Hands-On SModelS: a Tool for Making Systematic Use of Simplified Models Results

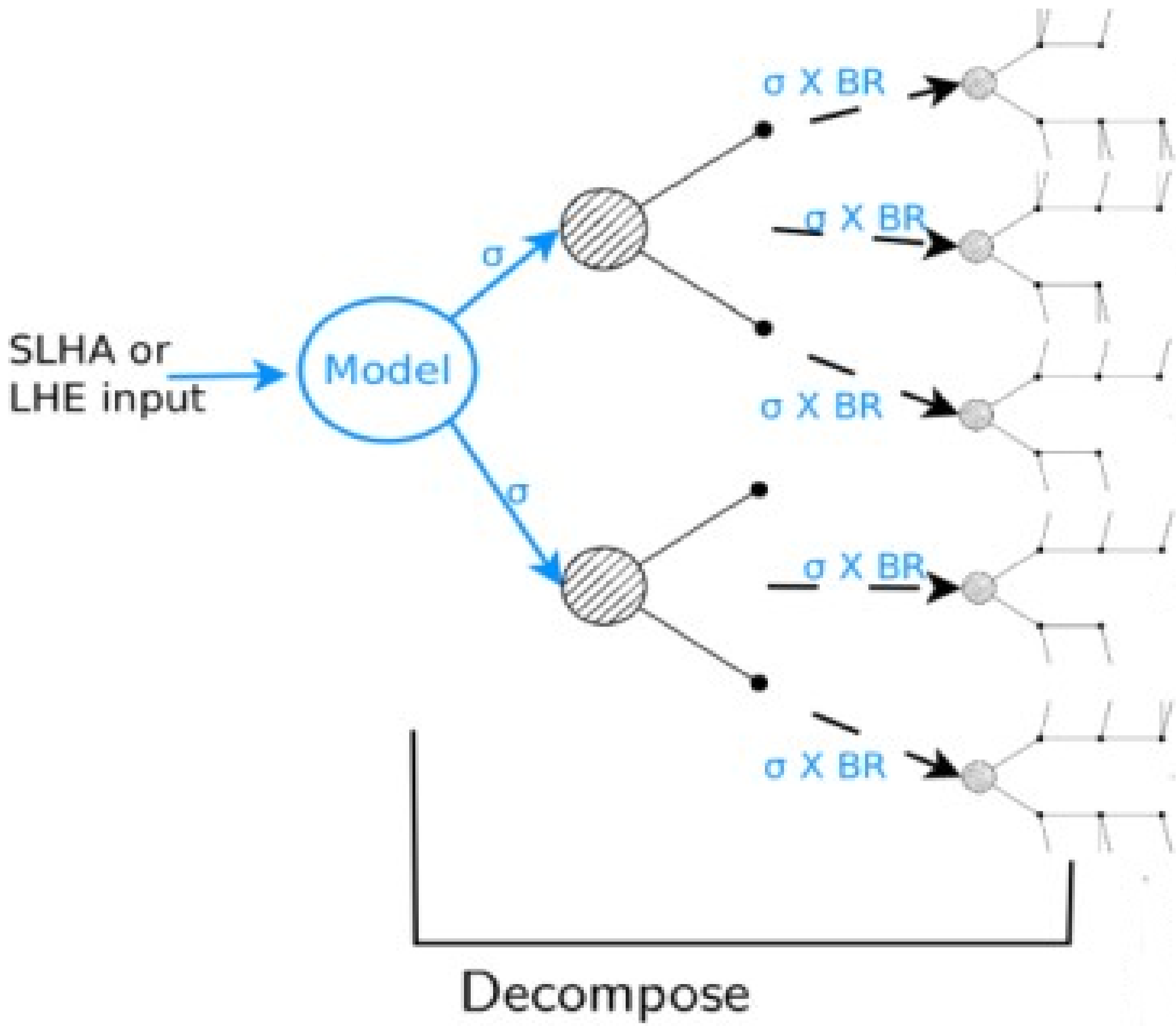
Wolfgang Waltenberger
HEPHY Vienna

for the SModelS group
F. Ambrogi, S. Kraml, S. Kulkarni, U. Laa, A. Lessa, J. Sonneveld

Reinterpretation Workshop
Fermilab, Chicago, USA, 16 – 18 October 2017

How SModelS works:

1) Decomposition of a fundamental model



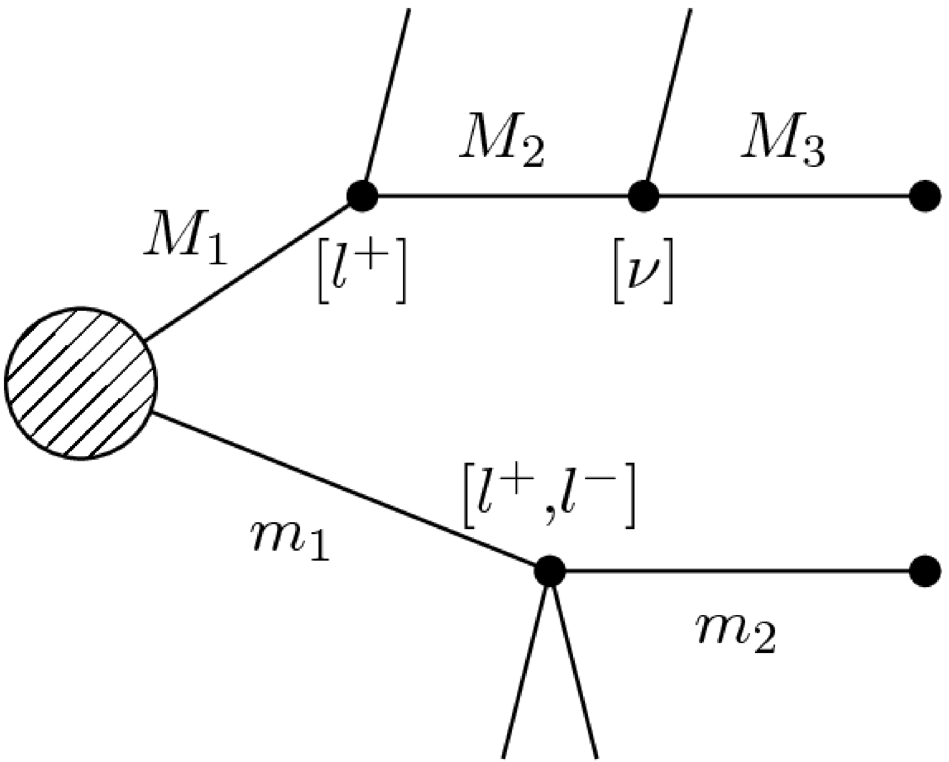
Input: SLHA file (mass spectrum, BRs) or LHE file (parton level)

Currently the model must have a $\mathbf{Z}_2$ symmetry

The decomposition produces a set of simplified model topologies (dubbed “elements”)

How SModelS works:

Simplified Model Topology:



$$= [[l^+],[\nu]]$$

$$= [\ [[l^+],[\nu]]\ ,\ [[l^+,l^-]]\ ]$$
$$([[M_1,M_2,M_3],[m_1,m_2]])$$

$$= [[l^+,l^-]]$$

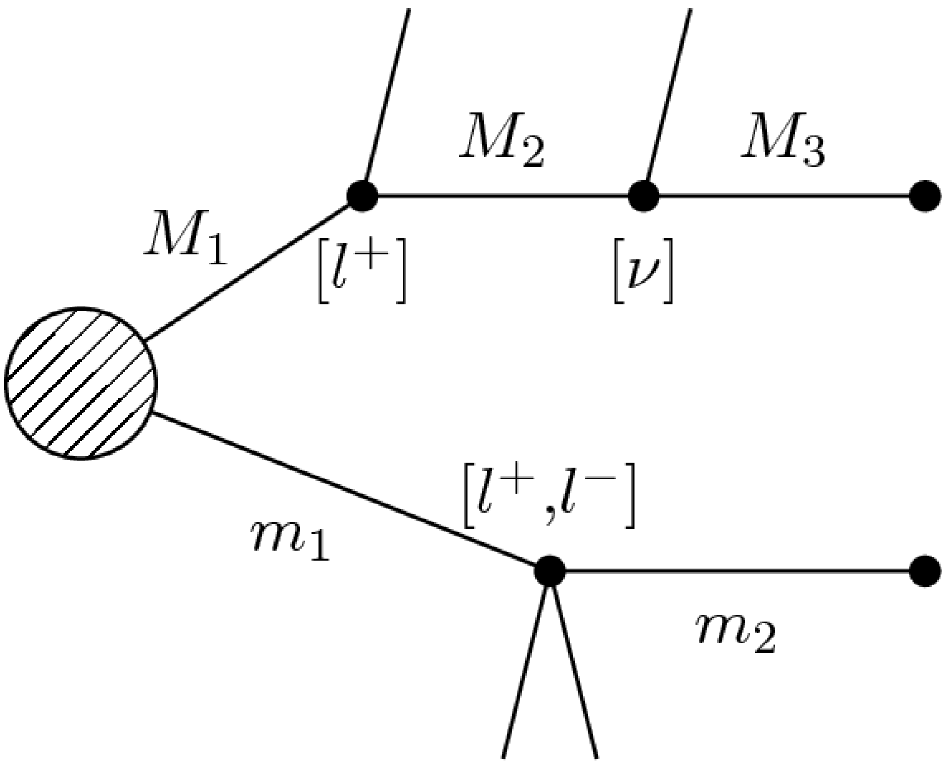
Each topology is described by:
- Topology shape + final states
- BSM masses
- σ x BR

We (currently) ignore spin, color, etc of the BSM particles

It is model independent, there is no reference to the original model

How SModelS works:

Simplified Model Topology:



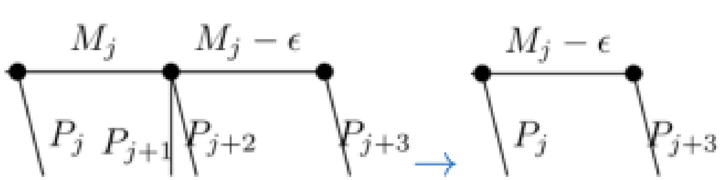
$$= [[l^+],[\nu]]$$

$$= [[l^+, l^-]]$$

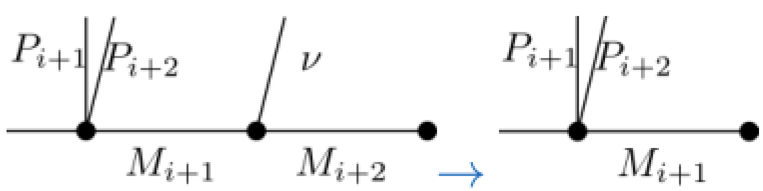
$$= [\ [[l^+],[\nu]]\ ,\ [[l^+, l^-]]\ ]$$

$$([[M_1, M_2, M_3],[m_1, m_2]])$$

Soft particles are ommitted:



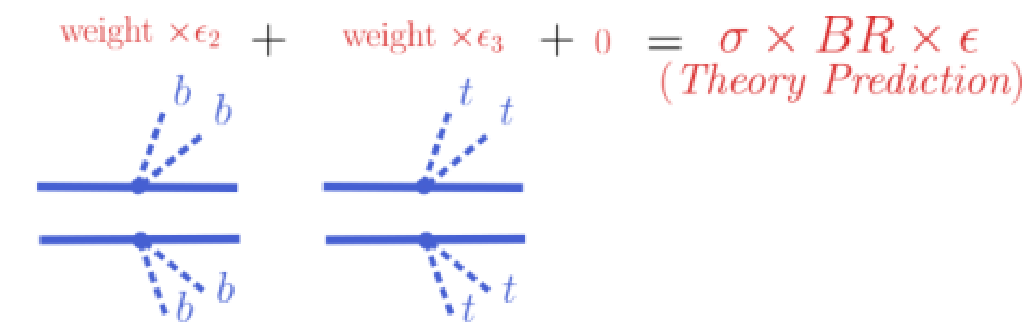
Invisible final states are grouped into effective LSPs:

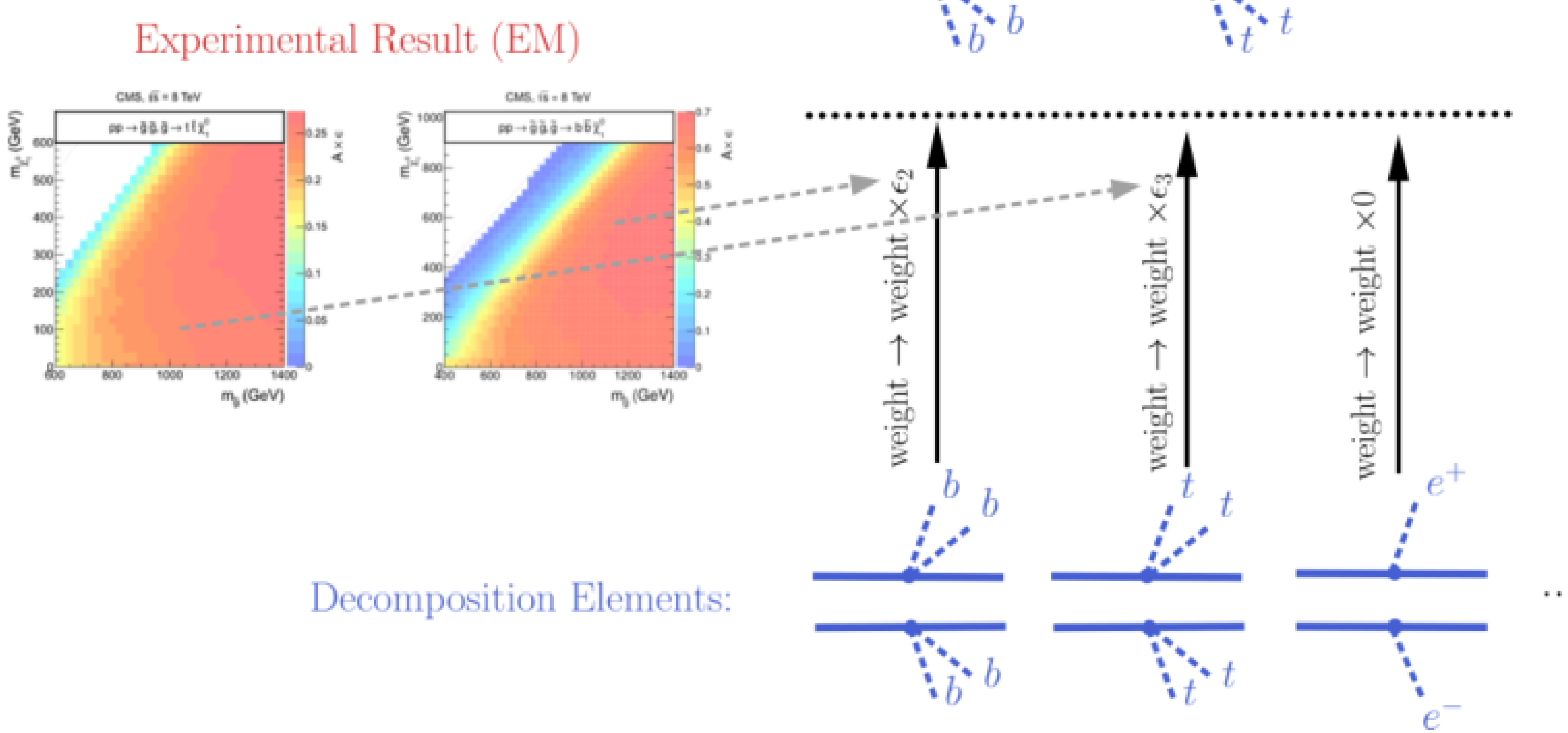


How SModelS works:

2) Computation of predicted signal strength:

For **efficiency map results** we have signal efficiencies for various “elements”, and we can add them together:

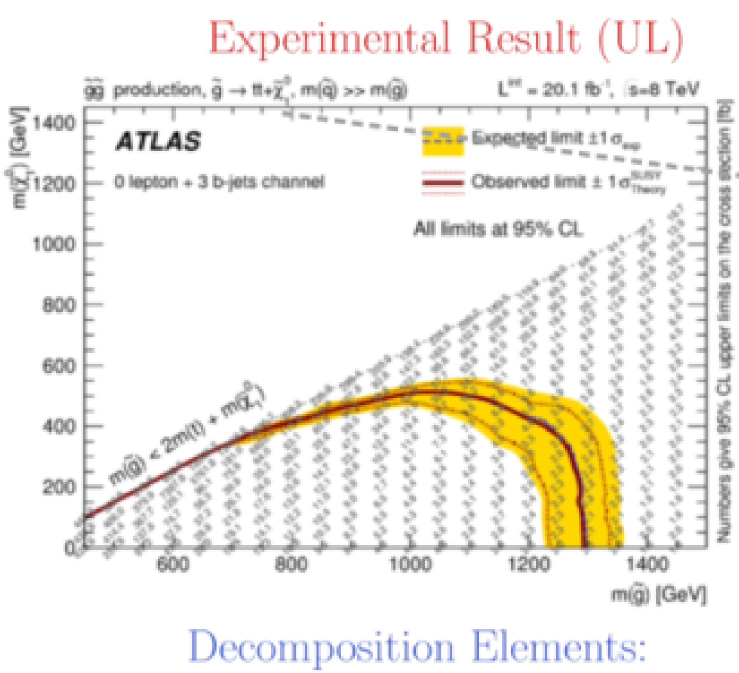


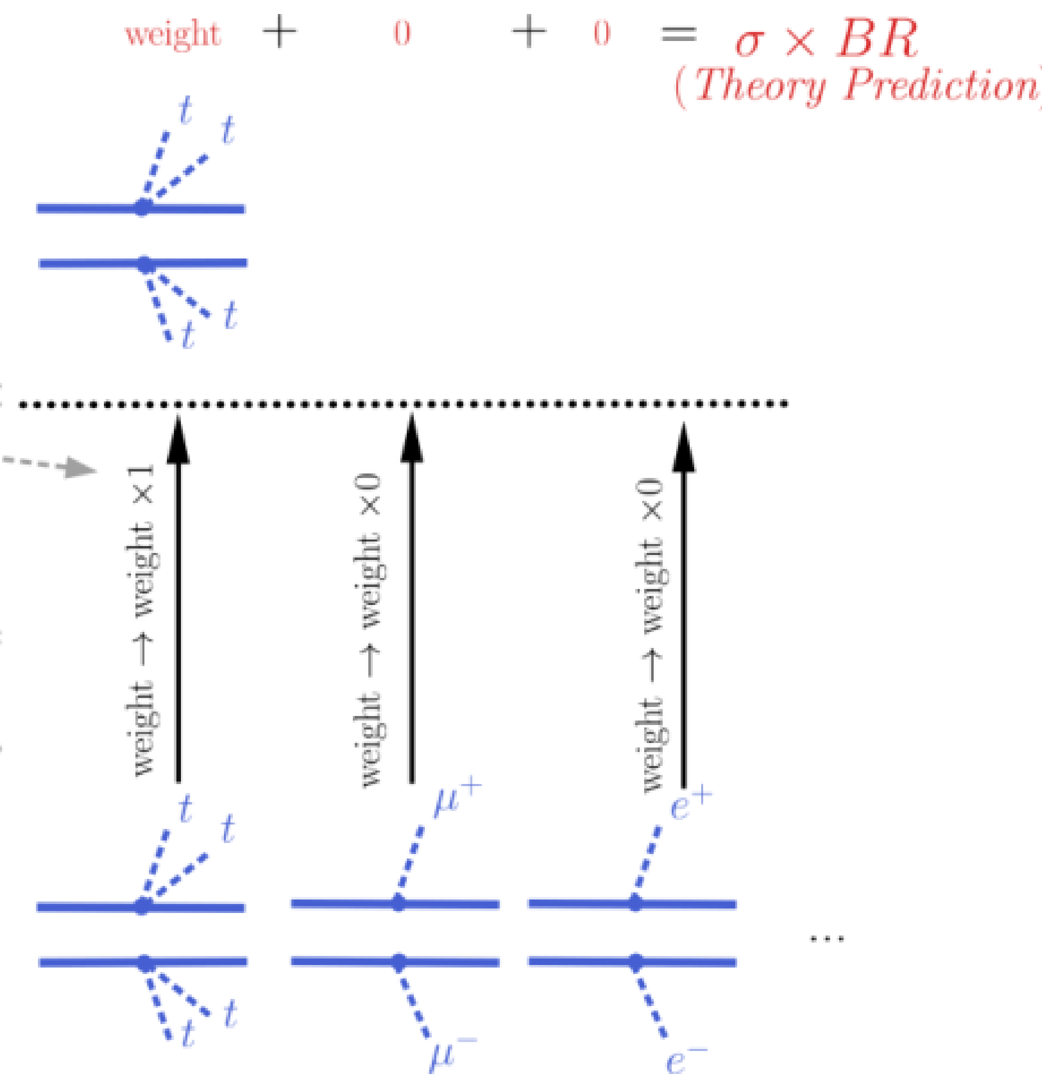


How SModelS works:

2) Computation of predicted signal strength:

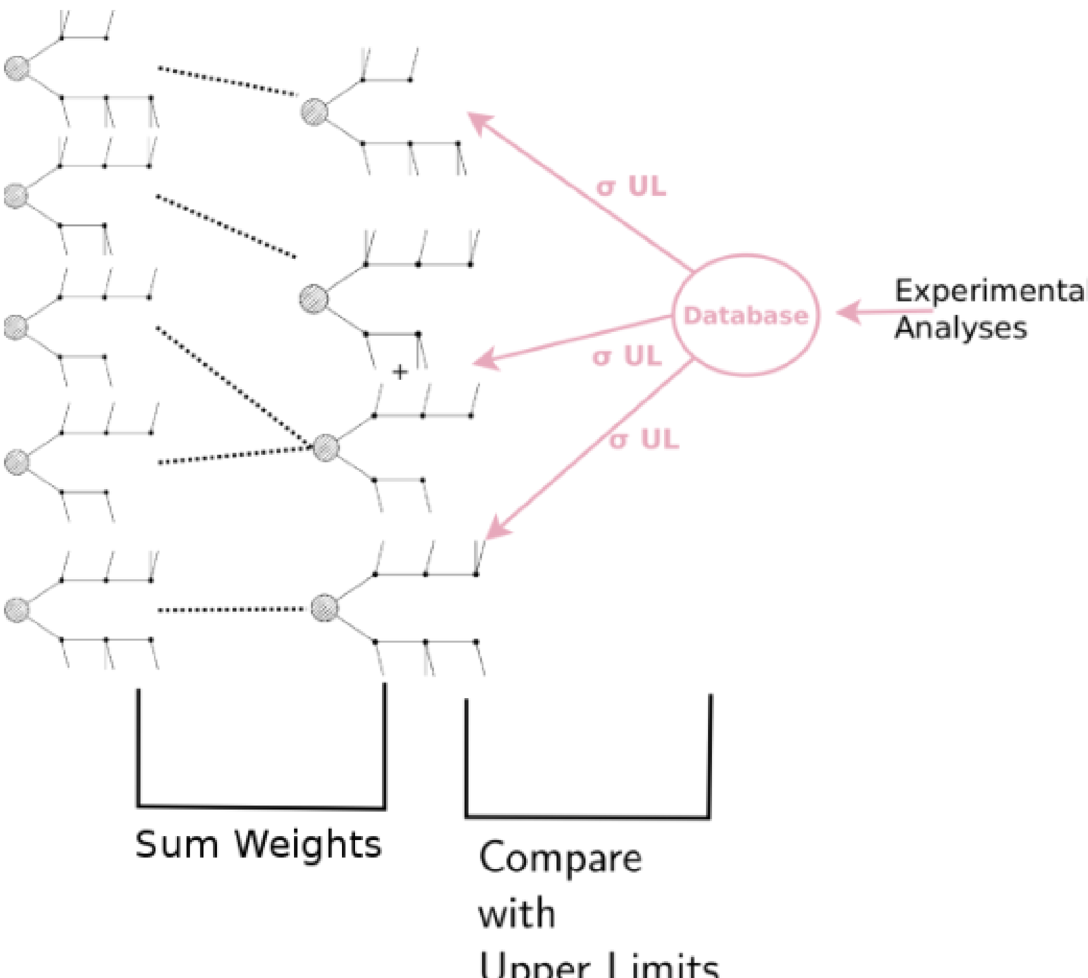
Upper limit results we cannot add up:





How SModelS works:

3) Comparison of predicted signal strengths with experimental result:



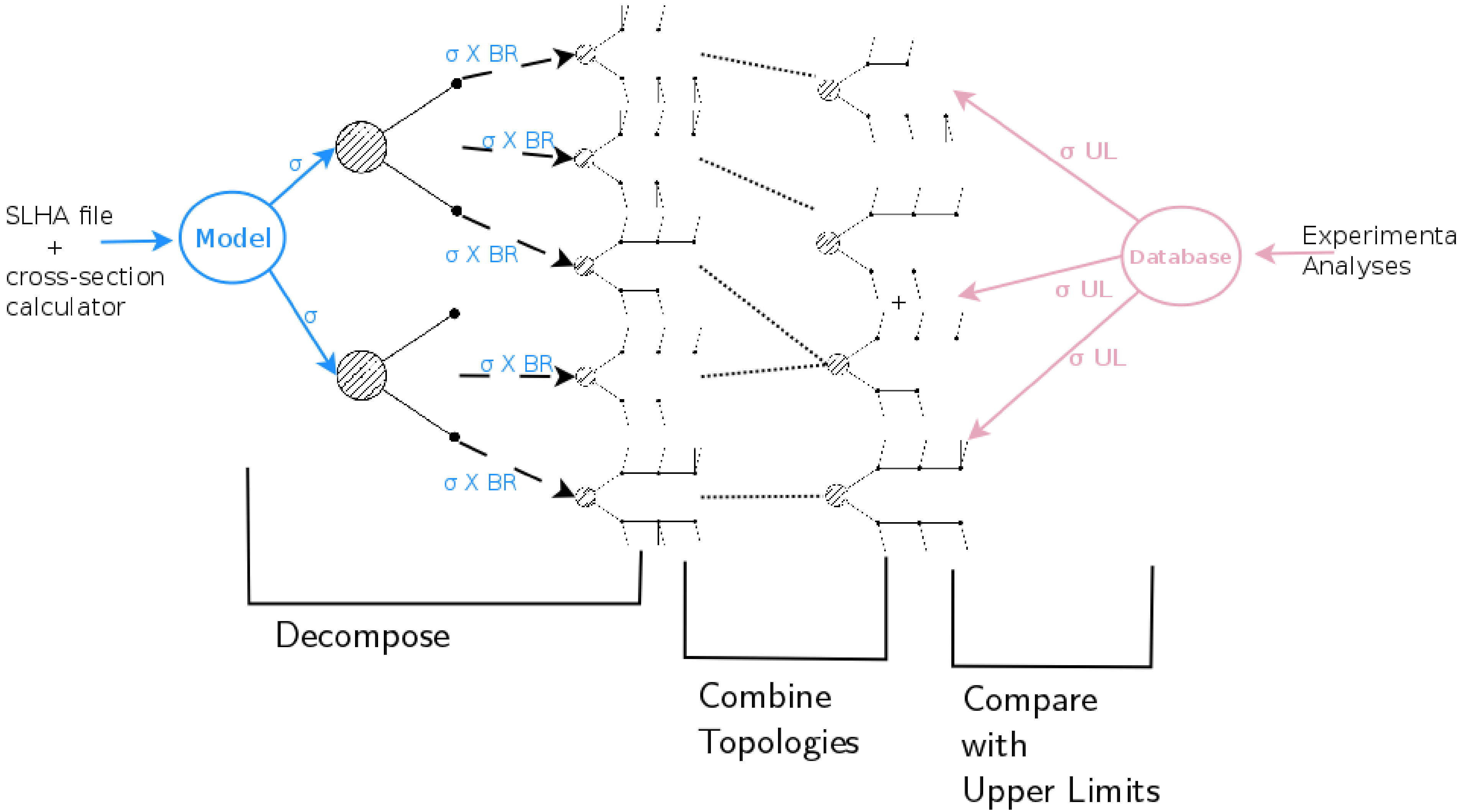
- **Upper Limit Results:**
  Predicted signal strength = σ x BR
  Experimental result: σ_{UL}

- **Efficiency Map Results:**
  Predicted signal strength = ∑ σ x BR x ε
  Experimental result: $\sigma_{UL}=N_{UL}/L$ from $N_{observed}$, expected(BG), error(BG)

- r = predicted / σ_{UL}

- Model is excluded if most constraining analysis has r > 1

How SModelS works:



What's in the database?

ID	Topologies	Type	$\mathcal{L}$ [fb^{-1}]	$\sqrt{s}$
CMS-SUS-PAS-15-002	3: T1, T1bbbb, T1tttt[off]	ul	2.2	13
CMS-PAS-SUS-13-015	1: T2tt[off]	ul	19.4	8
CMS-PAS-SUS-13-015	1: T2tt[off]	eff	19.4	8
CMS-PAS-SUS-13-016	1: T1tttt[off]	ul	19.7	8
CMS-PAS-SUS-13-016	1: T1tttt[off]	eff	19.7	8
CMS-PAS-SUS-13-018	1: T2bb	ul	19.4	8
CMS-PAS-SUS-13-023	2: T2tt[off], T6bbWW[off]	ul	18.9	8
CMS-PAS-SUS-14-011	3: T1bbbb, T1tttt[off], T2tt[off]...	ul	19.3	8
CMS-SUS-12-024	2: T1bbbb, T1tttt[off]	ul	19.4	8
CMS-SUS-12-024	2: T1bbbb, T1tttt[off]	eff	19.4	8
CMS-SUS-12-028	6: T1, T1bbbb, T1tttt, T2, T2bb...	ul	11.7	8
CMS-SUS-13-002	1: T1tttt	ul	19.5	8
CMS-SUS-13-004	3: T1bbbb, T1tttt[off], T2tt[off]...	ul	19.3	8
CMS-SUS-13-006	5: TChiChipmSlepL, TChiChipmSlepStau...	ul	19.5	8
CMS-SUS-13-006	1: TChiWH	eff	19.5	8
CMS-SUS-13-007	2: T1tttt[off], T5tttt[off]	ul	19.3	8
CMS-SUS-13-007	1: T1tttt[off]	eff	19.3	8
CMS-SUS-13-011	2: T2tt[off], T6bbWW[off]	ul	19.5	8
CMS-SUS-13-011	1: T2tt[off]	eff	19.5	8
CMS-SUS-13-012	3: T1, T1tttt[off], T2	ul	19.5	8
CMS-SUS-13-013	2: T1tttt[off], T6ttWW[off]	ul	19.5	8
CMS-SUS-13-013	1: T1tttt[off]	eff	19.5	8
CMS-SUS-13-019	6: T1, T1bbbb, T1tttt[off], T2, T2bb...	ul	19.5	8
CMS-SUS-14-010	1: T1tttt[off]	ul	19.5	8
CMS-SUS-14-021	1: T2bbWW[off]	ul	19.7	8
CMS-SUS-14-021	1: T2bbWW[off]	eff	19.7	8

Table 1: SModelS database (CMS)

ID	Topologies	Type	$\mathcal{L}$ [fb^{-1}]	$\sqrt{s}$
ATLAS-SUSY-2015-09	1: T1tttt	ul	3.2	13
ATLAS-CONF-2013-024	1: T2tt	ul	20.5	8
ATLAS-CONF-2013-024	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.5	8
ATLAS-CONF-2013-024	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	1.0	8
ATLAS-CONF-2013-024	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	1.0	8
ATLAS-CONF-2013-035	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.7	8
ATLAS-CONF-2013-036	1: TChiChiSlepSlep	ul	20.7	8
ATLAS-CONF-2013-037	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.7	8
ATLAS-CONF-2013-047	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.3	8
ATLAS-CONF-2013-048	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.3	8
ATLAS-CONF-2013-049	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.3	8
ATLAS-CONF-2013-053	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.1	8
ATLAS-CONF-2013-053	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	1.0	8
ATLAS-CONF-2013-054	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.3	8
ATLAS-CONF-2013-061	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.1	8
ATLAS-CONF-2013-062	1: T5WW[off]	ul	20.3	8
ATLAS-CONF-2013-062	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.3	8
ATLAS-CONF-2013-089	2: T5WW[off], T6WW[off]	ul	20.3	8
ATLAS-CONF-2013-093	1: TChiWH	ul	20.3	8
ATLAS-CONF-2013-093	24: T1, T1bbbb, T1bbbt, T1bbqq, T1bbtt...	eff	20.3	8
ATLAS-SUSY-2013-02	6: T1, T2, T5WW[off], T5tctc, T6WW[off]...	ul	20.3	8
ATLAS-SUSY-2013-02	2: T1, T2	eff	20.3	8
ATLAS-SUSY-2013-04	1: T1tttt	ul	20.3	8
ATLAS-SUSY-2013-04	3: T1tttt, T5WW[off], T5ZZ[off]...	eff	20.3	8
ATLAS-SUSY-2013-05	2: T2bb, T6bbWW[off]	ul	20.1	8
ATLAS-SUSY-2013-05	1: T2bb	eff	20.1	8
ATLAS-SUSY-2013-08	1: T6ZZtt	ul	20.3	8
ATLAS-SUSY-2013-09	1: T1tttt	ul	20.3	8
ATLAS-SUSY-2013-09	1: T1tttt	eff	20.3	8
ATLAS-SUSY-2013-11	4: TChiWW, TChiWZ, TChipChimSlepSnu...	ul	20.3	8
ATLAS-SUSY-2013-11	1: TSlepSlep	eff	20.3	8
ATLAS-SUSY-2013-12	4: TChiChipmSlepL, TChiChipmStauL...	ul	20.3	8
ATLAS-SUSY-2013-14	2: TChiChipmStauL, TChipChimStauSnu...	ul	20.3	8
ATLAS-SUSY-2013-15	3: T2bbWW, T2tt, T6bbWW	ul	20.3	8
ATLAS-SUSY-2013-15	1: T2tt	eff	20.3	8
ATLAS-SUSY-2013-16	1: T2tt	ul	20.1	8
ATLAS-SUSY-2013-16	2: T2tt, T6bbWW[off]	eff	20.1	8
ATLAS-SUSY-2013-18	2: T1bbbb, T1tttt	ul	20.1	8
ATLAS-SUSY-2013-18	2: T1bbbb, T1tttt	eff	20.1	8
ATLAS-SUSY-2013-19	3: T2bbWW, T2tt, T6bbWW[off]	ul	20.3	8
ATLAS-SUSY-2013-21	3: T2bb, T2bbWW[off], T2cc	eff	20.3	8
ATLAS-SUSY-2013-23	1: TChiWH	ul	20.3	8
ATLAS-SUSY-2014-03	1: TScharm	eff	20.3	8

We collect the results of the experimental collaborations, and augment them with recast analyses (MadAnalysis5, CheckMATE), creating our own efficiency maps.
SModelS v1.1.1 ships with ~ 70 analyses, and close to 200 results.
http://smodels.hephy.at/wiki/ListOfAnalysesv111

Availability

SModelS is written almost entirely in python and is available here:

http://smodels.hephy.at
http://github.com/SModelS/smodels

It uses pythia and nllfast for the computation of the cross sections.
An online manual is available here:
http://smodels.readthedocs.io/en/latest/

Try:

pip install –user smodels

(Is $PYTHONPATH set up appropriately?)

Tutorial

1) Compute cross sections

2) Test if any analysis excludes model

3) List topologies with largest cross sections that are not covered by any analysis

4) Compute χ^2 values and likelihoods

5) Usage of SModelS as a python library

0) download slha file, database, config file

- In this tutorial we will be working with a small, specially prepared database:
- wget http://smodels.hephy.at/walten/SModelSTutorial.tar.gz
  or download from workshop webpage
- tar xzvf SModelSTutorial.tar.gz
- cd SModelSTutorial
- You will find a sample slha file, a configuration file,

  and a small database for SModelS.

1) Compute cross sections

- Use smodelsTools.py to compute cross sections (at various perturbation orders):

```
smodelsTools.py xseccomputer -6 -s 8 13 -p -N -f chicago.slha
```

(depending on how you installed SModelS, this might trigger compiling pythia)

1) Compute cross sections

- Use smodelsTools.py to compute cross sections (at various perturbation orders):

```
smodelsTools.py xseccomputer -6 -s 8 13 -p -N -f chicago.slha
```

use pythia6 for computing LO xsecs (alternative is pythia8)

compute up to NLO+NLL order (if possible)

compute for sqrt(s)=8 and 13 TeV

write result into slha file (but only highest perturbation orders)

2) Quickly check if SModelS excludes a model

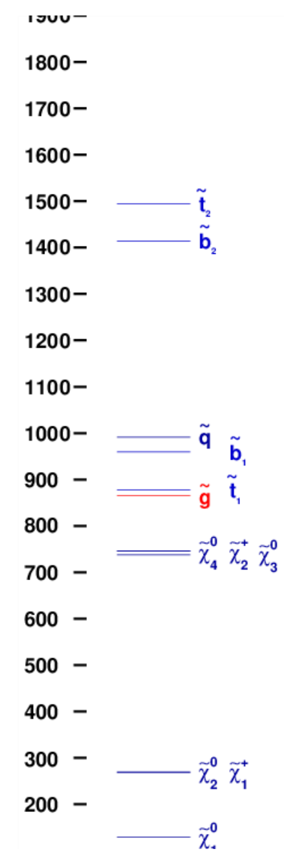
- Check out parameters.ini. Edit it.

  Specify “path = ./smodels-database-chicago” as the database path (in “[database]”). Have a look at the “printers”.

- Run “runSModelS.py”:

```
runSModelS.py -p parameters.ini -f chicago.slha
```

- Look at the summary in results/chicago.slha.smodels. Is the point excluded? If yes, by which analysis? If no, which analysis comes closest to excluding it?



chicago.slha

3) List topologies with largest cross sections that are not covered by any analysis

- Edit parameters.ini, set [options] testCoverage = True
- Look at the output given at stdout.
- What is the final state with the largest “missing” (“uncovered”) cross section?

4) Compute χ^2 values and likelihoods

- For efficiency maps, SModelS computes a likelihood, considering nobs to be the realization of a Poissonian variable, with the systematic error on the background estimate θ modelled as a Gaussian:

$$\mathcal{L}(\mu, \theta | D) = \frac{(\mu + b + \theta)^{n_{obs}} e^{\mu+b+\theta}}{n_{obs}!} exp\left(-\frac{\theta^2}{2\delta^2}\right)$$

- We may then define a test statistic T (= χ^2), from a likelihood ratio test, with H_0 being the signal hypothesis:

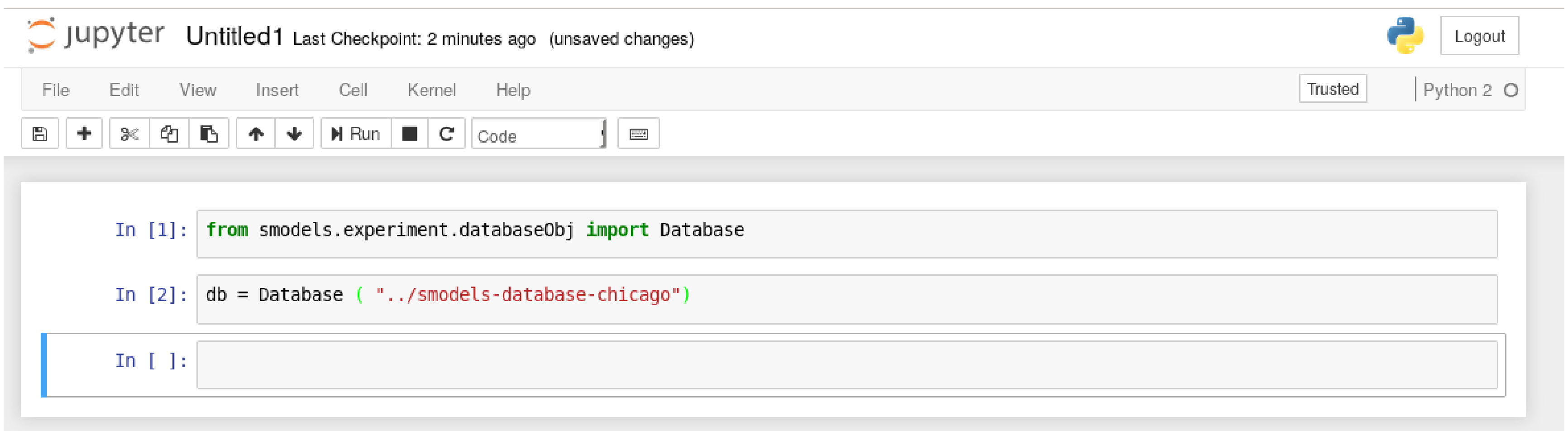
$$T = -2 \ln \frac{H_0}{H_1} = -2 \ln \left(\frac{\mathcal{L}(\mu = n_{\text{signal}}, \theta | D)}{sup\{\mathcal{L}(\mu, \theta | D) : \mu \in \mathbb{R}^+\}} \right)$$

4) Compute χ^2 values and likelihoods

- Currently likelihoods are computed only for efficiency-map type results.
- Edit parameters.ini, set “[options] computeStatistics=True”.
- What’s the χ^2 for ATLAS-SUSY-2013-18:T1tttt?
- Is the result compatible with the signal hypothesis?

5) Usage of SModelS as a python library

- Start ipython or jupyter



- How many simplified models results are in the database? What are the names of the analyses, which topologies do they cover?
(N.B. for most objects, the __str__ is defined sensibly, try printing the various objects)

5) Usage of SModelS as a python library

jupyter Untitled1 Last Checkpoint: 25 minutes ago (unsaved changes)

```
In [2]: from smodels.theory import slhaDecomposer
```

```
In [3]: listOfTopologies = slhaDecomposer.decompose ( "chicago.slha")
```

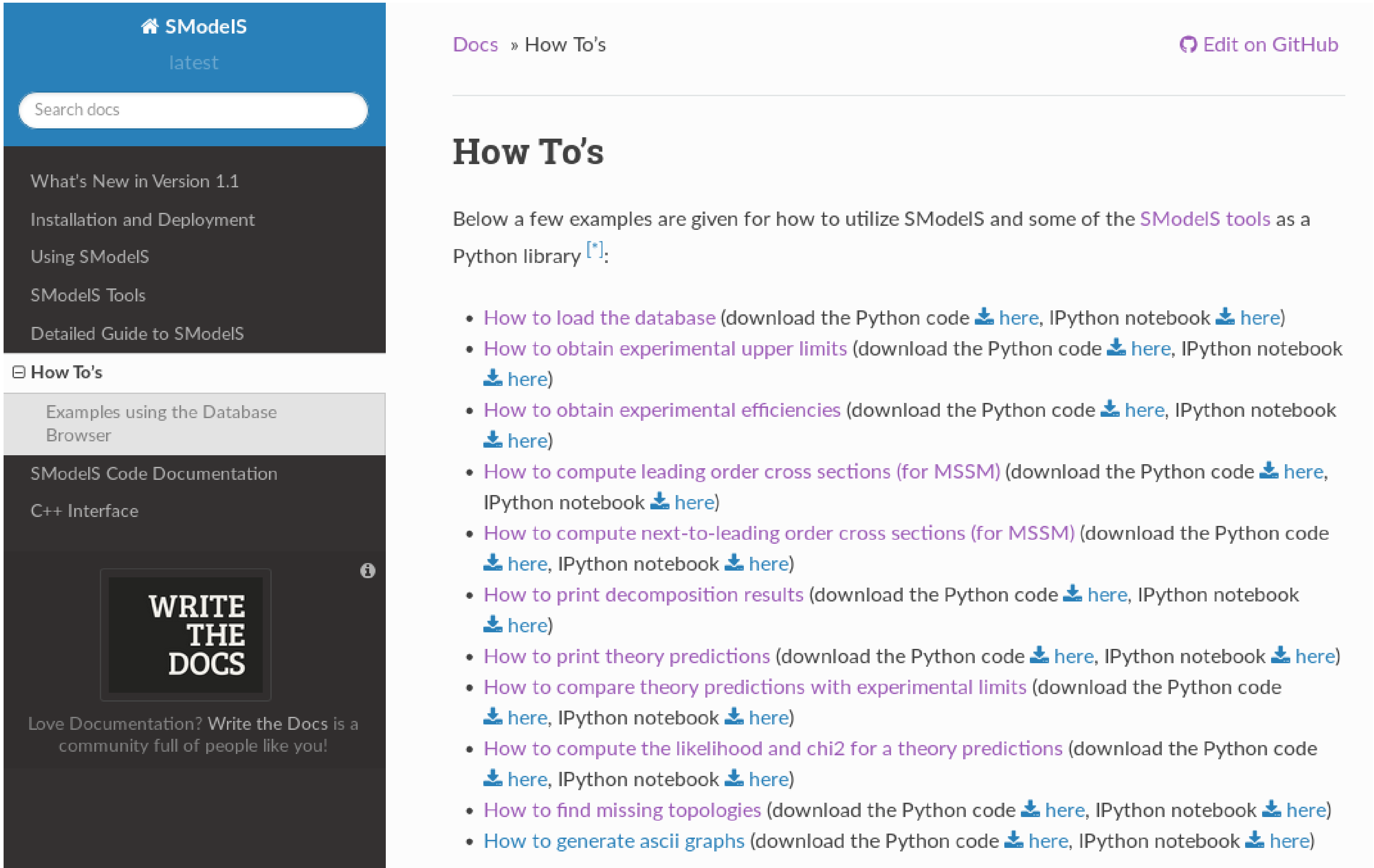
```
In [4]: for topo in listOfTopologies[:3]:
            for element in topo.elementList:
                print (element)

[[],[]]
[[],[[W+]]]
[[],[[q]]]
[[],[[q]]]
[[],[[c,c]]]
[[],[[q,q]]]
```

- What are the masses of the BSM particles of the individual elements? What are the element “weights” (= cross section * branching ratio)?

6) More Fun

For more fun with SModelS, see http://smodels.readthedocs.io/en/latest/

For more fun with SModelS in python, see:
http://smodels.readthedocs.io/en/latest/Examples.html



SModelS
latest
Search docs

What's New in Version 1.1
Installation and Deployment
Using SModelS
SModelS Tools
Detailed Guide to SModelS
⊟ How To's
Examples using the Database Browser
SModelS Code Documentation
C++ Interface

WRITE THE DOCS

Love Documentation? Write the Docs is a community full of people like you!

Docs » How To's

Edit on GitHub

How To's

Below a few examples are given for how to utilize SModelS and some of the SModelS tools as a Python library [*]:

- How to load the database (download the Python code here, IPython notebook here)
- How to obtain experimental upper limits (download the Python code here, IPython notebook here)
- How to obtain experimental efficiencies (download the Python code here, IPython notebook here)
- How to compute leading order cross sections (for MSSM) (download the Python code here, IPython notebook here)
- How to compute next-to-leading order cross sections (for MSSM) (download the Python code here, IPython notebook here)
- How to print decomposition results (download the Python code here, IPython notebook here)
- How to print theory predictions (download the Python code here, IPython notebook here)
- How to compare theory predictions with experimental limits (download the Python code here, IPython notebook here)
- How to compute the likelihood and chi2 for a theory predictions (download the Python code here, IPython notebook here)
- How to find missing topologies (download the Python code here, IPython notebook here)
- How to generate ascii graphs (download the Python code here, IPython notebook here)

Future

We intend to extend the functionality of SModelS in several ways:

- Extend to non-Z_2 / non-MET topologies
- Extend to long-lived particles (e.g. HSCP scenarios)
- Combine up to hundreds of correlated signal regions (experiments started to provide covariance matrices)
- Combine uncorrelated analyses
- Make use of positive results (“excesses”)