Reproducible research with HEP-Tools

UNIVERSIDAD DE ANTIOQUIA
1 8 0 3

Sarah Toolbox + Jupyter

Diego Restrepo

November 29, 2016

Instituto de Física
Universidad de Antioquia
Phenomenology Group
`http://gfif.udea.edu.co`

COMHEP

Table of Contents

1. HEP Tools
2. Problems
3. Version control
4. Testing
5. Reproducibility

HEP Tools

HEP Tools

HEP Tool 1 → Model A ...

$\Downarrow$

HEP Tool 2 → Model A ...

⋮ ⋮

LHA

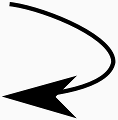
HEP Tools

HEP Tool 1 → Model A ...

$\Downarrow$

HEP Tool 2 → Model A ...

⋮ ⋮



LHA

LHA Input/Output text File

```
BLOCK SET_OF_PARAMETERS
1 Value_1 # comment 1
2 Value_2 # comment 2
        ⋮
```

Lagrangian (math): SARAH → Model

$\Downarrow$ (LesHouches.in)

Fortran: SPHENO → Model

$\vdots$ $\vdots$

SARAH Toolbox

Lagrangian (math): SARAH → Model

⇓

Fortran: SPHENO → Model

SPheno.spc

⇓

CalcHEP mdl files: MicrOMEGAS → Model

⇓

UFO files (Python): MadGraph → Model

⋮ ⋮

Problems

Problems

When changing versions …

- Broken HEP tools
- Broken models
- Lack of reproducibility

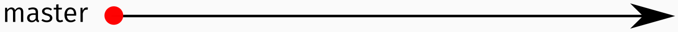
Problems

When changing versions …

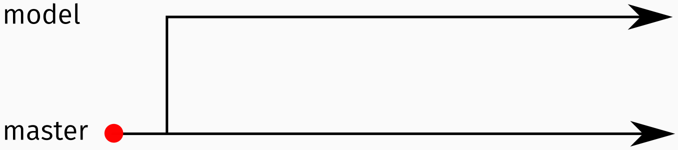
- Broken HEP tools → Version control + Testing
- Broken models → Version control + Testing
- Lack of reproducibility → Testing + Notebooks

Version control

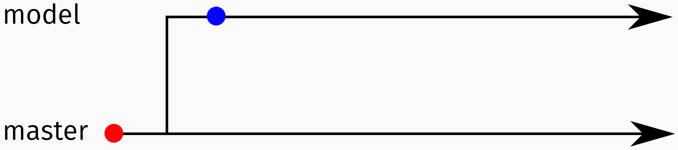
GitHub (public) - GitLab (private)



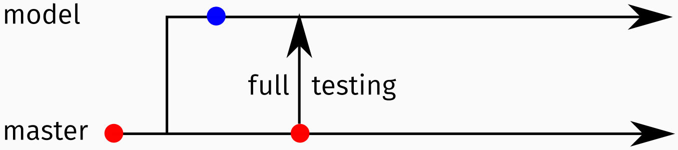
GitHub (public) - GitLab (private)



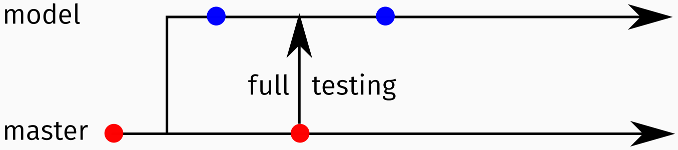
GitHub (public) - GitLab (private)



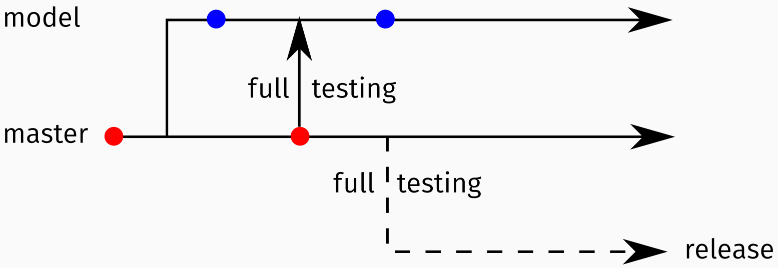
GitHub (public) - GitLab (private)

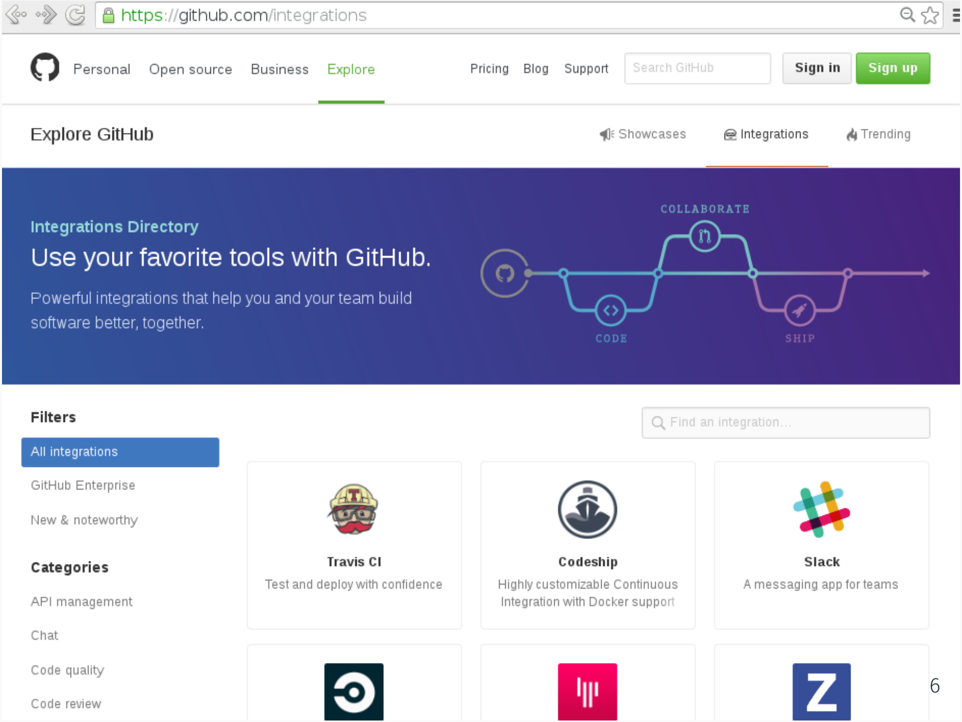


GitHub (public) - GitLab (private)



GitHub (public) - GitLab (private)




https://github.com/integrations
Personal
Open source
Business
Explore
Pricing
Blog
Support
Search GitHub
Sign in
Sign up
Explore GitHub
Showcases
Integrations
Trending
Integrations Directory
Use your favorite tools with GitHub.
Powerful integrations that help you and your team build software better, together.
COLLABORATE
CODE
SHIP
Filters
All integrations
GitHub Enterprise
New & noteworthy
Categories
API management
Chat
Code quality
Code review
Find an integration…
Travis CI
Test and deploy with confidence
Codeship
Highly customizable Continuous Integration with Docker support
Slack
A messaging app for teams

https://github.com/restrepo/BSM-Toolbox

restrepo / **BSM-Toolbox**

Watch 2 | Star 0 | Fork 1

<> Code | Issues 0 | Pull requests 0 | Projects 0 | Pulse | Graphs

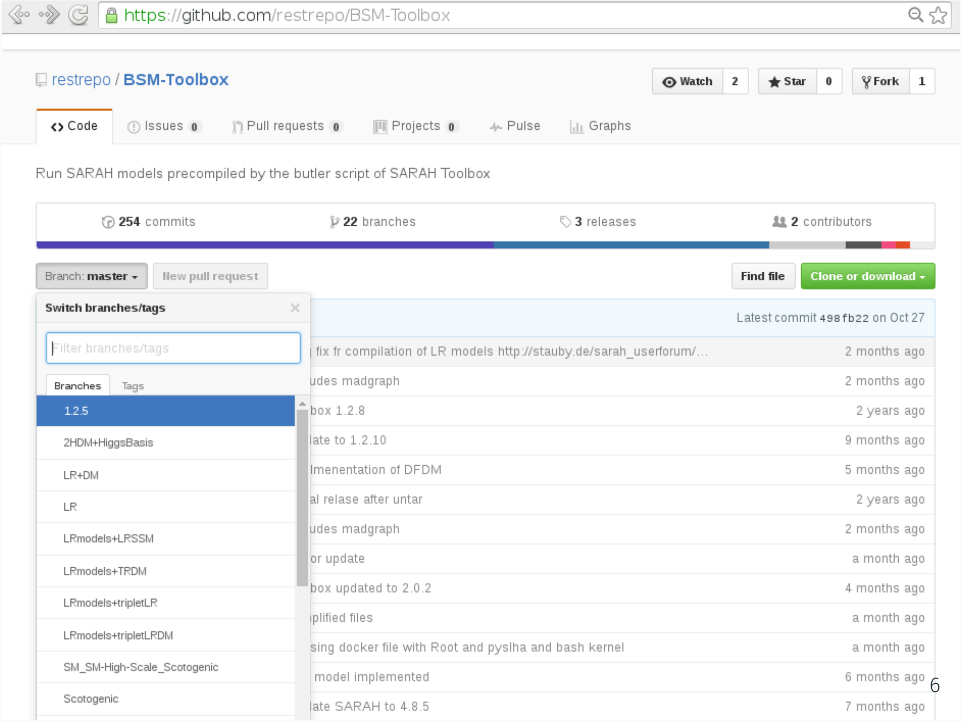
Run SARAH models precompiled by the butler script of SARAH Toolbox

254 commits | **22** branches | **3** releases | **2** contributors

Branch: **master** | New pull request | Find file | Clone or download

restrepo Simplified files — Latest commit 498fb22 on Oct 27

SARAH	Bug fix fr compilation of LR models http://stauby.de/sarah_userforum/...	2 months ago
SPHENO	Includes madgraph	2 months ago
SSP	toolbox 1.2.8	2 years ago
autom4te.cache	Update to 1.2.10	9 months ago
calchep	Implmenentation of DFDM	5 months ago
m4	Initial relase after untar	2 years ago
madgraph	Includes madgraph	2 months ago
micromegas	minor update	a month ago
tarballs	toolbox updated to 2.0.2	4 months ago
tests	Simplified files	a month ago
Dockerfile	Missing docker file with Root and pyslha and bash kernel	a month ago
Dockerfile_bak	Zee model implemented	6 months ago
GIT_MANAGEMENT	Update SARAH to 4.8.5	7 months ago


https://github.com/restrepo/BSM-Toolbox
restrepo / BSM-Toolbox
Watch 2
Star 0
Fork 1
Code
Issues 0
Pull requests 0
Projects 0
Pulse
Graphs
Run SARAH models precompiled by the butler script of SARAH Toolbox
254 commits
22 branches
3 releases
2 contributors
Branch: master
New pull request
Find file
Clone or download
Switch branches/tags
Filter branches/tags
Branches
Tags
1.2.5
2HDM+HiggsBasis
LR+DM
LR
LRmodels+LRSSM
LRmodels+TRDM
LRmodels+tripletLR
LRmodels+tripletLRDM
SM_SM-High-Scale_Scotogenic
Scotogenic
Latest commit 498fb22 on Oct 27
fix fr compilation of LR models http://stauby.de/sarah_userforum/…
2 months ago
udes madgraph
2 months ago
box 1.2.8
2 years ago
late to 1.2.10
9 months ago
lmenentation of DFDM
5 months ago
al relase after untar
2 years ago
udes madgraph
2 months ago
or update
a month ago
box updated to 2.0.2
4 months ago
plified files
a month ago
sing docker file with Root and pyslha and bash kernel
a month ago
model implemented
6 months ago
late SARAH to 4.8.5
7 months ago

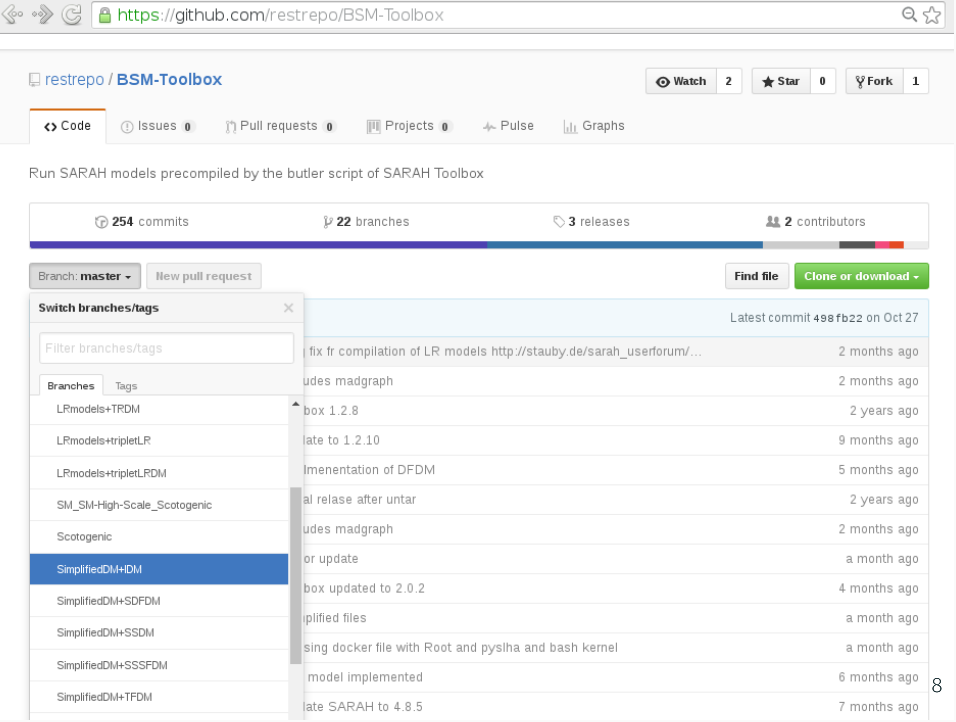
Testing

File Edit View Search Terminal Tabs Help

restr... × [~] :i... × restr... × restr... × restr... × latex... × ksna... × evinc... ×

```
.un nov 28 23:38:48 COT 2016
T13A_2scalars STATUS:PASSED PASSED:59 FAILED:14
=======================DiracNMSSM===================
.un nov 28 23:41:52 COT 2016
DiracNMSSM STATUS:PASSED PASSED:60 FAILED:14
=======================SMSSM===================
.un nov 28 23:52:02 COT 2016
SMSSM STATUS:PASSED PASSED:61 FAILED:14
=======================Omega_Short/Regime-2===================
nar nov 29 00:02:54 COT 2016
Omega_Short/Regime-2 STATUS:PASSED PASSED:62 FAILED:14
=======================Omega_Short/Regime-1===================
nar nov 29 03:18:25 COT 2016
Omega_Short/Regime-1 STATUS:PASSED PASSED:63 FAILED:14
=======================Inert===================
nar nov 29 03:19:11 COT 2016
Inert STATUS:PASSED PASSED:64 FAILED:14
=======================IItripletLR===================
nar nov 29 03:22:42 COT 2016
IItripletLR STATUS:PASSED PASSED:65 FAILED:14
=======================kk===================
nar nov 29 03:40:14 COT 2016
kk STATUS:FAILED PASSED:65 FAILED:15
=======================TMSSM===================
nar nov 29 03:42:14 COT 2016
TMSSM STATUS:PASSED PASSED:66 FAILED:15
=======================inverse-Seesaw-NMSSM===================
nar nov 29 03:53:15 COT 2016
inverse-Seesaw-NMSSM STATUS:PASSED PASSED:67 FAILED:15
=======================4VL===================
nar nov 29 04:08:52 COT 2016
4VL STATUS:PASSED PASSED:68 FAILED:15
=======================U1xMSSM3G/Vevacious===================
nar nov 29 04:12:19 COT 2016
```

0 bash 1 bash 2 bash 3 bash fisica 11/29 6:11am


https://github.com/restrepo/BSM-Toolbox
restrepo / BSM-Toolbox
Watch 2
Star 0
Fork 1
Code
Issues 0
Pull requests 0
Projects 0
Pulse
Graphs
Run SARAH models precompiled by the butler script of SARAH Toolbox
254 commits
22 branches
3 releases
2 contributors
Branch: master
New pull request
Find file
Clone or download
Switch branches/tags
Filter branches/tags
Branches
Tags
LRmodels+TRDM
LRmodels+tripletLR
LRmodels+tripletLRDM
SM_SM-High-Scale_Scotogenic
Scotogenic
SimplifiedDM+IDM
SimplifiedDM+SDFDM
SimplifiedDM+SSDM
SimplifiedDM+SSSFDM
SimplifiedDM+TFDM
Latest commit 498fb22 on Oct 27
fix fr compilation of LR models http://stauby.de/sarah_userforum/…
2 months ago
2 months ago
2 years ago
9 months ago
5 months ago
2 years ago
2 months ago
a month ago
4 months ago
a month ago
a month ago
6 months ago
7 months ago

https://github.com/restrepo/BSM-Toolbox/tree/SimplifiedDM+IDM

README_CREATES_NEW_MODE…	small fix	5 months ago
README_deploy_mybinder.md	small fix	5 months ago
README_mybinder	partial update	5 months ago
aclocal.m4	Initial relase after untar	2 years ago
butler	toolbox updated to 2.0.2	4 months ago
butler.in	toolbox updated to 2.0.2	4 months ago
butler.patch	1.29	2 years ago
clean.sh	Clean repo	5 months ago
clean_repo.sh	prepare merging	4 months ago
compile_spheno_directly.sh	Introdocing scalar singlet to have also a compiling micromegas.	a year ago
config.status	toolbox-1.2.7	2 years ago
configure	toolbox updated to 2.0.2	4 months ago
configure.ac	Update to 1.2.10	9 months ago
gitconfig.sh	small fix	5 months ago
index.ipynb	Fixed Model name	5 months ago
index_bash.ipynb	fix micromegas bash run	4 months ago
micromegas_prtcl_fix.sh	Reorder particles in prtcl1.mdl	5 months ago
requirements.txt	Zee model implemented	6 months ago
update-dirs.sh	1.2.9	2 years ago

README.md

Run SARAH models precompiled by the butler script of SARAH Toolbox.

SARAH Toolbox

Collection of models to be run from a docker image

Launch virtual docker image:

launch binder

Jupyter home (Files, New -> Terminal, etc)

Highly recommended:

Run from a terminal:

Implemented models

Each model is to be run in a specific virtual machine. Follow the binder button in each github repo

- `SARAH/Models/SSDM/` (for test purposes)
- `SARAH/Models/SM/HighScale` (For RGE running: ./butler SM/HighScale) Repo
- `SARAH/Models/SimplifiedDM/IDM` (This image)

Instructions to compile the model

In SPHENO and `micromegas`

See possible analysis based on the models in ./tests folder.

Below we define the model to be compiled:

```
In [1]: MODEL=SimplifiedDMIDM
```

mybinder.org/repo/restrepo/SimplifiedDM-IDM-Toolbox

binder

loading

restrepo/SimplifiedDM-IDM-Toolbox

104.197.121.217/user/cd5a815737ba430c36dab5b848d7e7a4/notebooks/index_bash.ipynb

jupyter index_bash (unsaved changes)

File Edit View Insert Cell Kernel Help | Bash

Markdown CellToolbar

SARAH Toolbox

Collection of models to be run from a docker image

Launch virtual docker image:

launch binder

Jupyter home (Files, New -> Terminal, etc)

Highly recommended:

Run from a terminal:

Implemented models

Each model is to be run in a specific virtual machine. Follow the binder button in each github repo

- `SARAH/Models/SSDM/` (for test purposes)
- `SARAH/Models/SM/HighScale` (For RGE running: ./butler SM/HighScale) Repo
- `SARAH/Models/SimplifiedDM/IDM` (This image)

104.197.121.217/user/cd5a815737ba430c36dab5b848d7e7a4/notebooks/index_bash.ipynb

File Edit View Insert Cell Kernel Help | Bash

Instructions to compile the model

In SPHENO and mic romegas

See possible analysis based on the models in ./tests folder.

Below we define the model to be compiled:

In [1]:
```
MODEL=SimplifiedDMIDM
```

To better control in the outputs, it is recomended that the commands be executed from a terminal. Only possible errors are to be shown below

In [2]:
```
./compile_spheno_directly.sh $MODEL > /dev/null
```

make a work dir

In [3]:
```
mkdir -p test_compilation
cd test_compilation
```

In [4]:
```
LHAINPUT=../SPHENO/$MODEL/Input_Files/LesHouches.in.$MODEL
if [ -f ../SARAH/Models/$MODEL/LesHouches.in.$MODEL ]; then
   LHAINPUT=../SARAH/Models/$MODEL/LesHouches.in.$MODEL
fi
```

Run SPheno and generate LHA output file

In [5]:
```
cp $LHAINPUT  #input
```

104.197.121.217/user/cd5a815737ba430c36dab5b848d7e7a4/notebooks/index_bash.ipynb

File Edit View Insert Cell Kernel Help Bash

```
CDM-nucleon micrOMEGAs amplitudes:
proton:  SI  -4.458E-12  SD  0.000E+00
neutron: SI  -4.502E-12  SD  0.000E+00
         BOX DIAGRAMS
CDM-nucleon micrOMEGAs amplitudes:
proton:  SI  -4.458E-12  SD  0.000E+00
neutron: SI  -4.502E-12  SD  0.000E+00
CDM-nucleon cross sections[pb]:
 proton  SI 7.259E-15  SD 0.000E+00
 neutron SI 7.404E-15  SD 0.000E+00

======== Direct Detection ========
73Ge: Total number of events=2.74E-09 /day/kg
Number of events in 10 - 50 KeV region=6.72E-12 /day/kg
131Xe: Total number of events=5.56E-09 /day/kg
Number of events in 10 - 50 KeV region=1.30E-14 /day/kg
23Na: Total number of events=5.36E-10 /day/kg
Number of events in 10 - 50 KeV region=3.36E-11 /day/kg
I127: Total number of events=5.36E-09 /day/kg
Number of events in 10 - 50 KeV region=3.20E-14 /day/kg
```

Go now to:

main ipyrhon file

Or to some specific scan in: Lux2016 notebook

104.197.121.217/user/cd5a815737ba430c36dab5b848d7e7a4/notebooks/tests/SimplifiedDM-IDM.

File Edit View Insert Cell Kernel Help | Python 2

Inert doublet model

According to this bug report, we need to change by hand the file prtcls1.mdl of the micromegas model files, to be sure that the DM candidate appears as the first defined Z_2-particle. In this case:

```
etR       |~etR    |~etR    |35 ...
etI       |~etI    |~etI    |36 ...
etp       |~etp    |~Etp    |37 ...
```

In [1]:
```
%pylab inline
```

Populating the interactive namespace from numpy and matplotlib

In [2]:
```
import pandas as pd
import numpy as np
import os, sys, inspect
import commands
from hep import *
```

Define functions to change from general basis to physical basis

In [3]:
```
def run_official_idm(MHX,MH3,MHC,laL,la2,Mh,check=False):
    pd.Series({'MHX':MHX,'MH3':MH3,'MHC':MHC,'laL':laL,'la2':la2,'Mh':Mh}).to_csv('
    omegah2=-1
    if os.path.isfile('../micromegas/IDM/main'):
        mo=commands.getoutput("../micromegas/IDM/main mo.dat")

    return mo

def phys_to_int(mH,mA,mHc,lambda_L,v):
```

104.197.121.217/user/cd5a815737ba430c36dab5b848d7e7a4/notebooks/tests/SimplifiedDM-IDM.

File Edit View Insert Cell Kernel Help | Python 2

Inert doublet model

According to this bug report, we need to change by hand the file prtcls1.mdl of the `micromegas` model files, to be sure that the DM candidate appears as the first defined Z_2-particle. In this case:

```
etR        |~etR    |~etR    |35 ...
etI        |~etI    |~etI    |36 ...
etp        |~etp    |~Etp    |37 ...
```

In [1]:
```
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [2]:
```
import pandas as pd
import numpy as np
import os, sys, inspect
import commands
from hep import *
```

Define functions to change from general basis to physical basis

In [3]:
```
def run_official_idm(MHX,MH3,MHC,laL,la2,Mh,check=False):
    pd.Series({'MHX':MHX,'MH3':MH3,'MHC':MHC,'laL':laL,'la2':la2,'Mh':Mh}).to_csv('
    omegah2=-1
    if os.path.isfile('../micromegas/IDM/main'):
        mo=commands.getoutput("../micromegas/IDM/main mo.dat")

    return mo

def phys_to_int(mH,mA,mHc,lambda_L,v):
```

104.197.121.217/user/cd5a815737ba430c36dab5b848d7e7a4/notebooks/tests/SimplifiedDM-IDM.

File Edit View Insert Cell Kernel Help Python 2

Inert doublet model

According to this bug report, we need to change by hand the file prtcls1.mdl of the micromegas model files, to be sure that the DM candidate appears as the first defined Z_2-particle. In this case:

```
etR        |~etR    |~etR    |35 ...
etI        |~etI    |~etI    |36 ...
etp        |~etp    |~Etp    |37 ...
```

In [1]:
```
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [2]:
```
import pandas as pd
import numpy as np
import os, sys, inspect
import commands
from hep import *
```

Define functions to change from general basis to physical basis

In [3]:
```
def run_official_idm(MHX,MH3,MHC,laL,la2,Mh,check=False):
    pd.Series({'MHX':MHX,'MH3':MH3,'MHC':MHC,'laL':laL,'la2':la2,'Mh':Mh}).to_csv('
    omegah2=-1
    if os.path.isfile('../micromegas/IDM/main'):
        mo=commands.getoutput("../micromegas/IDM/main mo.dat")

    return mo

def phys_to_int(mH,mA,mHc,lambda_L,v):
```

Check one point

With SARAH implementation

Based in Scotogenic model implementation by Avelino Vicente. Model files in the SARAH/Models/SimplifiedDM/IDM folder of this repo. We use below the python hep module to automalically manage input/output SARAH-Toolbox files (in a similar way to SSP)

In [6]:
```
a=hep(MODEL='SimplifiedDMIDM')
```

`a-object` is an object with many attributes and methods. Use the tab to explore them. Some of them are

- a.Series: pandas Series object with the "relevant" variables
- a.LHA: Input LesHouces file as pyslha object
- a.runSPheno() -> a.LHA_out: return LHA output files as pyslha object
- a.runmicromegas() -> a.runSPheno() -> Updated the `a-object` with micrOMEGAS "relevant" output

Benchmark BP1

from arXiv:1504.05949. See also: arXiv:1207.0084

In [8]:
```
v=a.vev
#lambda_1=0.13
ipt=pd.Series({'MHX':66,'MH3':300,'MHC':300,'lambda_L':0.0107}) #Official micromega
mu2,lambda_3,lambda_4,lambda_5=phys_to_int(ipt.MHX,ipt.MH3,ipt.MHC,ipt.lambda_L,v)
print 'expected:',ipt.MHX,ipt.MH3,ipt.MHC
print 'obtained:',int_to_phys(mu2,lambda_3,lambda_4,lambda_5,v)[0]
```

104.197.121.217/user/cd5a815737ba430c36dab5b848d7e7a4/notebooks/tests/SimplifiedDM-IDM.

File Edit View Insert Cell Kernel Help | Python 2

Benchmark BP1

from arXiv:1504.05949. See also: arXiv:1207.0084

In [8]:

```
v=a.vev
#lambda_1=0.13
ipt=pd.Series({'MHX':66,'MH3':300,'MHC':300,'lambda_L':0.0107}) #Official micromega
mu2,lambda_3,lambda_4,lambda_5=phys_to_int(ipt.MHX,ipt.MH3,ipt.MHC,ipt.lambda_L,v)
print 'expected:',ipt.MHX,ipt.MH3,ipt.MHC
print 'obtained:',int_to_phys(mu2,lambda_3,lambda_4,lambda_5,v)[0]
devnull=commands.getoutput('rm -f SPheno.spc.%s' %a.MODEL)
a.LHA.blocks['SPHENOINPUT'].entries[55]='0            # Calculate one loop masse
a.LHA.blocks['MINPAR'][3]='%0.8E       #lambda3Input' %lambda_3
a.LHA.blocks['MINPAR'][4]='%0.8E       #lambda4Input' %lambda_4
a.LHA.blocks['MINPAR'][5]='%0.8E       #lambda5Input' %lambda_5
a.LHA.blocks['MINPAR'][6]='%0.8E       #mEt2Input' %mu2
moc=a.runmicromegas(Direct_Detection=True)
print 'Omega h^2, SI proton, neutron =',a.Series.Omega_h2,a.Series.proton_SI,a.Seri
print 'mu2,lambda_3,lambda_4,lambda_5',np.sqrt(mu2),lambda_3,lambda_4,lambda_5,(lam
```

```
expected: 66.0 300.0 300.0
obtained: [  66.   300.   300.]
Omega h^2, SI proton, neutron = 0.108 8.853e-10 9.03e-10
mu2,lambda_3,lambda_4,lambda_5 60.8877419158 2.84678917566 -1.41269458783
-1.41269458783 0.0107
```

See full `LesHouches.in.SimplifiedDMIDM` and `SPheno.spc.SimplifiedDMIDM` in **Appendix 1**

BP1 at one loop

Scan m_{H^0}

For the next two plots we fix:

- $m_h = 126$ GeV
- $m_{A^0} = 701$ GeV
- $m_{H^+} = 701$ GeV
- $\lambda_L = 0.1$

And vary

- $40 < m_{H^0}/\text{GeV} < 700$

In [11]:

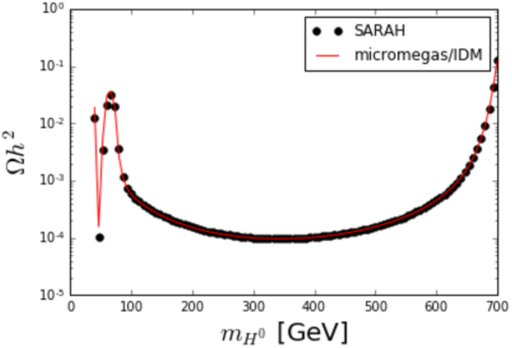
```
df=pd.DataFrame()
ipt=pd.Series({'MHX':40,'MH3':701,'MHC':701,'lambda_L':0.1})
a.LHA.blocks['SPHENOINPUT'].entries[55]='0                # Calculate one loop masse
dm_masses=np.linspace(40,700,100)
for MHX in dm_masses:
    if np.where(dm_masses==MHX)[0][0]%10==0: #find the index of the array entry
        print np.where(dm_masses==MHX)[0][0]
    ipt.MHX=MHX
    mu2,lambda_3,lambda_4,lambda_5=phys_to_int(ipt.MHX,ipt.MH3,ipt.MHC,ipt.lambda_L
    a.LHA.blocks['MINPAR'][5]='%0.8E       #lambda5Input' %lambda_5
    a.LHA.blocks['MINPAR'][3]='%0.8E       #lambda4Input' %lambda_3
    a.LHA.blocks['MINPAR'][4]='%0.8E       #lambda3Input' %lambda_4
    a.LHA.blocks['MINPAR'][6]='%0.8E       #mEt2Input' %mu2
    a.runmicromegas(Direct_Detection=True)
    a.Series=a.Series.append(ipt)
    a.Series=a.Series.append(pd.Series({'MH0':a.LHA_out.blocks['MASS'][35],\
                                        'MA0':a.LHA_out.blocks['MASS'][36],\
                                        'MHc':a.LHA_out.blocks['MASS'][37]}))
```

104.197.121.217/user/cd5a815737ba430c36dab5b848d7e7a4/notebooks/tests/SimplifiedDM-IDM.

File Edit View Insert Cell Kernel Help Python 2

Relic density

In [12]:
```
dfm=df[df.MH0<df.MHc]
plt.semilogy(dfm.MH0,dfm.Omega_h2,'ko',label='SARAH')
plt.semilogy(dfm.MH0,dfm.Omega_h2_official,'r-',label='micromegas/IDM'
plt.xlabel(r'$m_{H^0}$ [GeV]',size=20)
plt.ylabel(r'$\Omega h^2$',size=20)
plt.legend(loc='best')
plt.savefig('omega.pdf')
```

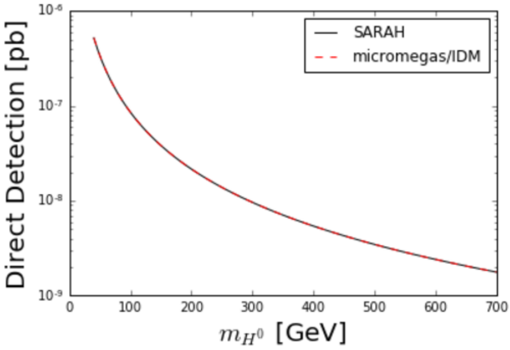


104.197.121.217/user/cd5a815737ba430c36dab5b848d7e7a4/notebooks/tests/SimplifiedDM-IDM.

File Edit View Insert Cell Kernel Help | Python 2

m_{H^0} [GeV]

In [13]:
```
dfm=df[df.MH0<df.MHc]
plt.semilogy(dfm.MH0,dfm.proton_SI,'k-',label='SARAH')
plt.semilogy(dfm.MH0,dfm.proton_SI_official,'r--',label='micromegas/ID
plt.xlabel(r'$m_{H^0}$ [GeV]',size=20)
plt.ylabel(r'Direct Detection [pb]' ,size=20)
plt.legend(loc='best')
plt.savefig('dd.pdf')
```



In [14]:
```
plt.plot(dfm.MH0,dfm.proton_SI_official/dfm.proton_SI,'k-')
plt.xlabel(r'$m_{H^0}$ [GeV]',size=20)
```

Reproducibility

localhost:8888/notebooks/prog/toolbox/mybinders/SimplifiedDM/SSSFDM/tests/SimplifiedDM-SSSFD

File Edit View Insert Cell Kernel Help Python 2

Singlet Scalar Singlet (charged) Fermion dark matter model

SSSFDM

- [arXiv:1307.6181]
- [arXiv:1307.6480]

Particle content

In [61]:

```
%%latex
\begin{array}{llllll}
\text{Name} & \text{Symbol} & \text{SU}(3)_c & \text{SU}(2)_L & \text{U}(1)_
 \begin{pmatrix}\nu_L & e_L\end{pmatrix}^{\operatorname{T}}&
\begin{pmatrix}\xi_{1\alpha} & \xi_{2\alpha} \end{pmatrix}^{\operatorname{T
(e_R)^{\dagger} & \eta^{\alpha}_1 & \mathbf{1} & \mathbf{1} & +1 & +1 \\
(\psi_R)^{\dagger} & \eta^{\alpha}_2 & \mathbf{1} & \mathbf{1} & +1 & -1 \\
\psi_L & \xi_{3\alpha} & \mathbf{1} & \mathbf{1} & -1 & -1 \\
\end{array}
```

Name	Symbol	$\text{SU}(3)_c$	$\text{SU}(2)_L$	$\text{U}(1)_Y$	Z_2
$\begin{pmatrix}\nu_L & e_L\end{pmatrix}^{\text{T}}$	$\begin{pmatrix}\xi_{1\alpha} & \xi_{2\alpha}\end{pmatrix}^{\text{T}}$	$\mathbf{1}$	$\mathbf{2}$	$-1/2$	$+1$
$(e_R)^{\dagger}$	η^{α}_1	$\mathbf{1}$	$\mathbf{1}$	$+1$	$+1$
$(\psi_R)^{\dagger}$	η^{α}_2	$\mathbf{1}$	$\mathbf{1}$	$+1$	-1
ψ_L	$\xi_{3\alpha}$	$\mathbf{1}$	$\mathbf{1}$	-1	-1

After the spontaneous symmetry breaking, the relevant Yukawa terms are

$$\begin{aligned}
\mathcal{L} =& \frac{h_e v}{\sqrt{2}}\left(\eta_1^\alpha \xi_{2\alpha} + \xi_{2\dot\alpha}^\dagger \eta_1^{\dagger\dot\alpha}\right) + M_\psi \left(\eta_2^\alpha \xi_{3\alpha} + \xi_{3\dot\alpha}^\dagger \eta_2^{\dagger\dot\alpha}\right) + h_S \left(S\eta_1^\alpha \xi_{3\alpha} + S\xi_{3\dot\alpha}^\dagger \eta_1^{\dagger\dot\alpha}\right)\\
=& \frac{h_e v}{\sqrt{2}}\begin{pmatrix}\eta_1^\alpha & \xi_{2\dot\alpha}^\dagger\end{pmatrix}\begin{pmatrix}\xi_{2\alpha}\\ \eta_1^{\dagger\dot\alpha}\end{pmatrix} + M_\psi \begin{pmatrix}\eta_2^\alpha & \xi_{3\dot\alpha}^\dagger\end{pmatrix}\begin{pmatrix}\xi_{3\alpha}\\ \eta_2^{\dagger\dot\alpha}\end{pmatrix}\\
&+ \left[h_S S \begin{pmatrix}\eta_1^\alpha & \xi_{2\dot\alpha}^\dagger\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}\xi_{3\alpha}\\ \eta_2^{\dagger\dot\alpha}\end{pmatrix} + \text{h.c}\right]\\
=& \frac{h_e v}{\sqrt{2}}\begin{pmatrix}\xi_{2\dot\alpha}^\dagger & \eta_1^\alpha\end{pmatrix}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}\xi_{2\alpha}\\ \eta_1^{\dagger\dot\alpha}\end{pmatrix} + M_\psi \begin{pmatrix}\xi_{3\dot\alpha}^\dagger & \eta_2^\alpha\end{pmatrix}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}\xi_{3\alpha}\\ \eta_2^{\dagger\dot\alpha}\end{pmatrix}\\
&+ \left[h_S S \begin{pmatrix}\xi_{2\dot\alpha}^\dagger & \eta_1^\alpha\end{pmatrix}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}\xi_{3\alpha}\\ \eta_2^{\dagger\dot\alpha}\end{pmatrix} + \text{h.c}\right]
\end{aligned}$$

Defining

$$e = \begin{pmatrix}\xi_{2\alpha}\\ \eta_1^{\dagger\dot\alpha}\end{pmatrix} = \begin{pmatrix}e_L\\ e_R\end{pmatrix} \qquad \Psi = \begin{pmatrix}\xi_{3\alpha}\\ \eta_2^{\dagger\dot\alpha}\end{pmatrix} = \begin{pmatrix}\psi_L\\ \psi_R\end{pmatrix}$$

the relevant Yukawa Lagrangian in terms of Dirac fermions is

$$\begin{aligned}
\mathcal{L} =& \frac{h_e v}{\sqrt{2}} e^\dagger \gamma^0 e + M_\psi \Psi^\dagger \gamma^0 \Psi + \left(h_S S e^\dagger \gamma^0 P_L \Psi + \text{h.c}\right)\\
=& \frac{h_e v}{\sqrt{2}} \bar{e} e + M_\psi \overline{\Psi}\Psi + h_S \left(S\,\bar{e}\Psi_L + \text{h.c}\right)\,,
\end{aligned}$$

where

$$\Psi_L = P_L \Psi = \begin{pmatrix}\psi_L\\ 0\end{pmatrix}\,.$$

localhost:8888/notebooks/prog/toolbox/mybinders/SimplifiedDM/SSSFDM/tests/SimplifiedDM-SSSFD

File Edit View Insert Cell Kernel Help Python 2

In [1]:
```
%pylab inline
```
```
Populating the interactive namespace from numpy and matplotlib
```

In [2]:
```
import pandas as pd
import numpy as np
from matplotlib.colors import LogNorm
import os, sys, inspect
import commands
from hep import *
```

Check one point

In [3]:
```
a=hep(MODEL='SimplifiedDMSSSFDM')
```

`a-object` is an object with many attributes and methods. Use the tab to explore them. Some of them are

- a.Series: pandas Series object with the "relevant" variables
- a.LHA: Input LesHouces file as pyslha object
- a.runSPheno() -> a.LHA_out: return LHA output files as pyslha object
- a.runmicromegas() -> a.runSPheno() -> Updated the `a-object` with micrOMEGAS "relevant" output

In [4]:
```
pd.Series(a.LHA.blocks['MINPAR'].entries)
```

Out[4]:
```
1     2.5500000E-01   # Lambda1IN
2       0.0000000E+00   # LamSHIN
3        0.0000000E+00   # LamSIN
4     2.0000000E+02   # MS2Input
5        2.0000000E+02   # MSFIN
```

localhost:8888/notebooks/prog/toolbox/mybinders/SimplifiedDM/SSSFDM/tests/SimplifiedDM-SSSFD

File Edit View Insert Cell Kernel Help Python 2

```
dtype: object
```

In [5]:

```
v=a.vev
lambda_1=0.26
lambda_SH=0.
MS=150**2
MF=200
Yse=1.9
Ymu=0
Ytau=0
devnull=commands.getoutput('rm -f SPheno.spc.%s' %a.MODEL)
a.LHA.blocks['SPHENOINPUT'].entries[55]='0              # Calculate one lo
a.LHA.blocks['SPHENOINPUT'].entries[520]='0.            # Write effective H
a.LHA.blocks['MINPAR'][1]='%0.8E       # lambda1'  %lambda_1
a.LHA.blocks['MINPAR'][2]='%0.8E       # lambdaSH'  %lambda_SH
a.LHA.blocks['MINPAR'][4]='%0.8E       # MS'  %MS
a.LHA.blocks['MINPAR'][5]='%0.8E       # MF' %MF
a.LHA.blocks['YSIN'][1]='%0.8E       # Ys(1)' %Yse
a.LHA.blocks['YSIN'][2]='%0.8E       # Ys(1)' %Ymu
a.LHA.blocks['YSIN'][3]='%0.8E       # Ys(3)' %Ytau
moc=a.runmicromegas(Direct_Detection=True)
print 'Omega h^2, SI proton, neutron =',a.Series.Omega_h2,a.Series.proton_S
```

```
Omega h^2, SI proton, neutron = 0.111 0.0 0.0
```

Scan the parameter space

In [ ]:

```
import time
st=time.time()
a=hep(MODEL='SimplifiedDMSSSFDM')
v=a.vev
Omega_h2_delta=0.0022
```

localhost:8888/notebooks/prog/toolbox/mybinders/SimplifiedDM/SSSFDM/tests/SimplifiedDM-SSSFD

File Edit View Insert Cell Kernel Help | Python 2

Scan the parameter space

In [ ]:

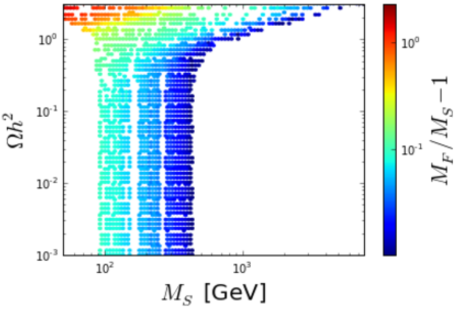
```
import time
st=time.time()
a=hep(MODEL='SimplifiedDMSSSFDM')
v=a.vev
Omega_h2_delta=0.0022
CL=3
Omega_h2=0.1197
Omega_h2_exp=[Omega_h2-CL*Omega_h2_delta,Omega_h2,Omega_h2+CL*Omega_h2_delt
lambda_1=0.26
lambda_SH=0
a.LHA.blocks['SPHENOINPUT'].entries[55]='0              # Calculate one lo
a.LHA.blocks['SPHENOINPUT'].entries[520]='0.           # Write effective H

df=pd.DataFrame()
a.LHA.blocks['SPHENOINPUT'].entries[55]='0              # Calculate one lo
dfmin=100 #40
dfmax=600 #1E4
npoints=1000
df_masses=np.logspace(np.log10(dfmin),np.log10(dfmax),npoints) #np.array([2
DEBUG=False
for i in range(1):
    for MF in df_masses:
        rm1=10**np.random.uniform(np.log10(1E-2),np.log10(3))
        r=rm1+1.
        M_S=MF/r
        MS2=M_S**2-a.vev**2*lambda_SH
        Yse_range=np.logspace(np.log10(np.pi),np.log10(1E-3),200)
        Omega_h2_old=1E32
        for Yse in Yse_range:
            devnull=commands.getoutput('rm -f SPheno.spc.%s' %a.MODEL)
            Ymu=0. #10**np.random.uniform(.log10(1E-3),np.log10(np.pi),)
```

localhost:8888/notebooks/prog/toolbox/mybinders/SimplifiedDM/SSSFDM/tests/SimplifiedDM-SSSFD

File Edit View Insert Cell Kernel Help Python 2

In [34]:
```
df=df[df.MF>=100]
plt.hexbin(df.ss,df.Ys1,df.MF/df.ss-1,xscale='log',yscale='log',norm=LogNorn
cl=plt.colorbar()
cl.set_label(r'$M_F/M_S-1$',size=20)
plt.xlim(50,df.MF.max())
plt.xlabel(r'$M_S$ [GeV]',size=20)
plt.ylabel(r'$\Omega h^2$',size=20)
```

Out[34]: <matplotlib.text.Text at 0x7fc18defe5d0>



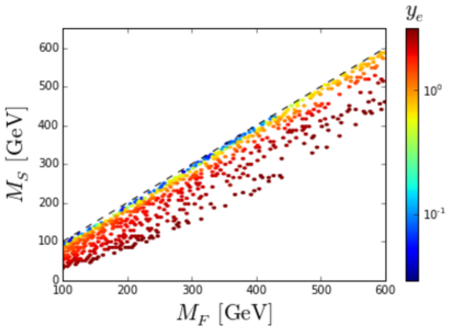
In [20]:
```
MF_max=600
plt.hexbin(df[df.MF<MF_max].MF,df[df.MF<MF_max].ss,df[df.MF<MF_max].Ys1,norn
cb=plt.colorbar()
x=np.linspace(100,1000,10)
plt.plot(x,x,'k--')
```

localhost:8888/notebooks/prog/toolbox/mybinders/SimplifiedDM/SSSFDM/tests/SimplifiedDM-SSSFD

File Edit View Insert Cell Kernel Help Python 2

In [20]:

```
MF_max=600
plt.hexbin(df[df.MF<MF_max].MF,df[df.MF<MF_max].ss,df[df.MF<MF_max].Ys1,norm
cb=plt.colorbar()
x=np.linspace(100,1000,10)
plt.plot(x,x,'k--')

plt.xlabel(r'$M_F\ {\rm [GeV]}$',size=20)
plt.ylabel(r'$M_S\ {\rm [GeV]}$',size=20)
plt.text(630,680,r'$y_e$',size=20)
plt.xlim(100,600)
plt.ylim(0,650)

plt.tight_layout()
#plt.savefig('singlet_exc.pdf')
```



localhost:8888/notebooks/prog/toolbox/mybinders/SimplifiedDM/SSSFDM/tests/SimplifiedDM-SSSFD

File Edit View Insert Cell Kernel Help Python 2

In [7]:
```python
from madgraph import *
```

In [10]:
```python
generate_cross_section(a.MODEL,processes=['generate p p > fre frebar'],sqrts
```

In [11]:
```python
launch_cross_section(a.MODEL)
```

Out[11]:
```
(0.02713, 5e-05)
```

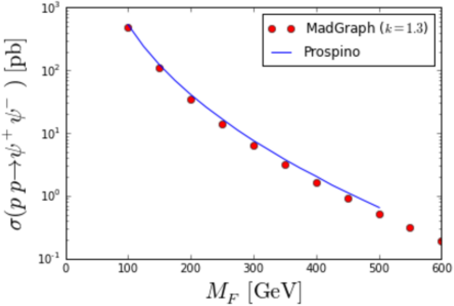
In [12]:
```python
from madgraph import *
a=hep(MODEL='SimplifiedDMSSSFDM')

generate_cross_section(a.MODEL,processes=['generate p p > fre frebar'],sqrts
v=a.vev
lambda_1=0.26
lambda_SH=0
a.LHA.blocks['SPHENOINPUT'].entries[55]='0              # Calculate one loo
a.LHA.blocks['SPHENOINPUT'].entries[520]='0.            # Write effective H
df=pd.DataFrame()
dfmin=100 #40
dfmax=600 #1E4
npoints=11
df_masses=np.linspace(dfmin,dfmax,npoints) #np.array([200]) 1E-4
for MF in df_masses:
        M_S=20
        MS2=M_S**2-a.vev**2*lambda_SH
        Yse=1.
        devnull=commands.getoutput('rm -f SPheno.spc.%s' %a.MODEL)
        Ymu=0. #10**np.random.uniform( log10(1E-3),np.log10(np.pi) )
        Ytau=0. #10
        a.LHA.blocks['MINPAR'][1]='%0.8E       # lambda1'  %lambda_1
        a.LHA.blocks['MINPAR'][2]='%0.8E       # lambdaSH'  %lambda_SH
        a.LHA.blocks['MINPAR'][4]='%0.8E       # MS2'  %MS2
        a.LHA.blocks['MINPAR'][5]='%0.8E       # MF' %MF
```

localhost:8888/notebooks/prog/toolbox/mybinders/SimplifiedDM/SSSFDM/tests/SimplifiedDM-SSSFD

File Edit View Insert Cell Kernel Help | Python 2

In [19]:
```
plt.semilogy(df.MF,df.cs*1000*1.3,'ro',label='MadGraph ($k=1.3$)')
plt.semilogy(pr.mc1,pr.cs,label='Prospino')
plt.xlabel(r'$M_F\ \rm{[GeV]}$',size=20)
plt.ylabel(r'$\sigma(p\ p\to \psi^+\psi^-)\ \rm{[pb]}$',size=20)
plt.legend(loc='best')
```

Out[19]:
```
<matplotlib.legend.Legend at 0x7f4243c7b290>
```



Results in Poster Session:

Marta Liliana Sánchez Pélaez

Conclusion

Sophisticated HEP tools require high programming standars

Check: CERN-Root development