HiggsTools: A Toolbox for BSM Scalar Phenomenology

Henning Bahl

gitlab.com/higgsbounds/higgstools

THE UNIVERSITY OF CHICAGO

Workshop on Automatic Phenomenology, June 9th 2022

This work is based on a collaboration with

Jonas Wittbrodt, Thomas Biekötter, Sven Heinemeyer, Cheng Li, Steven Paasch, and Georg Weiglein.

Pheno workflow

Theoretical model → Predictions → Compatibility with experimental data

This talk

Theoretical model

- Define field content + interactions.
- Check theoretical constraints:
  - gauge invariance,
  - anomaly cancellation,
  - perturbative unitarity,
  - vacuum stability.
- …

Predictions

- Decay rates.
- Collider production cross sections.
- Astrophysical/cosmological (e.g. DM) observables.
- Electroweak precision observables.
- Flavor observables.
- …

Compatibility with experimental data

- Collider searches.
- Collider measurements.
- Astrophysical/cosmological data.
- Electroweak precision data.
- Flavor data.
- …

Some of these steps are easy, most are quite involved! → Automation.

HiggsBounds and HiggsSignals — recap

HiggsBounds-5

- Input: Higgs XS and BRs or effective couplings.
- Limits: database of over 200 LEP, Tevatron, LHC Higgs searches.
- Output: check if model point is excluded by one of these searches.
- Developed since ~ 2008, written in Fortran.

HiggsSignals-2

- Input: Higgs XS and BRs or effective couplings.
- Measurements: LHC Run 1 + 2 Higgs mass and signal strength measurements.
- Output: χ^2 value quantifying compatibility with experimental Higgs data.
- Develop since ~2012, written in Fortran.

Presenting HiggsTools

HiggsTools is a complete and extended rewrite of HiggsBounds and HiggsSignals in modern C++.

HiggsPredictions-1

HiggsBounds-6

HiggsSignals-3

- Handles user input (model predictions).
- Provides tabulated cross sections and BRs.
- Common process definitions and clustering.

→ C++ interface for high performance; Python and Mathematica interfaces for ease of use.

HiggsPredictions

What does the model predict for Higgs XS and BRs?

HiggsPredictions overview

- All information about particles and their properties are stored in `Predictions` class.
- Information about each particle is stored in `BSMParticle` class:
  - Quantum numbers: electric charge, CP.
  - All relevant production and decay modes for LEP and LHC.
  - Decays into mixed SM/BSM pairs (e.g. $H \to ZA$) and into pure BSM pairs (e.g. $h \to HH$).
- Tabulated XS and BRs for reference particles (e.g. SMHiggs).
- Effective coupling input to set particle properties relative to these reference models.
- Automatically calculates XS and BRs in terms of effective couplings.

HiggsPredictions input vs. HB-5 input

HB-5 input

```
subroutine HiggsBounds_neutral_input_SMBR (  BR_hjss , BR_hjcc , BR_hjbb , BR_hjtt , BR_hjmumu ,
                                             BR_hjtautau , BR_hjWW , BR_hjZZ ,
                                             BR_hjZga , BR_hjgaga , BR_hjgg )
```

- Arrays of length #particles are used as input.
- Every XS and BR has to be set for every particle.
- Adding new decay modes breaks existing code.

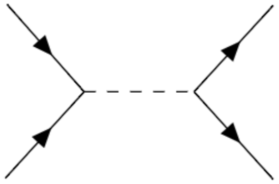
HiggsPredictions input

- Object oriented → properties of all particle can be set independently: e.g. p.setBr("bb", 0.4).
- Adding new types of production and decay modes straightforward.

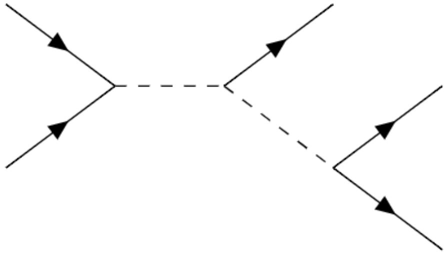
HiggsPredictions — process types

All processes used in HiggsBounds and HiggsSignals are now consistently defined as one of four process types:

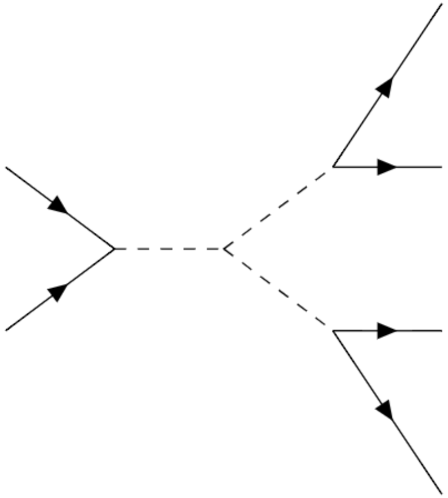
a) Channel (1 BSM particles).
b) Chain decay (2 BSM particles).
c) Pair decay (3 BSM particles).
d) Pair production (2 BSM particles).



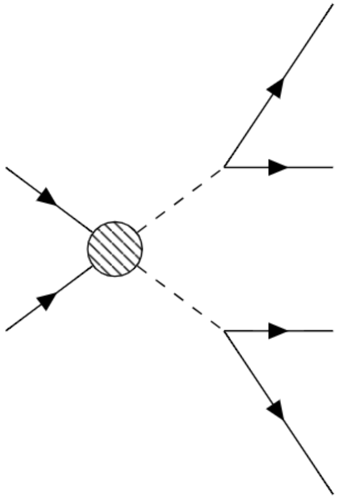
(a) channel



(b) chain decay



(c) pair decay



(d) pair production

HiggsPredictions — XS and BR predictions

- All XS and BRs can be set by the user.
- For many BSM Higgs models, rescaling of SM results is a very good approximation.
- Therefore, user can also employ effective coupling input.
- For effective coupling input, XS and BR are calculated by HiggsPredictions (assuming absence of contribution by BSM particle.

```
cpls = Higgs.predictions.NeutralEffectiveCouplings()
cpls.tt = 1
cpls.bb = 1
cpls.tautau = 1
cpls.ss = 1
cpls.mumu = 1
cpls.gg = 1
cpls.ZZ = 1
cpls.WW = 1
cpls.gamgam = 1
cpls.Zgam = 1
cpls.cc = 0.9 + 1j * 0.1
Higgs.predictions.effectiveCouplingInput(
    h,
    cpls,
    reference=HP.ReferenceModel.SMHiggsEW)
```

HiggsPredictions — XS predictions

prod. channel	coupling dep.	mass range [GeV]	source
ggH	$c_t, \tilde{c}_t, c_b, \tilde{c}_b$	$10 - 3000$	SusHi
bbH	$c_b, \tilde{c}_b$	$10 - 3000$	resc. of SM result
VBF	c_Z, c_W	LHC8: $1 - 1050$, LHC13: $1 - 3050$	HAWK
$t\bar{t}H$	$c_t, \tilde{c}_t$	$25 - 1000$	MadGraph
tH (t channel)	$c_t, \tilde{c}_t, c_W$	$25 - 1000$	MadGraph
tWH	$c_t, \tilde{c}_t, c_W$	$25 - 1000$	MadGraph
WH	c_W, c_t	$1 - 2950$	vh@nnlo
$qq \to ZH$	c_Z, c_t	$1 - 5000$	vh@nnlo
$gg \to ZH$	$c_t, c_b, c_Z, \tilde{c}_t, \tilde{c}_b$	$1 - 5000$	vh@nnlo
$b\bar{b} \to ZH$	c_b	$1 - 5000$	vh@nnlo
$q_i q_j \to H$	$c_{q,ij}, \tilde{c}_{q,ij}$	$1 - 5000$	vh@nnlo
$q_i q_j \to H^\pm$	$c_{qL,ij}, c_{qR,ij}$	$200 - 1150$	2109.10366
$q_i q_j \to H + \gamma$	$c_{q,ij}, \tilde{c}_{q,ij}$	$200 - 1150$	2109.10366
$q_i q_j \to H^\pm + \gamma$	$c_{qL,ij}, c_{qR,ij}$	$200 - 1150$	2109.10366
$b\bar{b} \to ZH$	c_b	$200 - 1150$	2109.10366
$pp \to H^\pm tb$	$c_{L,tb}, c_{R,tb}$	$145 - 2000$	1507.02549,1607.05291
$pp \to H^\pm \phi$	$c_{H^\pm \phi W^\mp}$	$m_\phi : 10 - 500$, $m_{H^\pm} : 100 - 500$	2103.07484

- Predictions also available for decay modes: $H \to f\bar{f}, WW, ZZ, \gamma\gamma, gg$.
- Additional K-factor based on YR4 numbers is applied to derived XS and BR values.

HiggsBounds

Is your model excluded by searches for BSM scalars?

HiggsBounds — overview

HiggsBounds uses a library of experimental limits. For every limit, it

1. checks which particles in the model are relevant for each *role* in the process;

2. finds all *maximal clusters* for each *role* that fulfill the analysis assumptions;

3. computes the model predictions for all assignments of *clusters* to the process roles;

4. obtains the expected and observed ratios (i.e., model prediction/limit).

For each particle, the most sensitive limit is selected based on the expected ratio. The parameter point is regarded as allowed if the observed ratio < 1 for all selected limits.

HiggsBounds — clustering

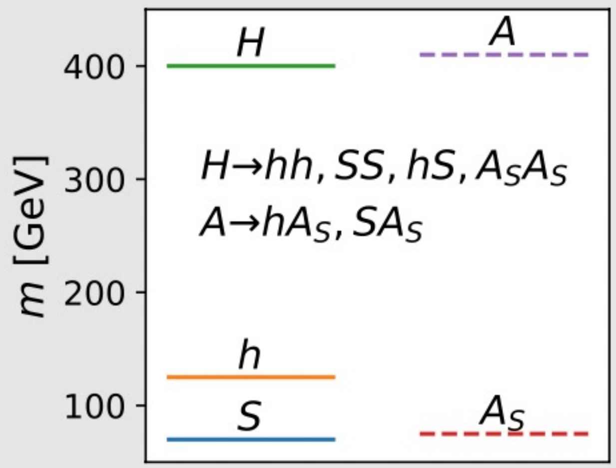
Multiple particles of similar mass can remain unresolved.

→ Define clusters of particles with masses m_i fulfilling

$$\max(m_i) - \min(m_i) \leq r_{\text{abs}} + r_{\text{rel}} \cdot \text{mean}(m_i)$$

- Mass resolutions given by experiment or estimated.
- Can also account for theoretical mass uncertainties Δm_i:
  - `Cautious`: only if entire $\pm\Delta m_i$ regions overlap.
  - `Eager`: as soon as $\Delta m + r$ regions touch.
  - `Ignore`: ignore Δm_i for clustering.
- Clustering for all particle roles in all search topologies.
- Consistent treatment of all implemented searches.

$pp \to \phi_i \to h_{125}\phi_j, h_{125} \to \tau\tau, \phi_j \to bb$



Clustering to $\{H, A\} \to \{h\}\,\{S, A_S\}$

HiggsBounds — experimental searches

HiggsBounds-5

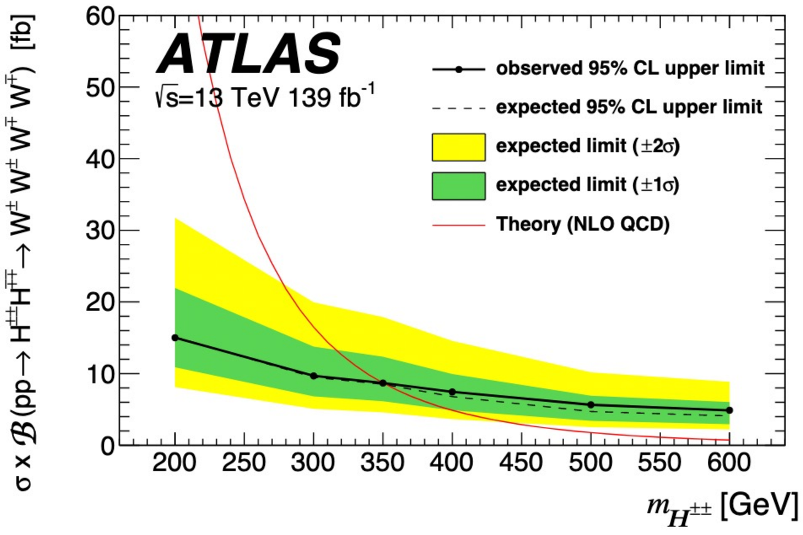
- All process definitions were hard-code in Fortran.
  ⚡ Maintainability: adding new limits requires non-trivial code changes.
  ⚡ Consistency: implementation of different limits can differ significantly.

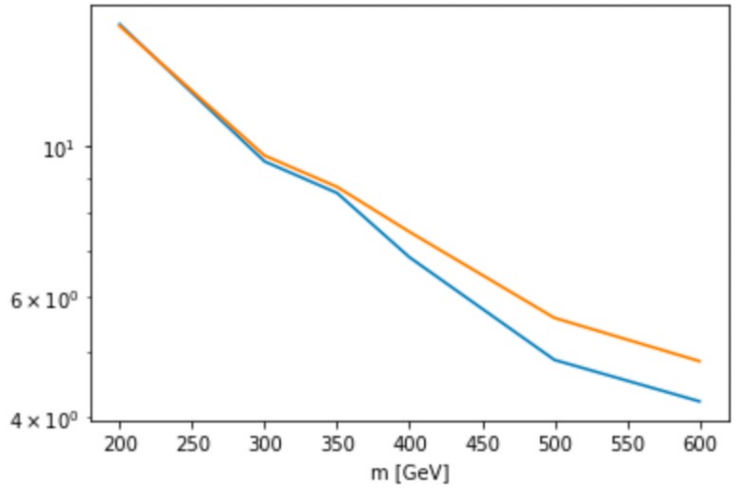
HiggsBounds-6

- All searches fully defined through *json* datafiles.
  → implementation of new limit does not require any changes in C++ code.
- Common set of constraints used to set analysis assumptions.
- Database of experimental results separate and independent of main code.
- Public implementation and validation scripts for every analysis.

HiggsBounds — limit example

- Publicly available iPython notebooks for every limit.
- If possible, data is pulled from HEPdata.
- Outputs json limit file containing all information about a limit.
- Validation plots are generated automatically.






```
{
    "limitClass": "PairProductionLimit",
    "id": 210111961,
    "reference": "2101.11961",
    "source": "https://www.hepdata.net/record/ins1688938",
    "citeKey": "ATLAS:2021jol",
    "collider": "LHC13",
    "experiment": "ATLAS",
    "luminosity": 139.0,
    "process": {
        "firstDecay": [
            "WWsamesign"
        ],
        "secondDecay": [
            "WWsamesign"
        ]
    },
    "analysis": {
        "equalParticleMasses": true,
        "grid": {
            "massFirstParticle": [
                200.0,
                300.0,
                350.0,
                400.0,
                500.0,
                600.0
            ]
        },
        "limit": {
            "observed": [
                15.025,
                9.6896,
                8.7162,
                7.4858,
                5.5951,
                4.8339
            ],
            "expected": [
                15.111,
                9.4993,
```

HiggsBounds — dataset

HiggsBounds data set available at gitlab.com/higgsbounds/hbdataset.

Current status:

- 258 limits from 165 experimental publications:
    - 25 LEP searches from 13 publications (mostly combinations),
    - 90 LHC Run 1 searches from 26 ATLAS and 37 CMS publications,
    - 143 LHC Run 2 searches from 44 ATLAS and 45 CMS publications.

- dataset strictly superior to the HB-5 dataset:
    - full Run-2 results in many channels,
    - doubly charged Higgs searches. ← New!

HiggsSignals

Is your model compatible with measurements of the SM-like Higgs boson?

HiggsSignals overview

HiggsSignals uses a library of Higgs measurement. Based on these measurements it computes

$$\chi^2 = (\mu - \hat{\mu})^T \left[\Delta_{\text{obs}}^T \text{Corr}_{\text{obs}} \Delta_{\text{obs}} + \Delta_{\text{theo}}^T \text{Corr}_{\text{theo}} \Delta_{\text{theo}}\right]^{-1} (\mu - \hat{\mu})$$

with μ being a normalized signal rate, mass, or coupling measurement.

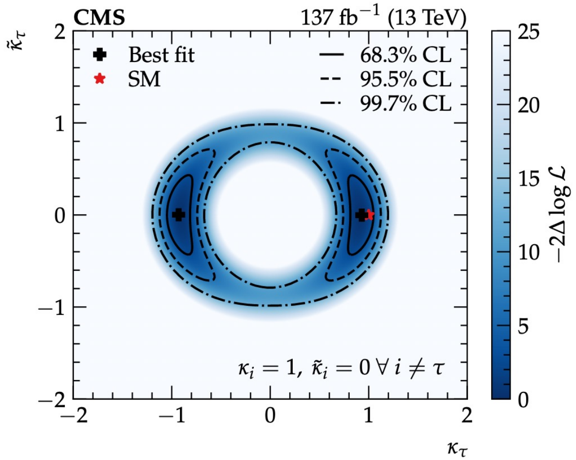
HS-2 distinguished between three slightly different types of measurements:

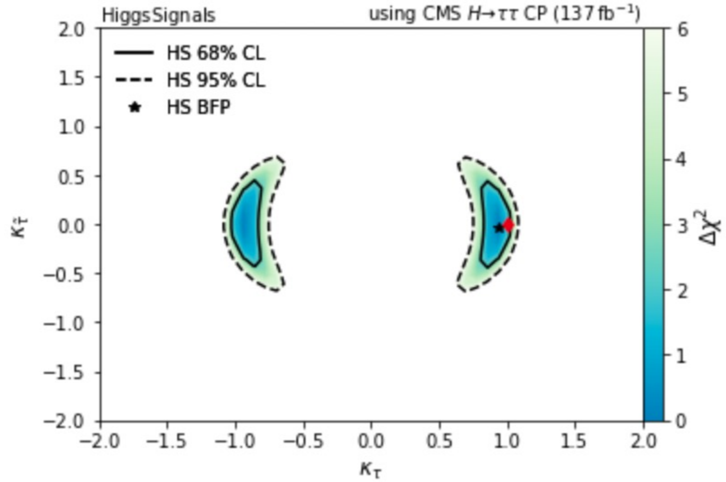
- peak observables,
- ATLAS/CMS Run 1 combinations,
- STXS observables.

With HS-3, all measurements are treated equally and are consistently implemented using json files.

HiggsSignals —meas. example

- Publicly available iPython notebooks for every measurement.
- If possible, data is pulled from HEPdata.
- Outputs json limit file containing all information about a measurement.
- Validation plots are generated automatically.






```json
{
    "id": 211004836,
    "reference": "2110.04836",
    "source": "Aux. Tab. 2, Aux. Fig. 30",
    "citeKey": "CMS:2021sdq",
    "collider": "LHC13",
    "experiment": "CMS",
    "luminosity": 137.0,
    "referenceMass": 125.38,
    "referenceModel": "SMHiggsEW",
    "massResolution": 18.75,
    "subMeasurements": {
        "alphaCP": {
            "coupling": "alphaCPTauYuk",
            "obsCoupling": [
                -0.3490658503988659,
                -0.017453292519943295,
                0.3141592653589793
            ],
            "process": {
                "channels": [
                    [
                        "H",
                        "tautau"
                    ],
                    [
                        "vbfH",
                        "tautau"
                    ]
                ]
            }
        },
```

HiggsSignals — dataset

HiggsSignals data set available at gitlab.com/higgsbounds/hsdataset.

Current status:

- 22 measurements (11 ATLAS Run-2, 9 CMS Run-2 and 2 Run-1 Combination) with 136 individual observables.

- dataset strictly superior to the HS-2 dataset:
  - full Run-2 results in many channels,
  - Updated mass measurements,
  - new type of measurement: CMS measurement of $H \to \tau\tau$ CP phase.

Interfaces and code examples

How to use HiggsTools?

HiggsTools — quick start guide

Extensive online documentation: higgsbounds.gitlab.io/higgstools/index.html

C++ library:

1. Make sure you have the right dependencies (gcc >= 9, clang >= 5, CMake >= 3.17, Python >=3.5).
2. Download HiggsTools code and data repositories from gitlab.com/higgsbounds.
3. In the code directory, type

```
mkdir build && cd build
cmake ..
make
```

Python interface: In the code directory, type `pip install .`

Mathematica interface:

```
cmake -DHiggsTools_BUILD_MATHEMATICA_INTERFACE=ON ..
```

SLHA and datafile input still available via Python interface.

Code example

Conclusions

Conclusions

- HiggsTools is a complete rewrite of HiggsBounds and HiggsSignals in modern C++.
- Three sub-packages:
  - HiggsPredictions → model input + predictions,
  - HiggsBounds,
  - HiggsSignals.
- C++, Python, and Mathematica interfaces.
- Many new features: predictions for most relevant XS and BRs in terms of effective couplings, easy extensible datasets, doubly-charged scalars, …
- Ready to use **now** (paper in preparation).
- Code and extensive documentation available at

gitlab.com/higgsbounds/higgstools