters>]
```

https://smodels.readthedocs.io/en/latest/RunningSModelS.html#the-parameters-file

The `parameters.ini` file (cont.)

The basic options and parameters used by `runSModelS.py` are defined in the parameters file. A commented example, including all available parameters together with short descriptions, is given in `parameters.ini` in the top-level directory.

```
dataselector= all ; Set all to use all upper limit and efficiency maps results in the database. Set to
upperLimit (efficiencyMap) to use only UL (EM) results:
#dataselector = upperLimit
#It can also be used to select specific datasets (signal regions) from efficiency map results. For the
latter provide a list of the desired dataset ids
#dataselector = SRA mCT150,SRA mCT200
# Wildcards are understood as in shell-expansion of file names: * ? [<list of letters>]

[printer]
outputType = python,summary ; Define the output formats
#available output formats: summary, stdout, log, python, xml, slha
(type log redirects stdout in *.log output file)
```

For detailed description of [each type of] output, see OutputDescription section in the online manual.

https://smodels.readthedocs.io/en/latest/RunningSModelS.html#the-parameters-file

Examples

Copy `inputFiles/` and `plotting/` from

https://indico.cern.ch/event/982553/contributions/4220784/attachments/2184008/3702004/files_tutorial_rif_2021.zip

or

https://github.com/SModelS/tutorials/tree/rif2021

in your SModelS source directory

...or run SModelS with mybinder:

https://mybinder.org/v2/gh/SModelS/tutorials/rif2021?filepath=index.ipynb

'Trivial' example

```
> ./runSModelS.py -f slhaFiles/simplyGluino.slha -p parameters.ini -o results/
```

Input file: `slhaFiles/simplyGluino.slha` ← basically a simplified model

```
BLOCK MASS  # Mass Spectrum
# PDG code           mass       particle
[…]
   1000016     1.00000000E+04   # ~nu_tauL
   1000021     675.0            # ~g
   1000022     200.0            # ~chi_10
   1000023     1.00000000E+04   # ~chi_20
[…]
#          PDG            Width
DECAY   1000021     1.00000000E+00   # gluino decays
#          BR         NDA      ID1       ID2     ID3
    0.50000000E+00    3     1000022    -1       1    # BR(~gl -> N1 dbar d)
    0.50000000E+00    3     1000022    -2       2    # BR(~gl -> N1 ubar u)
#         PDG            Width
DECAY   1000022     0.00000000E+00   # neutralino1 decays
[…]
XSECTION  8.00E+03  2212 2212 2 1000021 1000021   # Nevts: 10000 xsec unit: pb
      0   2   0   0   0   0    5.72168935E-01 SModelS 0.99

XSECTION  1.30E+04  2212 2212 2 1000021 1000021   # Nevts: 10000 xsec unit: pb
  0   2   0   0   0   0    4.30903465E+00 SModelS 1.0.93
[…]
```

'Trivial' example (cont.)

```
> ./runSModelS.py -f slhaFiles/simplyGluino.slha -p parameters.ini -o results/
```

Output file: `results/simplyGluino.slha.smodels`

$$r = \frac{\sigma(\text{signal})}{\sigma(95\%\text{CL})}$$, r ≥ 1 considered excluded

```
Input status: 1
Decomposition output status: 1 #decomposition was successful
# Input File: slhaFiles/simplyGluino.slha
[…]
# Database version: 2.0.0-beta
================================================================================
#Analysis   Sqrts   Cond_Violation   Theory_Value(fb)   Exp_limit(fb)   r   r_expected

     CMS-SUS-16-033   1.30E+01     0.0   4.309E+03   1.990E+01   2.165E+02   1.088E+02
 Signal Region:   (UL)
 Txnames:   T1
--------------------------------------------------------------------------------
     CMS-SUS-16-036   1.30E+01     0.0   4.309E+03   2.782E+01   1.549E+02   8.619E+01
 Signal Region:   (UL)
 Txnames:   T1
--------------------------------------------------------------------------------
[…]
```

(largest *r* on top)

Cross section upper limit (UL) result, only one topology

'Trivial' example (cont.)

```
> ./runSModelS.py -f slhaFiles/simplyGluino.slha -p parameters.ini -o results/
```

Output file: `results/simplyGluino.slha.smodels`

```
[…]
================================================================================
The highest r value is = 216.508383510753
```

excluded

```
Total cross-section for missing topologies (fb):  0.000E+00
Total cross-section for missing topologies with displaced decays (fb):  0.000E+00
Total cross-section for missing topologies with prompt decays (fb):  0.000E+00
Total cross-section for topologies outside the grid (fb):  0.000E+00

Full information on unconstrained cross sections
================================================================================
No missing topologies found
================================================================================
No missing topologies with displaced decays found
================================================================================
No missing topologies with prompt decays found
================================================================================
No topologies outside the grid found
================================================================================
```

Summary information

Accordingly, no missing topologies or topologies outside the grids

MSSM example

```
> ./runSModelS.py -f slhaFiles/wino_11010599.slha -p parameters.ini -o results/
```

Output file: `results/wino_11010599.slha.smodels`

```
Input status: 1
Decomposition output status: 1 #decomposition was successful
# Input File: slhaFiles/wino_11010599.slha
[…]
# Database version: 2.0.0-beta
================================================================================
#Analysis   Sqrts   Cond_Violation   Theory_Value(fb)   Exp_limit(fb)   r   r_expected

 ATLAS-SUSY-2016-06   1.30E+01     0.0   3.144E+00   2.200E-01   1.429E+01   1.123E+01
 Signal Region:   SR_EW
 Txnames:   TDTM1F, TDTM2F
 Chi2, Likelihood =   1.769E+02   6.444E-41
--------------------------------------------------------------------------------
     CMS-SUS-19-006   1.30E+01     0.0   1.171E+00   1.648E+00   7.105E-01   7.715E-01
 Signal Region:   (UL)
 Txnames:   T2
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-07   1.30E+01     0.0   1.171E+00   1.954E+00   5.992E-01   N/A
 Signal Region:   (UL)
 Txnames:   T2
[…]
```

Efficiency map (EM) result, sum over several topologies, Chi2 and likelihood evaluated

Cross section upper limit (UL) result, only one topology

Cross section upper limit (UL) result, only one topology

MSSM example (cont.)

```
> ./runSModelS.py -f slhaFiles/wino_11010599.slha -p parameters.ini -o results/
```

Output file: `results/wino_11010599.slha.smodels`

```
[…]
================================================================================
The highest r value is = 14.288770889808

Total cross-section for missing topologies (fb):  2.512E+01
Total cross-section for missing topologies with displaced decays (fb):  3.626E+01
Total cross-section for missing topologies with prompt decays (fb):  1.596E+02
Total cross-section for topologies outside the grid (fb):  1.782E+01

Full information on unconstrained cross sections
================================================================================
missing topologies with the highest cross sections (up to 10):
Sqrts (TeV)   Weight (fb)                  Element description
 13.0          2.374E+00    #                      [[[W]],[[W],[jet]]]  (MET,MET)
 13.0          2.315E+00    #                  [[[W],[jet]],[[Z],[jet]]]  (MET,MET)
[…]
================================================================================
missing topologies with displaced decays with the highest cross sections (up to 10):
Sqrts (TeV)   Weight (fb)                  Element description
 13.0          8.463E+00    #                    [[[jet]],[[jet],[jet]]]  (MET,MET)
 13.0          4.786E+00    #              [[[jet],[jet]],[[jet],[jet]]]  (MET,MET)
[…]
```

Summary information

Topologies not covered by the results in the database (SModelS bracket notation)

Parameter scan in Inert Doublet Model (IDM)

Change in parameters.ini: `model=share.models.idm` and ensure that `outputType` contains `python`

```
> ./runSModelS.py -f slhaFiles/IDM_points/ -p parameters.ini -o results/IDM_output/
```

```
Input status: 1
Decomposition output status: 1 #decomposition was successful
# Input File: slhaFiles/IDM_points/30000000105_paramcard.slha
[…]
# Database version: 2.0.0-beta
================================================================================
#Analysis   Sqrts   Cond_Violation   Theory_Value(fb)   Exp_limit(fb)   r   r_expected

     CMS-EXO-13-006   8.00E+00     0.0   7.783E-02   1.700E-01   4.578E-01   3.538E-01
 Signal Region:  c200
 Txnames:  THSCPM1b, THSCPM2b
 Chi2, Likelihood =  4.046E+00  4.796E-01
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-32   1.30E+01     0.0   2.889E-02   8.000E-02   3.612E-01   3.612E-01
 Signal Region:  SR2FULL_350
 Txnames:  THSCPM1b
 Chi2, Likelihood =  1.946E+00  1.508E+01
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-32   1.30E+01     0.0   3.822E-01   1.089E+00   3.511E-01   3.611E-01
 Signal Region:  (UL)
 Txnames:  THSCPM1b
--------------------------------------------------------------------------------
 CMS-PAS-EXO-16-036   1.30E+01     0.0   3.822E-01   1.212E+00   3.155E-01   N/A
 Signal Region:  (UL)
 Txnames:  THSCPM1b
[…]
```

```
Input status: 1
Decomposition output status: 1 #decomposition was successful
# Input File: slhaFiles/IDM_points/30000000110_paramcard.slha
[…]
# Database version: 2.0.0-beta
================================================================================
#Analysis   Sqrts   Cond_Violation   Theory_Value(fb)   Exp_limit(fb)   r   r_expected

 ATLAS-SUSY-2016-06   1.30E+01     0.0   6.591E-03   2.200E-01   2.996E-02   2.354E-02
 Signal Region:  SR_EW
 Txnames:  TDTM1S, TDTM2S
 Chi2, Likelihood =  4.698E-01  1.341E-02
--------------------------------------------------------------------------------
     CMS-EXO-13-006   8.00E+00     0.0   1.738E-08   1.700E-01   1.022E-07   7.898E-08
 Signal Region:  c200
 Txnames:  THSCPM1b, THSCPM2b
 Chi2, Likelihood =  1.120E+00  2.072E+00
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-32   1.30E+01     0.0   6.911E-09   1.000E-01   6.911E-08   6.911E-08
 Signal Region:  SR1FULL_600
 Txnames:  THSCPM1b, THSCPM2b
 Chi2, Likelihood =  8.113E-01  4.891E-01


================================================================================
The highest r value is = 0.029960804811
[…]
```

Parameter scan in Inert Doublet Model (IDM)

Change in parameters.ini: `model=share.models.idm` and ensure that `outputType` contains `python`

```
> ./runSModelS.py -f slhaFiles/IDM_points/ -p parameters.ini -o results/IDM_output/
```

```
Input status: 1
Decomposition output status: 1 #decomposition was successful
# Input File: slhaFiles/IDM_points/30000000105_paramcard.slha
[…]
# Database version: 2.0.0-beta
================================================================================
#Analysis   Sqrts   Cond_Violation   Theory_Value(fb)   Exp_limit(fb)   r   r_expected

     CMS-EXO-13-006   8.00E+00     0.0   7.783E-02   1.700E-01   4.578E-01   3.538E-01
 Signal Region:  c200
 Txnames:  THSCPM1b, THSCPM2b
 Chi2, Likelihood =  4.046E+00  4.796E-01
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-32   1.30E+01     0.0   2.889E-02   8.000E-02   3.612E-01   3.612E-01
 Signal Region:  SR2FULL_350
 Txnames:  THSCPM1b
 Chi2, Likelihood =  1.946E+00  1.508E+01
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-32   1.30E+01     0.0   3.822E-01   1.089E+00   3.511E-01   3.611E-01
 Signal Region:  (UL)
 Txnames:  THSCPM1b
--------------------------------------------------------------------------------
 CMS-PAS-EXO-16-036   1.30E+01     0.0   3.822E-01   1.212E+00   3.155E-01   N/A
 Signal Region:  (UL)
 Txnames:  THSCPM1b
[…]
```

HSCP searches, cτ= 22m

```
Input status: 1
Decomposition output status: 1 #decomposition was successful
# Input File: slhaFiles/IDM_points/30000000110_paramcard.slha
[…]
# Database version: 2.0.0-beta
================================================================================
#Analysis   Sqrts   Cond_Violation   Theory_Value(fb)   Exp_limit(fb)   r   r_expected

 ATLAS-SUSY-2016-06   1.30E+01     0.0   6.591E-03   2.200E-01   2.996E-02   2.354E-02
 Signal Region:  SR_EW
 Txnames:  TDTM1S, TDTM2S
 Chi2, Likelihood =  4.698E-01  1.341E-02
--------------------------------------------------------------------------------
     CMS-EXO-13-006   8.00E+00     0.0   1.738E-08   1.700E-01   1.022E-07   7.898E-08
 Signal Region:  c200
 Txnames:  THSCPM1b, THSCPM2b
 Chi2, Likelihood =  1.120E+00  2.072E+00
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-32   1.30E+01     0.0   6.911E-09   1.000E-01   6.911E-08   6.911E-08
 Signal Region:  SR1FULL_600
 Txnames:  THSCPM1b, THSCPM2b
 Chi2, Likelihood =  8.113E-01  4.891E-01


================================================================================
The highest r value is = 0.029960804811
[…]
```

Parameter scan in Inert Doublet Model (IDM)

Change in parameters.ini: `model=share.models.idm` and ensure that `outputType` contains `python`

```
> ./runSModelS.py -f slhaFiles/IDM_points/ -p parameters.ini -o results/IDM_output/
```

```
Input status: 1
Decomposition output status: 1 #decomposition was successful
# Input File: slhaFiles/IDM_points/30000000105_paramcard.slha
[…]
# Database version: 2.0.0-beta
================================================================================
#Analysis   Sqrts   Cond_Violation   Theory_Value(fb)   Exp_limit(fb)   r   r_expected

     CMS-EXO-13-006   8.00E+00     0.0   7.783E-02   1.700E-01   4.578E-01   3.538E-01
 Signal Region:  c200
 Txnames:  THSCPM1b, THSCPM2b
 Chi2, Likelihood =  4.046E+00  4.796E-01
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-32   1.30E+01     0.0   2.889E-02   8.000E-02   3.612E-01   3.612E-01
 Signal Region:  SR2FULL_350
 Txnames:  THSCPM1b
 Chi2, Likelihood =  1.946E+00  1.508E+01
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-32   1.30E+01     0.0   3.822E-01   1.089E+00   3.511E-01   3.611E-01
 Signal Region:  (UL)
 Txnames:  THSCPM1b
--------------------------------------------------------------------------------
 CMS-PAS-EXO-16-036   1.30E+01     0.0   3.822E-01   1.212E+00   3.155E-01   N/A
 Signal Region:  (UL)
 Txnames:  THSCPM1b
[…]
```

```
Input status: 1
Decomposition output status: 1 #decomposition was successful
# Input File: slhaFiles/IDM_points/30000000110_paramcard.slha
[…]
# Database version: 2.0.0-beta
================================================================================
#Analysis   Sqrts   Cond_Violation   Theory_Value(fb)   Exp_limit(fb)   r   r_expected

 ATLAS-SUSY-2016-06   1.30E+01     0.0   6.591E-03   2.200E-01   2.996E-02   2.354E-02
 Signal Region:  SR_EW
 Txnames:  TDTM1S, TDTM2S
 Chi2, Likelihood =  4.698E-01  1.341E-02
--------------------------------------------------------------------------------
     CMS-EXO-13-006   8.00E+00     0.0   1.738E-08   1.700E-01   1.022E-07   7.898E-08
 Signal Region:  c200
 Txnames:  THSCPM1b, THSCPM2b
 Chi2, Likelihood =  1.120E+00  2.072E+00
--------------------------------------------------------------------------------
 ATLAS-SUSY-2016-32   1.30E+01     0.0   6.911E-09   1.000E-01   6.911E-08   6.911E-08
 Signal Region:  SR1FULL_600
 Txnames:  THSCPM1b, THSCPM2b
 Chi2, Likelihood =  8.113E-01  4.891E-01


================================================================================
The highest r value is = 0.029960804811
[…]
```

Disappearing track, cτ=38cm

Parameter scan IDM - plotting

Either use python script: `plotting/plotIDMResults.py`
or run jupyter notebook:

```
> jupyter notebook plotting/plotIDMResults.ipynb
```

Plot IDM results

In [1]:

```
import warnings
warnings.filterwarnings("ignore", message="numpy.dtype size changed")
import pyslha
import matplotlib.pyplot as plt
import numpy as np
import imp,glob,os
cm = plt.cm.get_cmap('RdYlBu')
```

Define path to SLHA and results folders

In [2]:

```
slhaFolder = '../slhaFiles/IDM_points'
resultsFolder = '../results/IDM_output'
```

Parameter scan IDM - plotting (cont.)

Read SModelS results:

In [3]:

```
#Convert Experimental Results list to a dictionary
data = []
for f in glob.glob(resultsFolder+'/*.py'):
    smodelsDict = imp.load_source(f.replace('.py',''),f).smodelsOutput
    slhaFile = os.path.basename(smodelsDict['OutputStatus']['input file'])
    slhaFile = os.path.join(slhaFolder,slhaFile)
    #Read SLHA file (using pyslha):
    slhaData = pyslha.readSLHAFile(slhaFile)
    data.append((slhaData,smodelsDict))
```

In [5]:

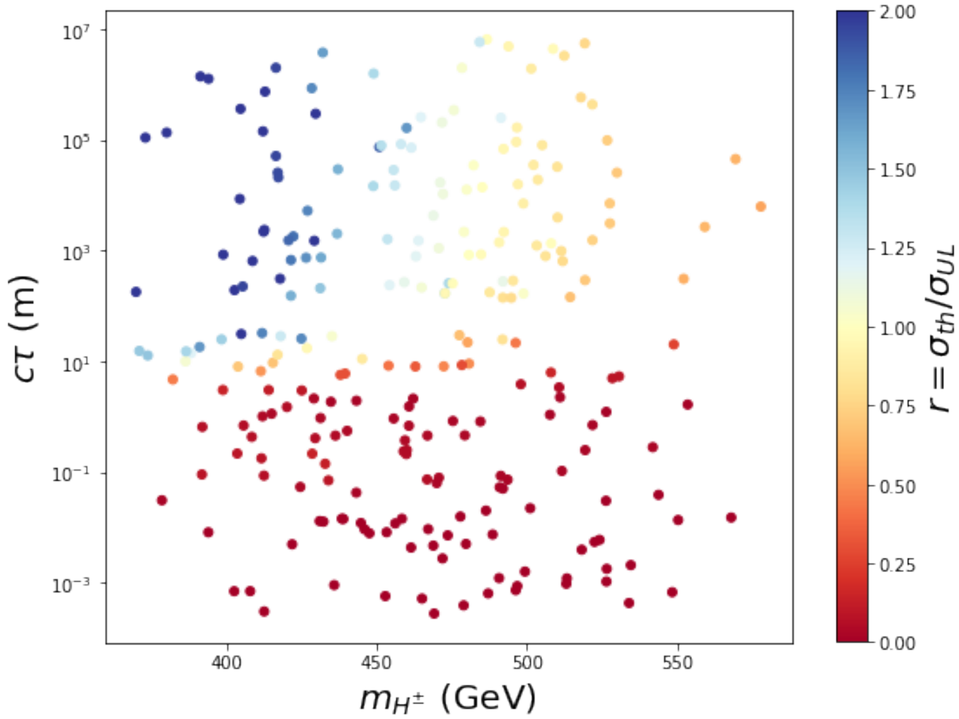
```
mHc = []
ctau = []
r = []
for slhaData,smodelsDict in data:
    if not 'ExptRes' in smodelsDict: #If no results were applicable, point is all
        rmax = 0.0
    else:
        rmax = smodelsDict['ExptRes'][0]['r'] #First result is the most constrain
    mass = slhaData.blocks['MASS'][37] #H+ mass
    width = slhaData.decays[37].totalwidth #H+ mass
    if width:
        ct = 1.967e-16/width
    else:
        ct = 1e7

    mHc.append(mass)
    ctau.append(ct)
    r.append(rmax)
```

Parameter scan IDM - plotting (cont.)

In [6]:

```
plt.figure(figsize=(8,6))
cp = plt.scatter(mHc,ctau, c=r, vmin=0., vmax=2., s=25, cmap=cm)
cb = plt.colorbar(cp)
plt.yscale('log')
plt.xlabel(r'$m_{H^{\pm}}$ (GeV)',fontsize=20)
plt.ylabel(r'$c\tau$ (m)',fontsize=20)
cb.set_label(r'$r=\sigma_{th}/\sigma_{UL}$',fontsize=20)
plt.tight_layout()
plt.show()
```



Including other models — the particles module

- SModelS includes predefined model files for the MSSM, NMSSM, MDGSSM and the IDM (stored in `smodels/share/models/`).

- To write a new model, one can either (a) employ the particle module, defining the BSM states by their quantum numbers

- or (b) define the quantum numbers via the QNUMBERS block directly in the slha file. They are be passed to SModelS by setting:
`model=path/to/your/slha_file.slha` in parameters.ini

- Another convenient way to automatically generate the model file is to use the micrOMEGAs interface. An updated version compatible with SModelS 2.0 will be released soon!

a. python module:

idm.py

```python
from smodels.theory.particle import Particle, MultiParticle

H0 = Particle(Z2parity=-1, label='H0', pdg=35, eCharge=0, colordim=1, spin=0)
A0 = Particle(Z2parity=-1, label='A0', pdg=36, eCharge=0, colordim=1, spin=0)
H = Particle(Z2parity=-1, label='H+', pdg=37, eCharge=+1, colordim=1, spin=0)
```

b. SLHA file with QNUMBERs:

```
Block QNUMBERS 35  # h2
        1 0  # 3 times electric charge
        2 1  # number of spin states (2S+1)
        3 1  # colour rep (1: singlet, 3: triplet, 8: octet)
        4 0  # Particle/Antiparticle distinction (0=own anti)
Block QNUMBERS 36  # h3
        1 0  # 3 times electric charge
        2 1  # number of spin states (2S+1)
        3 1  # colour rep (1: singlet, 3: triplet, 8: octet)
        4 0  # Particle/Antiparticle distinction (0=own anti)
Block QNUMBERS 37  # h+
        1 3  # 3 times electric charge
        2 1  # number of spin states (2S+1)
        3 1  # colour rep (1: singlet, 3: triplet, 8: octet)
        4 1  # Particle/Antiparticle distinction (0=own anti)
```

idm_example.slha

SModelS as a python library — your own code

- Users more familiar with Python and the SModelS language may prefer to write their own main program. A simple example code for this purpose is provided in Example.py on the top-level directory.

```
from smodels import particlesLoader
from smodels.theory import slhaDecomposer,lheDecomposer
from smodels.tools.physicsUnits import fb, GeV, TeV
from smodels.theory.theoryPrediction import theoryPredictionsFor
from smodels.experiment.databaseObj import Database
from smodels.tools import coverage
from smodels.tools.smodelsLogging import setLogLevel
setLogLevel("info")

[…]
```

- Step-by-step explanations: RunningSModelS.html#example-py

- Online tutorial from PyHEP2020 conference: jupyter notebook and youtube video !

SModelS Tools

`smodelsTools.py` provides a number of convenient applications for the user:

- **`xseccomputer`** : a cross section calculator (for MSSM particles) based on Pythia and NLLfast,
- **`slhachecker`** and **`lhechecker`** : to check SLHA or LHE input files for completeness and sanity,
- **`database-browser`** : for easy direct access to the database of experimental results,
- **`interactive-plots`** : a plotting tool to make interactive plots based on plotly.

Other useful functionalities provided through smodelsTools.py:

- **`fixpermissions`** : to fix a problem with file permissions for the cross section computers in system-wide installs,
- **`toolbox`** : to quickly show the state of the external tools.

https://smodels.readthedocs.io/en/latest/SModelSTools.html

When you use SModelS, please cite:

- ***A SModelS interface for pyhf likelihoods***,
  Gaël Alguero, Sabine Kraml, Wolfgang Waltenberger, arXiv:2009.01809
- ***SModelS database update v1.2.3***,
  Charanjit K. Khosa, Sabine Kraml, Andre Lessa, Philipp Neuhuber, Wolfgang Waltenberger, arXiv:2005.00555, LHEP 158 2020
- ***SModelS v1.2: long-lived particles, combination of signal regions, and other novelties***,
  Federico Ambrogi et al., arXiv:1811.10624, CPC 251 (2020) 106848
- ***Constraining new physics with searches for long-lived particles: Implementation into SModelS***,
  Jan Heisig, Sabine Kraml, Andre Lessa, arXiv:1808.05229, Phys.Lett. B788 (2019) 87-95.
- ***SModelS extension with the CMS supersymmetry search results from Run 2***,
  Juhi Dutta, Sabine Kraml, Andre Lessa, Wolfgang Waltenberger, arXiv:1803.02204, LHEP 1 (2018) no.1,5-12
- ***SModelS v1.1 user manual: improving simplified model constraints with efficiency maps***,
  Federico Ambrogi, Sabine Kraml, Suchita Kulkarni, Ursula Laa, Andre Lessa, Veronika Magerl, Jory Sonneveld, Michael Traub, Wolfgang Waltenberger, arXiv:1701.06586, CPC 227 (2018) 72-98
- ***SModelS: a tool for interpreting simplified-model results from the LHC and its application to supersymmetry***,
  Sabine Kraml, Suchita Kulkarni, Ursula Laa, Andre Lessa, Wolfgang Magerl, Doris Proschofsky, Wolfgang Waltenberger, arXiv: 1312.4175, EPJC (2014) 74:2868