Karlsruher Institut für Technologie

# An introduction to HiggsTools: new versions of HiggsBounds and HiggsSignals


Based on arXiv:2210.09332 in collaboration with

Henning Bahl, Sven Heinemeyer, Cheng Li, Steven Paasch,

Georg Weiglein and Jonas Wittbrodt


TTP BSM seminar

23rd of January 2023


Thomas Biekötter

# Introduction

Karlsruher Institut für Technologie

**General idea:** I have a model with $X$ Higgs bosons

1. Does one of them behave in agreement with measurements of the detected Higgs boson $h_{125}$?
2. Would the other Higgs bosons be excluded by searches for additional Higgs bosons?

**Answer:** Compare model predictions and experimental data

Obstacle 1: Compute model predictions (→`HiggsPredictions`)

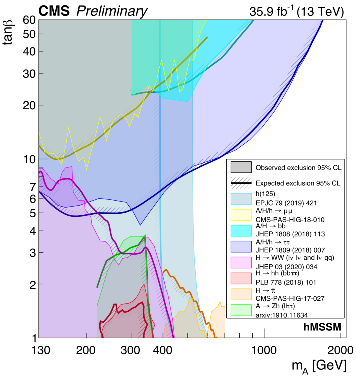
Obstacle 2: Sheer amount of experimental data (→`HiggsBounds` and `HiggsSignals`)

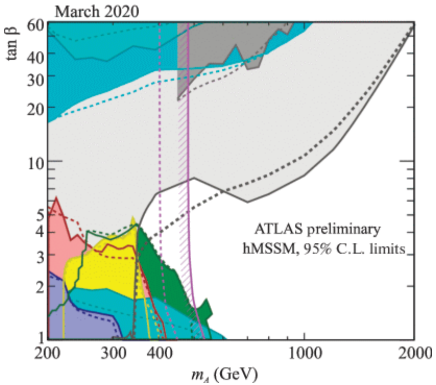
Obstacle 3: Which searches are actually relevant in my case (→`HiggsBounds`)

$\Rightarrow$ **Requires automatisation**

# Introduction

## **Searches** for BSM Higgs bosons

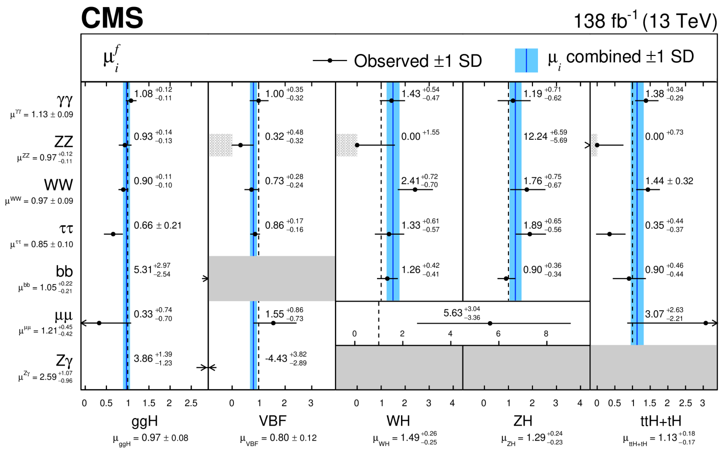


# Introduction

Karlsruher Institut für Technologie

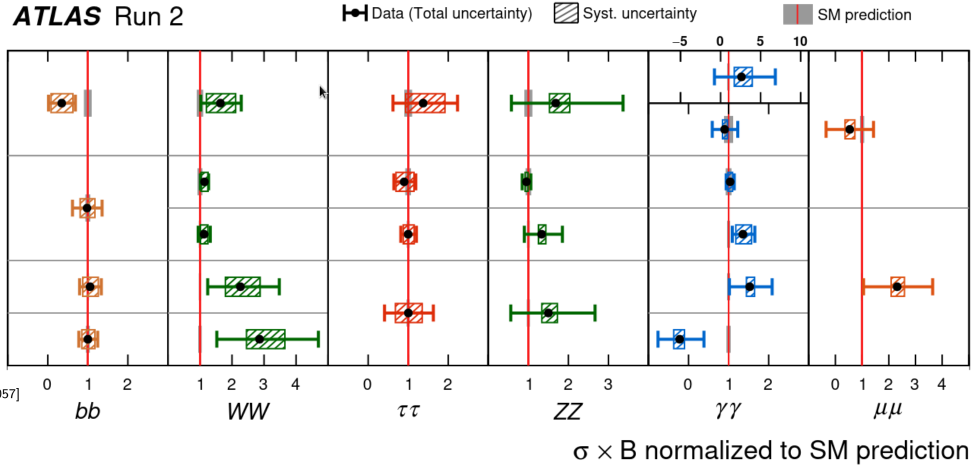
## Cross-section measurements of $h_{125}$




[CMS-HIG-22-001]

# Introduction

Cross-section measurements of $h_{125}$




[CERN-EP-2022-057]

# Introduction

Karlsruher Institut für Technologie

## History of `HiggsBounds` and `HiggsSignals`

Former members: Philip Bechtle, Oliver Brein, Karina E. Williams, Oscar Stal, Tim Stefaniak, Daniel Dercks, Tobias Klingl, Jonas Wittbrodt

HiggsBounds

`HiggsBounds` confronts models with cross-section limits from collider searches

- 02/2009, v.1 LEP and Tevatron limits
- 08/2010, v.2 Added support for charged scalars
- 05/2011, v.3 LHC 7 TeV limits included
- 05/2013, v.4 LHC 8 TeV limits included
- 03/2017, v.5 LHC 13 TeV limits included

`HiggsSignals` confronts models with cross-section and mass measurements of $h_{125}$

- 05/2013, v.1 Tevatron and LHC 7/8 TeV data
- 03/2017, v.2 LHC 13 TeV data included

# HiggsTools

Karlsruher Institut für Technologie

## HiggsTools: Rewrite and unification of HB and HS

The package consists of three subpackages:

- `HiggsPredictions v.1`
- `HiggsBounds v.6`
- `HiggsSignals v.3`

Written in modern C++ with python and mathematica interface

Why rewrite the whole thing?

A few pieces of code from the old versions...

# HiggsTools

# Why rewrite? Example 1

Reading data in old HS:

```
call system('ls -1 -p '//trim(adjustl(pathname_HS))//'Expt_tables/'//trim
    '/*.stxscorrTHU 2>/dev/null | xargs -L 1 basename > STXS_correlations
call system('rm -rf STXS_ncorrelations.txt')
```

→ Was maybe a good solution at some point, but not today

→ No PhD student would be willing to maintain code like this

→ Platform dependent

→ Cannot be run in parallel

# Why rewrite? Example 2

Setting particle properties in old HB:

```
subroutine HiggsBounds_neutral_input_effC( &
    ghjss_s, ghjss_p, ghjcc_s, ghjcc_p, &
    ghjbb_s, ghjbb_p, ghjtt_s, ghjtt_p, &
    ghjmumu_s, ghjmumu_p, &
    ghjtautau_s, ghjtautau_p, &
    ghjWW, ghjZZ, ghjZga, &
    ghjgaga, ghjgg, ghjhiZ)
```

- $\rightarrow$ User cannot set only input relevant for him
- $\rightarrow$ All input arrays have to be declared and set
- $\rightarrow$ Adding new arguments would break all existing codes

# Why rewrite? Example 3

Framework to add new data:


→ Data sets were extended *by hand*

→ No common data formats

→ Lack of consistency between implementation of similar processes

# HiggsTools

Karlsruher Institut für Technologie

## `HiggsPredictions`

Helper to compute, set and store model predicions

## `HiggsBounds`

Check BSM scalars against cross section limits

## `HiggsSignals`

Check if the cross-section measurements at 125 GeV are reproduced

A closer look at the three subpackages

# HiggsPredictions

Used to set and store the model predictions:

- Particle properties: Mass, charge, CP, cross sections, branching ratios
- Information depending on more particles, e.g. non-resonant pair production cross sections
- Contains properties of reference particles, e.g. SM Higgs boson
- Can compute cross sections and branching ratios as functions of effective couplings

```python
import Higgs.predictions as HP

pred = HP.Predictions()
h = pred.addParticle(HP.NeutralScalar("h"))
h.setMass(125.09)
HP.effectiveCouplingInput(h, HP.smLikeEffCouplings)

H = pred.addParticle(HP.NeutralScalar("H"))
H.setMass(1000.0)
H.setDecayWidth("h", "h", 1)
H.setCxn("LHC13", "ggH", 1)
```

# HiggsPredictions

Karlsruher Institut für Technologie

Used to set and store the model predictions:

- Particle properties: Mass, charge, CP, cross sections, branching ratios
- Information depending on more particles, e.g. non-resonant pair production cross sections
- Contains properties of reference particles, e.g. SM Higgs boson
- Can compute cross sections and branching ratios as functions of effective couplings

```
Install["/Path/To/HiggsTools/build/wstp/MHiggsTools"];

HPAddParticle["h", 125.09, "neutral", "even"];

HPSMLikeEffCouplings["h"]

HPAddParticle["H", 1000, "neutral", "even"];

HPSetDecayWidth["H", "h", "h", 1];

HPSetCxn["H", "LHC13", "ggH", 1];
```

# HiggsPredictions

Karlsruher Institut für Technologie

Excess to all YR-4 cross sections and branching ratios, and more...

| 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 |
| $q_i q_j \to H^\pm$ | $c_{qL,ij}, c_{qR,ij}$ | $200 - 1150$ | Ref. [7] |
| $q_i q_j \to H + \gamma$ | $c_{q,ij}, \tilde{c}_{q,ij}$ | $200 - 1150$ | Ref. [7] |
| $q_i q_j \to H^\pm + \gamma$ | $c_{qL,ij}, c_{qR,ij}$ | $200 - 1150$ | Ref. [7] |
| $b\bar{b} \to ZH$ | $c_b$ | $200 - 1150$ | Ref. [7] |
| $pp \to H^\pm tb$ | $c_{L,tb}, c_{R,tb}$ | $145 - 2000$ | Refs. [13, 14] |
| $pp \to H^\pm \phi$ | $c_{H^\pm \phi W^\mp}$ | $m_\phi : 10 - 500$, $m_{H^\pm} : 100 - 500$ | Ref. [15] |

# HiggsBounds

Compares the predicted signal rates to the experimental 95% CL limits

$$r-\text{ratios}: \quad r_{\text{exp,obs}} = \frac{\sigma_{\text{pred}}}{\sigma_{95\%\text{C.L.}}^{\text{exp,obs}}}\;, \; r_{\text{obs}} < 1 \text{ for search with } \max(r_{\text{exp}}) \Rightarrow \text{ point is allowed.}$$

- For each limit select relevant particles, possibly clusters of particles
- Compute the predicted signal rate for each limit after clustering
- Compute the "r-ratios"
- For each particle select the limit with the largest $r_{\text{exp}}$
- Exclude parameter point if for a particle the corresponding $r_{\text{obs}} > 1$

```python
import Higgs.bounds as HB

bounds = HB.Bounds('/Path/To/HBDataSet')
hbres = bounds(pred)
print(hbres)
```

# HiggsBounds

Karlsruher Institut für Technologie

Compares the predicted signal rates to the experimental 95% CL limits

$$r-\text{ratios}: \quad r_{\text{exp,obs}} = \frac{\sigma_{\text{pred}}}{\sigma^{\text{exp,obs}}_{95\%\text{C.L.}}}, \; r_{\text{obs}} < 1 \text{ for search with } \max(r_{\text{exp}}) \Rightarrow \text{ point is allowed.}$$

- For each limit select relevant particles, possibly clusters of particles
- Compute the predicted signal rate for each limit after clustering
- Compute the "r-ratios"
- For each particle select the limit with the largest $r_{\text{exp}}$
- Exclude parameter point if for a particle the corresponding $r_{\text{obs}} > 1$

```
HBInitialize["/Path/To/HBDataSet"];
hbres = {
    HBApplyBounds[],
    HBGetSelectedBounds[]};
hbres
```

# HiggsBounds

Karlsruher Institut für Technologie

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

## What is (currently) included?

- 258 search results 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

## What is not (yet) included?

- 7 TeV LHC and Tevatron results
- $Z'/W'$ searches — *possible extension*

# HiggsSignals

Karlsruher Institut für Technologie

Performs a $\chi^2$-fit to the measurements of $h_{125}$

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

- Mass measurements
- Inclusive cross section measurements
- Cross sections in terms of simplified template cross sections (STXS)

```
import Higgs.signals as HS

signals = HS.Signals('/Path/To/HSDataSet')
chisq = signals(pred)

print(f"HiggsSignals chisq: {chisq}")
```

# HiggsSignals

Karlsruher Institut für Technologie

Performs a $\chi^2$-fit to the measurements of $h_{125}$

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

- Mass measurements
- Inclusive cross section measurements
- Cross sections in terms of simplified template cross sections (STXS)

```
HSInitialize["/Path/To/HSDataSet"];

chisq = HSGetChisq[];
Print[
    "HiggsSignals chisq: " <>
    ToString[chisq]]
```

# HiggsSignals

Karlsruher Institut für Technologie

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

**What is (currently) included?**

- 22 measurements (11 ATLAS Run-2, 9 CMS Run-2 and 2 Run-1 Combination)
  - 136 individual observables
- dataset strictly superior to the HS-2 dataset
  - full Run-2 results in many channels
  - CMS measurement of the $\tau$-Yukawa CP-phase
  - updated mass measurements (WIP)

# How to get started?

1. Required compilers: C++17 (gcc>=9 or clang>=5), CMake>=3.17, python>=3.5
2. Clone/download the code from: gitlab.com/higgsbounds/higgstools

3 (a). Build the C++ code:

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

3 (b). If you want to use the python interface, build with:

```
python -m pip install .
```

3 (c). If you want to use the mathematica interface, build with: (needs WSTP library)

```
mkdir build && cd build
cmake -DHiggsTools_BUILD_MATHEMATICA_INTERFACE=ON ..
make
```

4. Download the datasets to a separate location:

`HiggsBounds` data set: gitlab.com/higgsbounds/hbdataset
`HiggsSignals` data set: gitlab.com/higgsbounds/hsdataset

# A few examples

Karlsruher Institut für Technologie

1. Constraining Yukawa couplings to the 2nd generation fermions*
2. Sensitivity comparison: $h_{125}$-pair production limits*
3. Status of the 2HDM in light of LHC Run 2*
4. Constraining the invisible decay of $h_{125}$ [TB, Mathias Pierre: 2208.05505]

*Complete example scripts in python and mathematica available in the example folder of the `HiggsTools` repository

# 1: Constraining $Y_c$

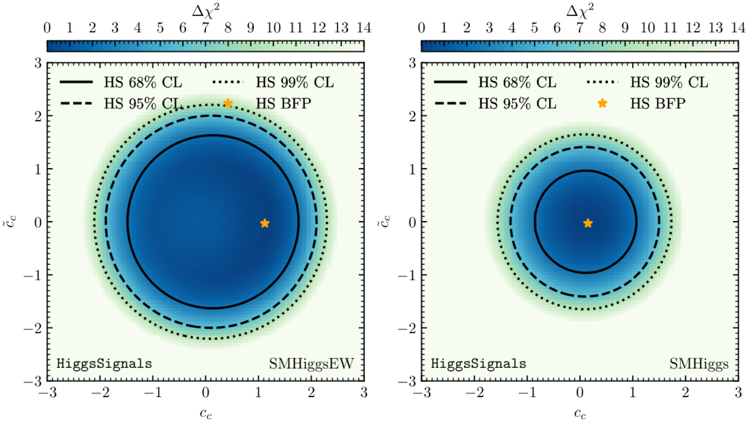
```
import Higgs.predictions as HP
import Higgs.signals as HS

pred = HP.Predictions() # create the model predictions
signals = HS.Signals('/Path/To/HSDataSet') # load HS dataset

h = pred.addParticle(HP.NeutralScalar("h")) # add a SM-like scalar
h.setMass(125.09)

cpls = HP.NeutralEffectiveCouplings() # initialize effC object
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
```

# 1: Constraining $Y_c$

Karlsruher Institut für Technologie

```python
def setEffC(ccRe, ccIm, refModel):
    cpls.cc = ccRe + 1j * ccIm
    HP.effectiveCouplingInput(h, cpls, reference=refModel)

@np.vectorize
def calcChisq(ccRe, ccIm, refModel):
    setEffC(ccRe, ccIm, refModel)
    return signals(pred)

x = np.linspace(-3, 3, N)
y = np.linspace(-3, 3, N)
X, Y = np.meshgrid(x, y)

df['ccRe'] = X.flatten()
df['ccIm'] = Y.flatten()

df['chisqSMHiggs'] = calcChisq(df['ccRe'], df['ccIm'], "SMHiggs")
df['chisqSMHiggsEW'] = calcChisq(df['ccRe'], df['ccIm'], "SMHiggsEW")
```

# 1: Constraining $Y_c$




Difference only comes from reference model!

`SMHiggsEW`: $\sigma(ggH)$ at N3LO QCD in heavy $m_t$-limit + EW corrections

`SMHiggs`: $\sigma(ggH)$ at NNLO QCD

# 2. $h_{125}$-pair production

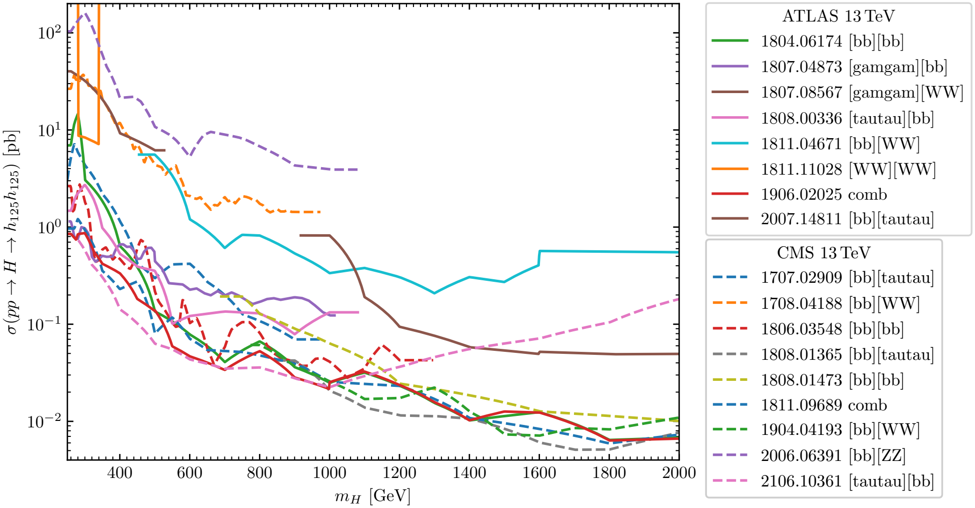
Karlsruher Institut für Technologie

```
import Higgs.predictions as HP
import Higgs.bounds as HB
import numpy as np
import pandas as pd


pred = HP.Predictions() # create the model predictions
bounds = HB.Bounds('/Path/To/HBDataSet') # load HB dataset

h = pred.addParticle(HP.NeutralScalar("h"))
h.setMass(125.09)
HP.effectiveCouplingInput(h, HP.smLikeEffCouplings)

H = pred.addParticle(HP.NeutralScalar("H"))
H.setDecayWidth("h", "h", 1)
H.setCxn("LHC13", "ggH", 1)

df = pd.DataFrame()
df['mass'] = np.linspace(250, 2001, 1000)
```

# 2. $h_{125}$-pair production

```
@np.vectorize
def runHB(mass):
    H.setMass(mass)
    return [a for a in bounds(pred).appliedLimits
        if "H" in a.contributingParticles()]
df['appliedLimits'] = runHB(df['mass'])

limits = list({a.limit() for res in df['appliedLimits'] for a in res})
limits.sort(key=lambda l: str(l.id()))
@np.vectorize
def get_obsratio(alims, id):
    for a in alims:
        if a.limit().id() == id:
            return a.obsRatio()
    return np.NaN

for lim in limits:
    df[lim.id()] = get_obsratio(df['appliedLimits'], lim.id())
```

# 2. $h_{125}$-pair production



# 3. Status of 2HDM

```python
h = pred.addParticle(HP.BsmParticle("h", "neutral", "even"))
h.setMass(125.09)
H = pred.addParticle(HP.BsmParticle("H", "neutral", "even"))
H.setMass(800)
A = pred.addParticle(HP.BsmParticle("A", "neutral", "odd"))
A.setMass(800)
X = pred.addParticle(HP.BsmParticle("X", "single"))
X.setMass(800)

...

def run_higgstools(cpl, pt): # set particle properties and run HB and HS
    set_h_properties(cpl[0], pt)
    set_H_properties(cpl[1], pt)
    set_A_properties(cpl[2], pt)
    set_X_properties(pt)
    res = bounds(pred)
    chisq = signals(pred)
    return res, chisq
```

# 3. Status of 2HDM

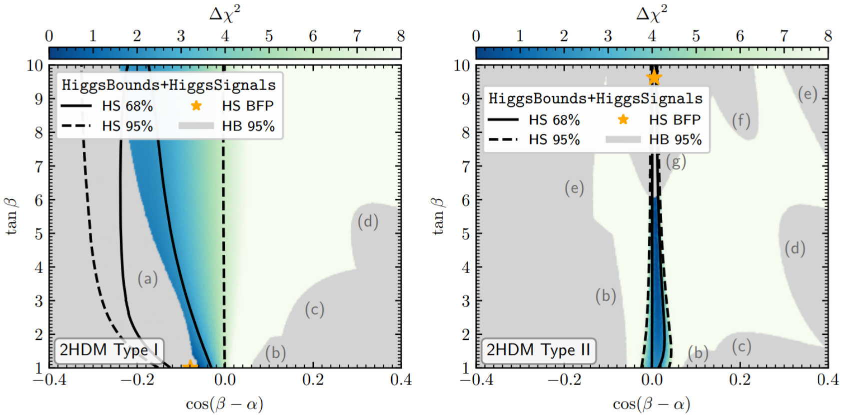
```python
def set_H_properties(dc, pt): # set properties of the H boson

    cpls = HP.NeutralEffectiveCouplings() # Set cross sections from eff. couplings
    cpls.tt = dc['tt']
    cpls.bb = dc['bb']
    cpls.ZZ = dc['ZZ']
    cpls.WW = dc['WW']
    HP.effectiveCouplingInput(H, cpls, reference=HP.ReferenceModel.SMHiggs)

    w = pt['WH'] # Set decays, here decay width obtained with NHDECAY and ScannerS
    H.setDecayWidth('gg', pt['BRH2gg'] * w)
    H.setDecayWidth('WW', pt['BRH2WW'] * w)
    H.setDecayWidth('ZZ', pt['BRH2ZZ'] * w)
    H.setDecayWidth('gamgam', pt['BRH2yy'] * w)
    H.setDecayWidth('tt', pt['BRH2tt'] * w)
    H.setDecayWidth('bb', pt['BRH2bb'] * w)
    H.setDecayWidth('tautau', pt['BRH2ll'] * w)
    H.setDecayWidth('h', 'h', pt['BRH2hh'] * w)
    H.setDecayWidth('A', 'A', pt['BRH2AA'] * w)
    H.setDecayWidth('Z', 'A', pt['BRH2ZA'] * w)
    ...
```

# 3. Status of 2HDM

Karlsruher Institut für Technologie

```
data = [] # process dataset and save output to file

for point in dataset:
    beta, alpha = point['beta'], point['alpha']
    cpl = calc_effective_couplings(alpha, beta, yuktype)
    reshb, Chisq = run_higgstools(cpl, point)
    data.append({
        'beta': beta, 'alpha': alpha,
        'chisq': Chisq, # Save HiggsSignals results
        'hexp': reshb.selectedLimits['H'].expRatio(), # Save HiggsBounds results
        'hobs': reshb.selectedLimits['H'].obsRatio(),
        'hcha': reshb.selectedLimits['H'].limit().citeKey(),
        'aexp': reshb.selectedLimits['A'].expRatio(),
        'aobs': reshb.selectedLimits['A'].obsRatio(),
        'acha': reshb.selectedLimits['A'].limit().citeKey(),
        'xexp': reshb.selectedLimits['X'].expRatio(),
        'xobs': reshb.selectedLimits['X'].obsRatio(),
        'xcha': reshb.selectedLimits['X'].limit().citeKey()})

df = pd.DataFrame(data)
df.to_csv(f'result_type{yuktype}.csv')
```

# 3. Status of 2HDM




(a) CMS: $pp \to \phi \to h_{125}h_{125} \to bb\gamma\gamma,\ bb\tau\tau,\ bbbb,\ bbVV$ [49],

(b) CMS: $pp \to \phi_1 \to h_{125}\phi_2 \to bb\tau\tau$ [43],

(c) CMS: $pp \to \phi \to Zh_{125} \to Zbb$ [31],

(d) ATLAS: $pp \to \phi \to WW,\ ZZ,\ WZ$ [75],

(e) ATLAS: $pp \to \phi \to h_{125}h_{125} \to bbbb$ [76],

(f) ATLAS: $pp \to \phi \to VV,\ Vh_{125}$ [77],

(g) ATLAS: $pp \to \phi \to \tau\tau$ [78],

# 4. Invisible decay of $h_{125}$

SM-like couplings:

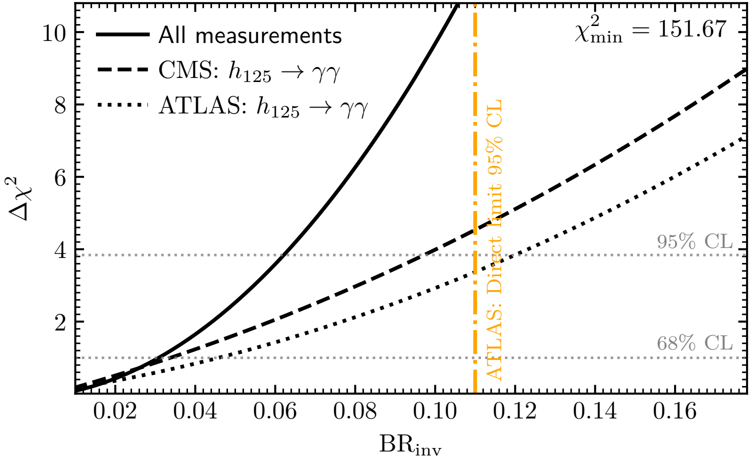
$$c_f = c_V = 1$$

Only BSM contribution:
$\mathrm{BR}(h_{125} \to \mathrm{inv})$

**Indirect limit:**

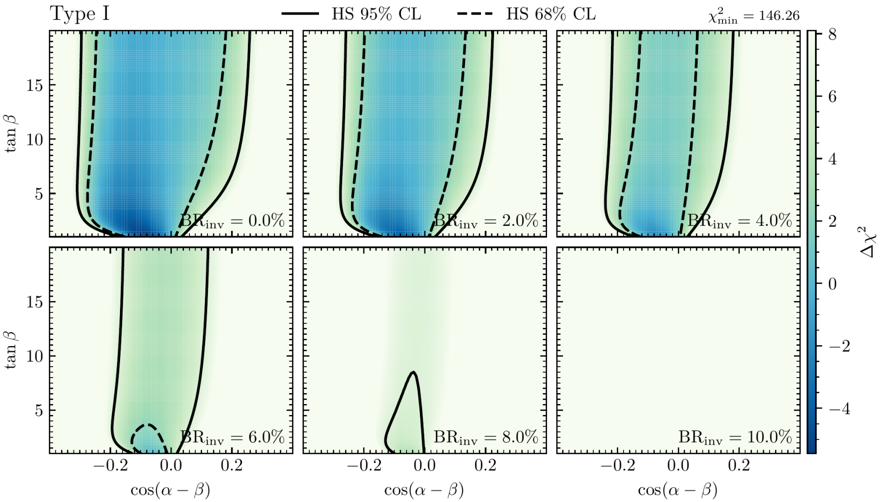
$$\mathrm{BR}_{\mathrm{inv}} < \mathbf{6.2\%} \text{ at } \mathbf{95\%}$$

Meas. of $h_{125} \to \gamma\gamma$ alone give rise to an indirect limit stronger than the direct limit on $\mathrm{BR}_{\mathrm{inv}}$ [CMS-HIG-19-015]




[TB, Mathias Pierre: 2208.05505]

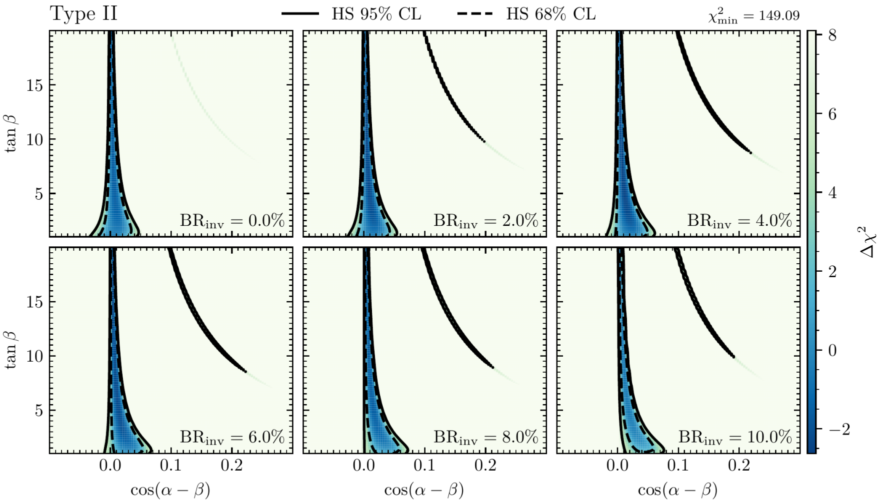
# 4. Invisible decay of $h_{125}$




[TB, Mathias Pierre: 2208.05505]

$\cos(\alpha - \beta) = 0$: Alignment limt $h_{125} = h_{\mathrm{SM}}$

# 4. Invisible decay of $h_{125}$




[TB, Mathias Pierre: 2208.05505]

$\cos(\alpha-\beta)=0$: Alignment limt $h_{125}=h_{\rm SM}$

# Summary: HiggsTools

## How to get started?

```
git clone https://gitlab.com/higgsbounds/higgstools.git
git clone https://gitlab.com/higgsbounds/hbdataset.git
git clone https://gitlab.com/higgsbounds/hsdataset.git
cd higgstools
python -m pip install .
```

## How to get help?

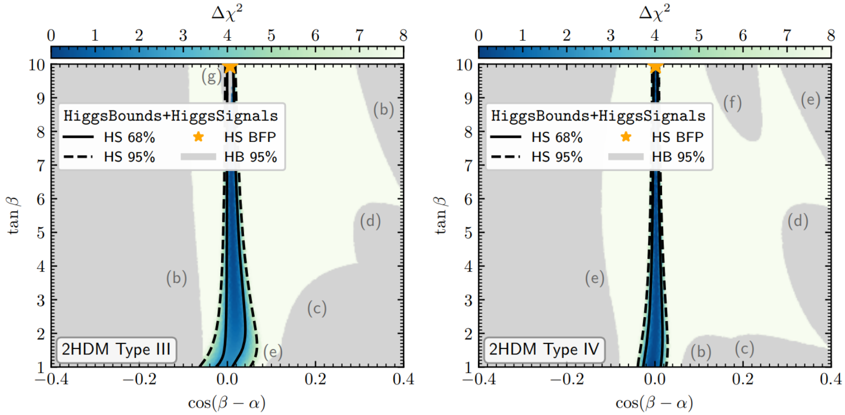
→ Write an email to the team: `higgstools@desy.de`

→ Raise an issue on gitlab: `https://gitlab.com`

→ ITP Room: 12-16 (my office)

# Thanks!

# 3. Status of 2HDM




(a) CMS: $pp \to \phi \to h_{125}h_{125} \to bb\gamma\gamma,\ bb\tau\tau,\ bbbb,\ bbVV$ [49],

(b) CMS: $pp \to \phi_1 \to h_{125}\phi_2 \to bb\tau\tau$ [43],

(c) CMS: $pp \to \phi \to Zh_{125} \to Zbb$ [31],

(d) ATLAS: $pp \to \phi \to WW,\ ZZ,\ WZ$ [75],

(e) ATLAS: $pp \to \phi \to h_{125}h_{125} \to bbbb$ [76],

(f) ATLAS: $pp \to \phi \to VV,\ Vh_{125}$ [77],

(g) ATLAS: $pp \to \phi \to \tau\tau$ [78],

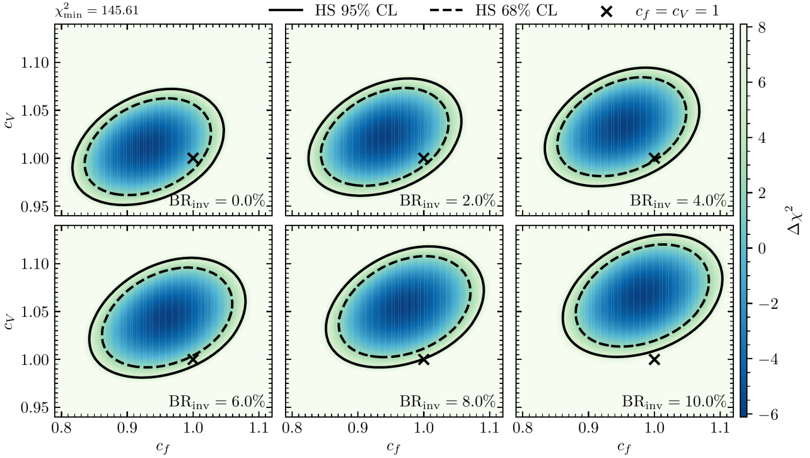
# 4. Invisible decay of $h_{125}$

Common LHC benchmark scenario:

$\{c_f, c_V\}$, where $c_f \equiv c_u = c_d = c_\ell$

Allowed regions in $\{c_f, c_V\}$ plane for all $\mathrm{BR_{inv}}$ values

**Complementarity** of direct and indirect constrains on $\mathrm{BR_{inv}}$ if $h_{125} \neq h_{SM}$




UV-completion: e.g. 2HDM type I ($c_V \leq 1$)

$\Delta\chi^2 < 0 \Rightarrow \chi^2 < \chi^2_{SM}$