HiggsTools: A Toolbox for BSM Scalar Phenomenology

Jonas Wittbrodt

TOOLS 2021, 23.11.21

Lund University, Theoretical Particle Physics

HiggsBounds

A Phenomenological Perspective

I have this awesome new model, it has dark matter, gravitational waves, and can do baryogenesis . . . now I just need to check if its 29 scalars are fine with the LHC data.

Using all applicable search results, set a 95 % C.L. limit on the model parameter space.

- **quantity**: There are a lot of Higgs measurements and even more searches.
- **assumptions**: Is analysis X really applicable to my particle Y?
- **statistical interpretation**: Need a reasonable combination of all of those results applied to all of those particles.
- **channels**: Large and increasing number of channels that need model predictions.

HiggsBounds and HiggsSignals — A Recap

HiggsBounds-5

[Bechtle et al. **2006.06007**]

- exclusion bounds from over 200 analyses at LEP, Tevatron and the LHC
- determine most sensitive analyses to obtain a combined limit at $\sim 95\,\%$ C.L..
- use model-independent limits (if possible) or check analysis assumptions
- input framework for all relevant model predictions

HiggsSignals-2

[Bechtle et al. **2012.09197**]

- LHC Run 1 combination + latest 13 TeV LHC results up to 139 fb^{-1}
- χ^2 fit of model predictions to data including all available correlations
- uses HiggsBounds to handle model input

HiggsTools — An Overhaul

HiggsTools is a complete rewrite of HiggsBounds and HiggsSignals in modern C++ with a native Python interface. It contains three parts:

HiggsPredictions

- handles user input for model predictions
- tabulated cross sections and BRs (SM, effC, ...)
- common process definitions and clustering

HiggsBounds

Compares experimental search results to provided model predictions.

HiggsSignals

Compares experimental measurements to provided model predictions.

HiggsPredictions — Overview

- stores model predictions for BSM particles in the BSMParticle class
  - quantum numbers: electric charge, CP, (potentially add spin)
  - all relevant production and decay modes for LEP and LHC
  - decays into pairs of other BSM particles, or mixed SM/BSM pairs (e.g. $H \to ZA$)
  - theoretical mass uncertainty
- Predictions class contains all BSM particles in a model and stores properties that depend on more than one BSM particle (e.g. non-resonant $pp \to h_1 h_2$)
- tabulated properties of reference particles, e.g. SMHiggs
- effective coupling input to set BSM properties relative to these reference models

Model Predictions

The HiggsBounds-5 input framework was suboptimal to use and hard to extend.

```
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)
```

Structure of arrays approach (each argument has one entry for each particle).

- every BR and cross section has to be provided for every particle
- adding new decay modes (e.g. $H \to jj$) breaks existing code

- object oriented interface where properties of all particles can be set independently, e.g. p.setBr("bb", 0.4)
- C++ interface for high performance, Python (and Mathematica) interface for ease of use

HiggsBounds — Overview

For every implemented limit HiggsBounds performs the following steps:

1. check which particles in the model are *relevant* for each *role* in the process
2. find all *maximal clusters* for each *role* that fulfill the analysis assumptions
3. for all *assignments* of clusters to the process *roles* compute the model predictions
4. obtain the observed and expected ratios (model prediction/limit)

Then select the most sensitive limit for each particle based on the `expRatio`. The parameter point is allowed if the observed ratio < 1 for all selected limits.

Experimental Results from Higgs Searches

In HiggsBounds-5, all process definitions were hard-coded in Fortran.

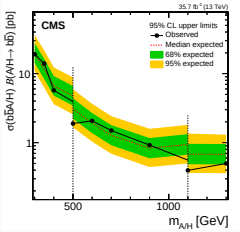
! maintainability: adding new limits requires non-trivial code changes

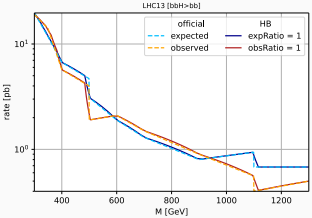
! consistency: implementations for similar processes can differ significantly

- fully define experimental search results through datafiles (`json` format)
- general class of constraints that can be used to verify analysis assumptions
- code changes only required for new process topologies, all topologies up to $xx \to \phi_i \to \phi_j \phi_k \to yyzz$ are implemented
- database of experimental results separate and independent of the main code
- public implementation and validation scripts for every analysis

Experimental Results from Higgs Searches — Example


```
{
    "limitClass": "ChannelLimit",
    "id": 180512191,
    "reference": "1805.12191",
    "source": "Fig. 6/Tab. 1",
    "citeKey": "CMS:2018hir",
    "collider": "LHC13",
    "experiment": "CMS",
    "luminosity": 35.7,
    "process": {"channels": [["bbH", "bb"]]},
    "analysis": {
        "massResolution": {"absolute": 15, "relative": 0.15},
        "grid": {
            "mass": [300.0, 350.0, 400.0, 499.0,
                     500.0, 600.0, 700.0, 900.0, 1099.0,
                     1100.0, 1300.0]
        },
        "limit": {
            "observed": [19.1, 14.0, 5.7, 4.0, 1.9, 2.1,
                         1.5, 0.9, 0.56, 0.4, 0.5],
            "expected": [19.7, 11.7, 6.7, 4.7, 3.1, 1.9,
                         1.3, 0.8, 0.94, 0.68, 0.68]
        }
    }
}
```



[CMS 1805.12191]

Experimental Results from Higgs Searches — The Dataset

HiggsBounds dataset at `gitlab.com/higgsbounds/hbdataset`.

What is (currently) included?

- 255 search results from 162 experimental publications
  - 25 LEP searches from 13 publications (mostly combinations)
  - 90 LHC Run 1 searches from 26 ATLAS and 36 CMS publications
  - 140 LHC Run 2 searches from 42 ATLAS and 45 CMS publications
- dataset strictly superior to the HB-5 dataset

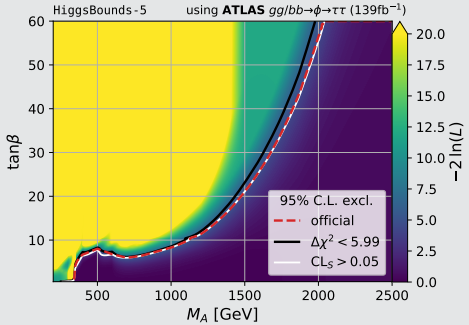
What is not (yet) included?

- 7 TeV LHC and Tevatron results
- double-charged scalars — *coming soon*
- Z'/W' searches — *possible extension*

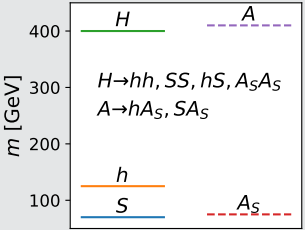
Clustering

Multiple particles of similar mass could remain unresolved.

MSSM $pp \to \{H, A\} \to \tau\tau$



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



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

Clustering — Implementation

A cluster C of particles with masses m_i is valid if

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

- mass resolutions given by experiment or estimated
- 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 for clustering

- clustering for all particle roles in all search topologies
- consistent treatment of all implemented searches

HiggsSignals — Overview

HiggsSignals computes a χ^2 of the model predictions compared to all of the available LHC Higgs measurements.

1. evaluate the model predictions for each observable and compute the residuals
2. obtain the covariance matrix including experimental uncertainties and correlations as well as reference model uncertainties and correlations
3. compute the χ^2 from rate measurements
4. add χ^2 contributions from other measurements (mass, total width, CP-phase, etc.)

Experimental Results — Measurements

Measurements in HiggsSignals-2 were already implemented through datafiles.

However, HiggsSignals-2 distinguished three observable types:

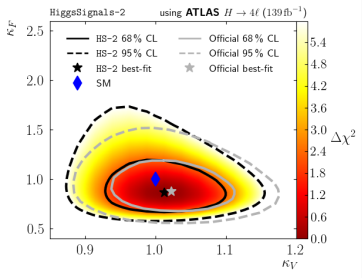
- peak observables
- the ATLAS/CMS Run 1 combinations
- STXS observables

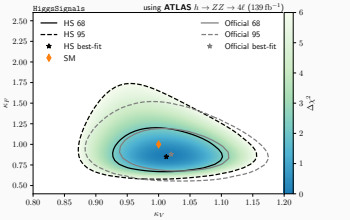
with subtly different implementations.

In HiggsTools, all HiggsSignals measurements are implemented through json files and treated equally.

Experimental Results — Example


```
{
  "id": 200403447,
  "reference": "2004.03447",
  "source": "https://doi.org/10.17182/hepdata.94311.v1/t25",
  "citeKey": "ATLAS:2020rej",
  "collider": "LHC13",
  "experiment": "ATLAS",
  "luminosity": 139.0,
  "referenceMass": 125,
  "massResolution": 2.5,
  "bins": {
      "gg2H_0j_pTH_Low": {
          "exp": [0.151, 0.176, 0.201],
          "obs": [0.115, 0.17, 0.225],
          "process": {"channels": [["H", "ZZ"]]}
      },
      "gg2H_0j_pTH_High": {
          "exp": [0.51, 0.55, 0.59],
          "obs": [0.52, 0.63, 0.74],
          "process": {"channels": [["H", "ZZ"]]}
      },
      "VH_Lep": {
          "exp": [0.016, 0.0164, 0.0168],
          "obs": [0.004, 0.022, 0.05],
          "process": { "channels": [["HZ", "ZZ"], ["HW", "WW"]]},
          "channelWeights": [0.067316, 0.2172]
      }
```



[ATLAS 2004.03447]

Development Status

HiggsTools is available now as public beta: gitlab.com/higgsbounds/higgstools

HiggsPredictions

- ready for use, C++, Python and Mathematica interface available
- some cross section parametrizations from HB-5 not yet reimplemented
- Python scripts for backwards compatible SLHA/datafile input will be added

HiggsBounds

- ready for use, dataset superior to the latest HB-5 dataset
- check the dataset repository for detailed status

HiggsSignals

- in development, will be done this year

Examples — The SM

```python
import Higgs
import Higgs.predictions as HP



pred = Higgs.Predictions()
h = pred.addParticle(HP.NeutralScalar("h", "even"))


h.setMass(125)
HP.effectiveCouplingInput(h, HP.smLikeEffCouplings)

bounds = Higgs.Bounds("/path/to/HBDataSet")
print(bounds(pred))

# signals = Higgs.Signals("/path/to/HSDataSet")
# print(signals(pred))
```

```cpp
#include "Higgs/Bounds.hpp"
#include "Higgs/Predictions.hpp"
#include <iostream>
namespace HP = Higgs::predictions;

int main() {
    auto pred = Higgs::Predictions{};
    auto &h = pred.addParticle(
        HP::BsmParticle{"h", HP::ECharge::neutral,
                        HP::CP::even});
    h.setMass(125);
    effectiveCouplingInput(h, HP::smLikeEffCouplings);

    const auto bounds = Higgs::Bounds("/...");
    std::cout << bounds(pred) << std::endl;

    // const auto signals = Higgs::Signals("/...");
    // std::cout << signals(pred) << std::endl;
}
```

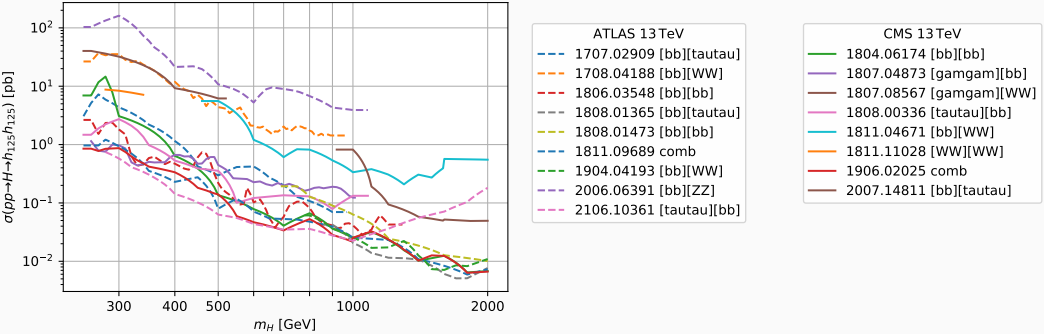
```
HiggsBounds result: allowed
particle | obsRatio | expRatio | selected limit description
---------|----------|----------|---------------------------
    h    |   0.789  |   4.049  | LHC8 [vbfH,HW,Htt,H,HZ]>[bb,tautau,WW,ZZ,gamgam]
         |          |          | from CMS-PAS-HIG-12-045 (CMS 17.3fb-1, M=(110, 600))
```

Example — Resonant SM-Higgs pair production

```
import Higgs
import Higgs.predictions as HP


pred = Higgs.Predictions()
h = pred.addParticle(HP.NeutralScalar("h","even"))
h.setMass(125)
HP.effectiveCouplingInput(h, HP.smLikeEffCouplings)

H = pred.addParticle(HP.NeutralScalar("H"))
H.setMass(400)
# set cxn through reference model
H.setNormalizedCxn("LHC13", "ggH", 0.05, "SMHiggs")
H.setDecayWidth("h", "h", 1.0)

bounds = Higgs.Bounds("/path/to/HBDataSet")
hbRes = bounds(pred)
print(hbRes)
```

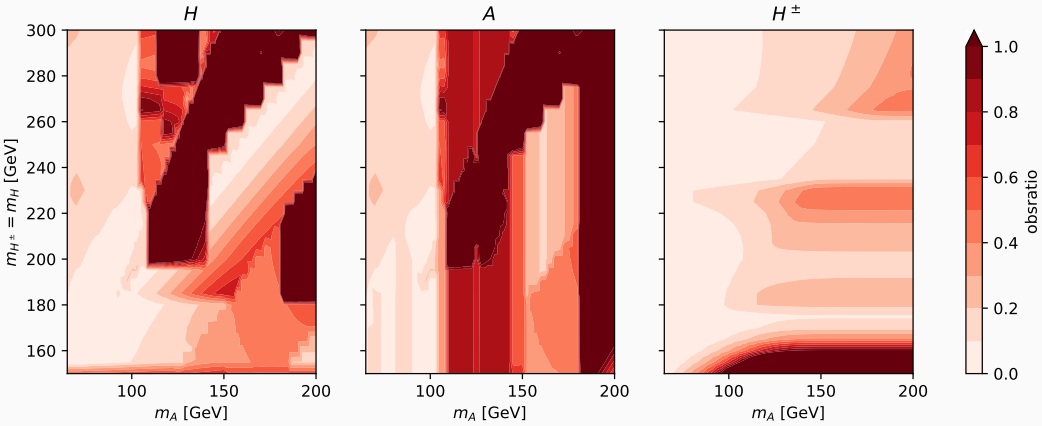
- excluded by several searches for resonant h_{125}-pair production

Example — Resonant SM-Higgs pair production II



Example — 2HDM

The cH(WA) benchmark plane from [Bahl et al. 2103.07484].



Summary

HiggsTools helps you to easily compare any model with modified scalar sector to all available collider searches and measurements.

- C++, Python, and Mathematica interfaces for easy access
- huge and continuously updated datasets of experimental results
- performs clustering within the estimated experimental mass resolution and checks analysis assumptions
- extensible to other EW BSM particles

`https://gitlab.com/higgsbounds/higgstools`

→ Tutorial this afternoon.