Computer approach to study particle Dark Matter

José Halim Montes de Oca Yemha
Departamento de física
FES-Cuautitlán UNAM

6/12/2019

4th ComHEP: Colombian Meeting on High Energy Physics (Barranquilla, Colombia)

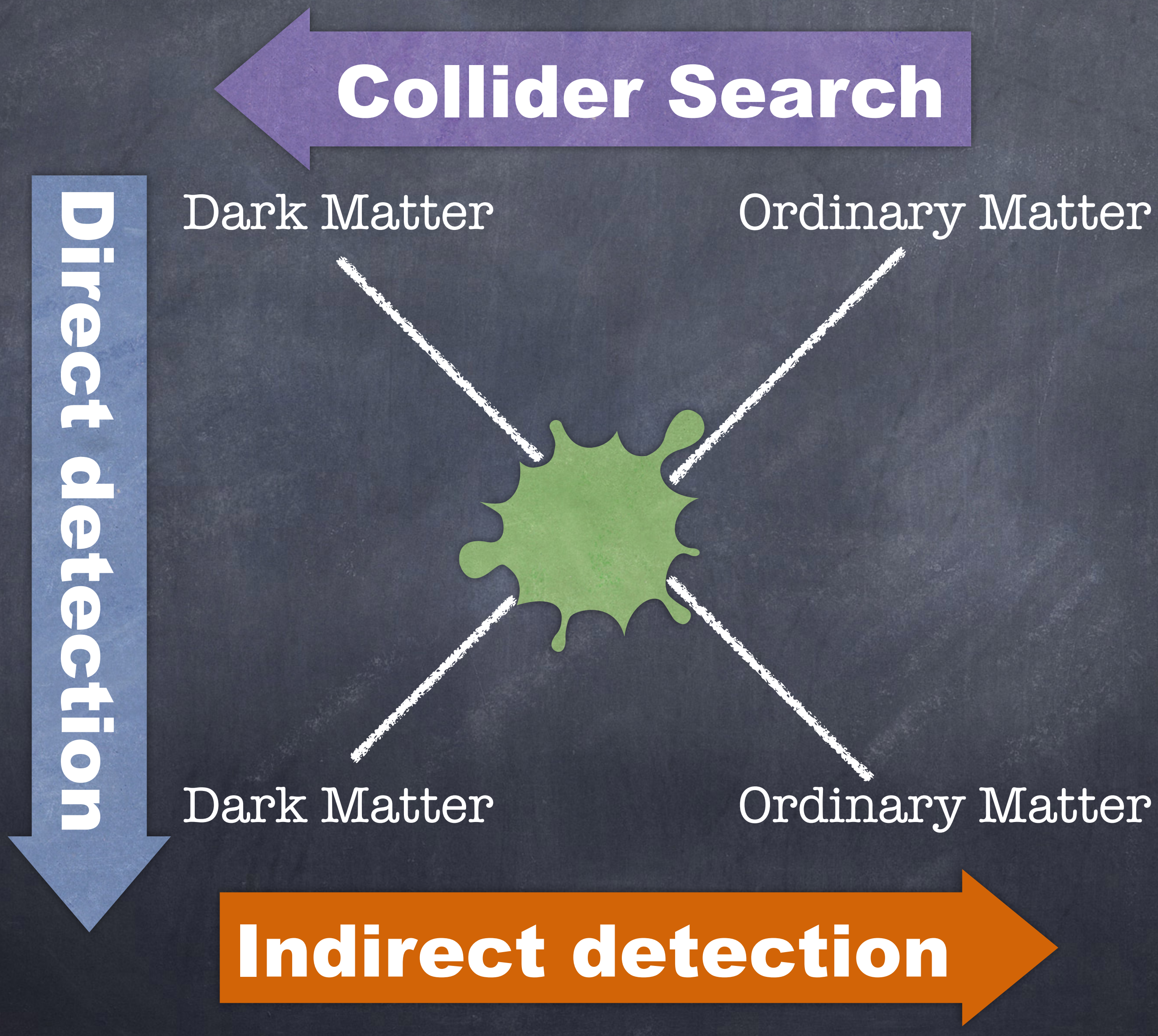
M.A. Arroyo-Ureña, R. Gaitan, R. Martinez, J.H. Montes de Oca.
e-Print: arXiv:1907.08231 [hep-ph]

This work was supported by PAPIIT with registration codes IA107118 and IN115319 in DGAPA-UNAM, and SNI in CONACyT

Outline

- Motivation
- LanHEP
- MicrOMEGAs
- Summary

Searches for Dark matter



Computer Tools

Model (Lagrangian)

* LanHEP[1] [2]
* FeynRules
* Sarah[3]

- MicrOmegas (CalcHEP)[5]
- MadDM (MadGraph)[6]
- FeynCalc and Feynarts

Dark Matter observables

1 theory.sinp.msu.ru/~semenov/lanhep.html

2 feynrules.irmp.ucl.ac.be/

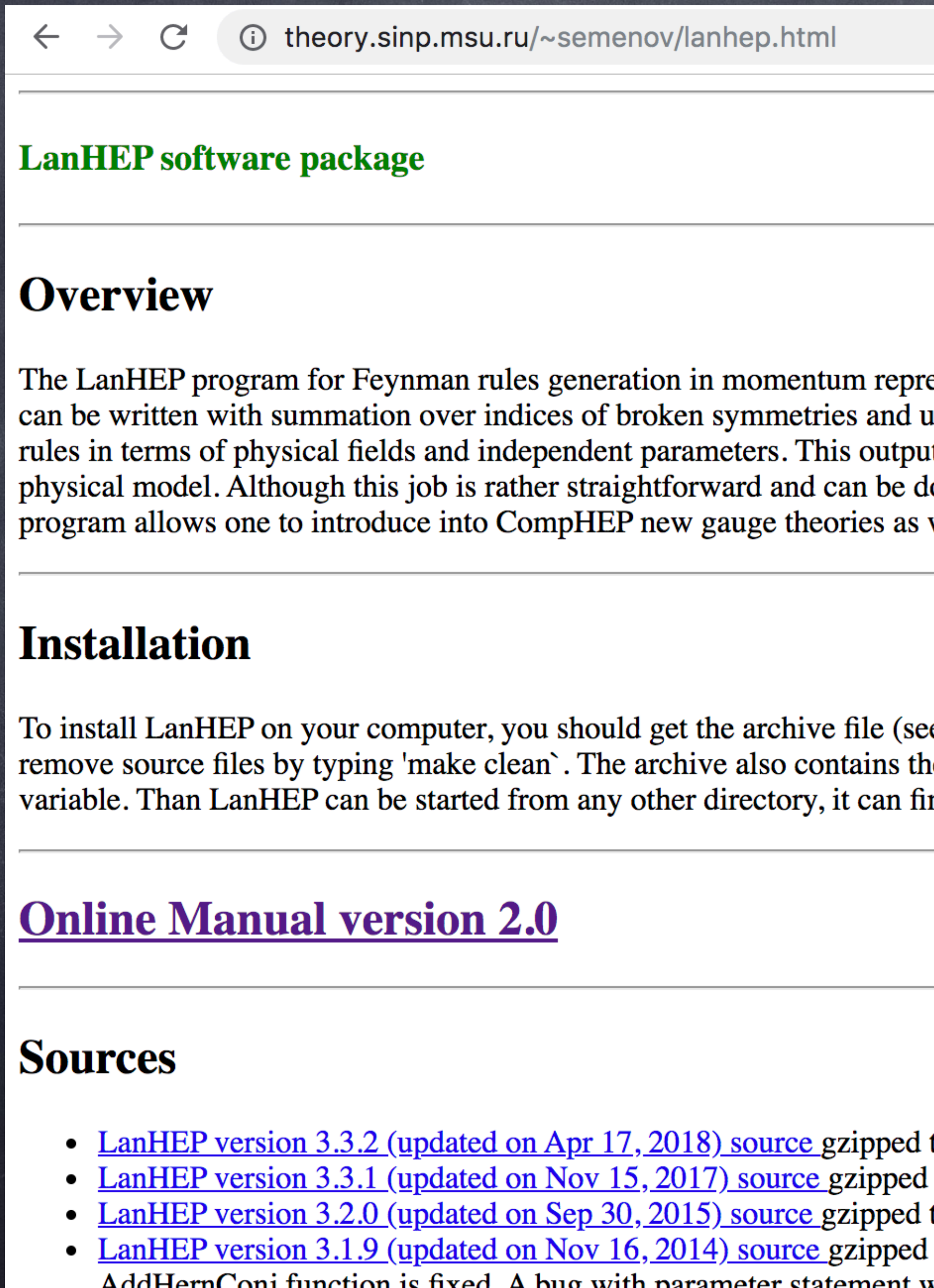
3 sarah.hepforge.org/

4 lapth.cnrs.fr/micromegas/

5 launchpad.net/maddm

6 feyncalc.github.io/
www.feynarts.de/

LanHep


theory.sinp.msu.ru/~semenov/lanhep.html

LanHEP software package

Overview

The LanHEP program for Feynman rules generation in momentum repre
can be written with summation over indices of broken symmetries and u
rules in terms of physical fields and independent parameters. This output
physical model. Although this job is rather straightforward and can be do
program allows one to introduce into CompHEP new gauge theories as v

Installation

To install LanHEP on your computer, you should get the archive file (see
remove source files by typing 'make clean`. The archive also contains the
variable. Than LanHEP can be started from any other directory, it can fin

Online Manual version 2.0

Sources

- LanHEP version 3.3.2 (updated on Apr 17, 2018) source gzipped t
- LanHEP version 3.3.1 (updated on Nov 15, 2017) source gzipped
- LanHEP version 3.2.0 (updated on Sep 30, 2015) source gzipped t
- LanHEP version 3.1.9 (updated on Nov 16, 2014) source gzipped
  AddHernConj function is fixed. A bug with parameter statement w

First step

Download from the web site and unpackaged the file, the latest version is recommended

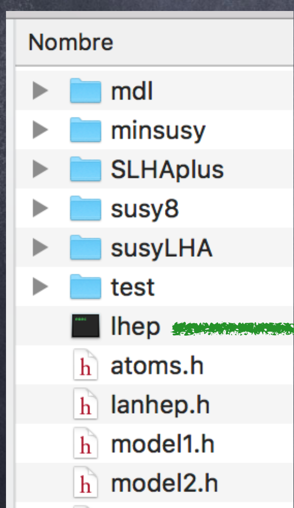
Compile (install)

To create the executable file lhep, type:

make

```
[MacBook-Pro-de-Jose:lanhep332 josehalim$ make
gcc -c -DOutputNotSet -O2 main.c
```

LanHEP directory



Executable file that generates the files with Feynaman rules

Model files directory

```
MacBook-Pro-de-Jose:lanhep332 josehalim$ cd mdl/
MacBook-Pro-de-Jose:mdl josehalim$ ls
IDMUx.mdl          calchep.rc      qed.mdl         smnlg.mdl       uedqed.mdl
IDMUx_2.mdl        feynarts.rc     qeddip.mdl      smren.mdl       ufo-static
IDMUx_case1.mdl    lhep.rc         slhaplus.mdl    stand.mdl       ufo.rc
IDMUx_case1X.mdl   newsm.mdl       sm_brst.mdl     stand1.mdl
IDMUx_case2X.mdl   qcd.mdl         sm_tex.mdl      stand_ufo.lhep
MacBook-Pro-de-Jose:mdl josehalim$
```

Create output files

./lhep -OutDir OUTPUT DIR/ FILE DIR/FILE.mdl -tex

Type of the output files

Path for your output files

Path for your input files

Path for your output files

```
MacBook-Pro-de-Jose:mdl josehalim$ cd ..
MacBook-Pro-de-Jose:lanhep332 josehalim$ ./lhep -OutDir SM/ mdl/stand.mdl -tex
Folder SM/ is created.
File mdl/stand.mdl processed, 0 sec.

MacBook-Pro-de-Jose:lanhep332 josehalim$
```

output directory

Source file

Output file extension

Types of files

```
-CompHEP or -co for CompHEP format;
-CalcHEP or -ca for CalcHEP format;
-FeynArts or -fa for FeynArts format;
-ufo for UFO format;
-tex for LaTeX format.
```

Now, for micrOmegas

```
MacBook-Pro-de-Jose:lanhep332 josehalim$ ./lhep -OutDir SM/ mdl/stand.mdl -ca
File mdl/stand.mdl processed, 0 sec.

MacBook-Pro-de-Jose:lanhep332 josehalim$
```

Output files:

- Lgrng.mdl → Contains the Feynman rules
- prtcls.mdl → It is for the particles
- Vars.mdl → It is for model parameters
- Func.mdl → It is for relations defined in model file

Extension .mdl is for -ca

File.mdl structure

stand.mdl 〉 No Selection

```
%
% Standard Model - unitary and t'Hooft-Feynman gauges.
%

keys gauge_fixing=Feynman.
keys CKMdim=3.

do_if gauge_fixing==Feynman.
    model 'Stand. Model (Feyn. gauge)'/4.
do_else_if gauge_fixing==unitary.
    model 'Stand. Model (un. gauge)'/3.
do_else.
    write('Error: the key "gauge" should be either "Feynman" or "unitary".').
    quit.
end_if.

let g5=gamma5.
```

To comment

To define character

To select a gauge

output directory

output directory

File.mdl structure

Introduces comments

It is used to define model parameters and their values

```
%%%%%%
%%%%

parameter  EE  = 0.31333 : 'Electromagnetic coupling constant (<->1/128)',
           GG  = 1.117   : 'Strong coupling constant (Z point)  (PDG-94)',
           SW  = 0.4740  : 'sin of the Weinberg angle (PDG-94,"on-shell")',
               s12 = 0.221   : 'Parameter of C-K-M matrix (PDG-94)',
           s23 = 0.040   : 'Parameter of C-K-M matrix (PDG-94)',
               s13 = 0.0035  : 'Parameter of C-K-M matrix (PDG-94)'.

parameter  CW  = sqrt(1-SW**2) : 'cos of the Weinberg angle'.

parameter  c12  = sqrt(1-s12**2) :  'parameter  of C-K-M matrix',
           c23  = sqrt(1-s23**2) :  'parameter  of C-K-M matrix',
           c13  = sqrt(1-s13**2) :  'parameter  of C-K-M matrix'.
```

Defines gauge bosons

```
vector
     A/A: (photon, gauge),
     Z/Z:('Z boson', mass MZ = 91.187, width wZ = 2.502, gauge),
     G/G: (gluon, color c8, gauge),
     'W+'/'W-': ('W boson', mass MW = MZ*CW, width wW = 2.094, gauge).
```

Anti-particle

Particle

Defines scalars

```
scalar H/H:(Higgs, mass MH = 200, width wH = 1.461).
```

File.mdl structure

Defines fermions

```
spinor      n1:(neutrino,left),          e1:(electron),
            n2:('mu-neutrino',left),     e2:(muon, mass Mm  = 0.1057),
            n3:('tau-neutrino',left),    e3:('tau-lepton', mass Mt  = 1.777).

spinor      u:('u-quark',color c3),
            d:('d-quark',color c3),
            c:('c-quark',color c3, mass Mc  = 1.300),
            s:('s-quark',color c3, mass Ms = 0.200),
            t:('t-quark',color c3, mass Mtop = 170, width wtop = 1.442),
            b:('b-quark',color c3, mass Mb =  4.300 ).
```

It is used to define relations between fields or parameters

```
let l1={n1,e1}, L1={N1,E1}.
let l2={n2,e2}, L2={N2,E2}.
let l3={n3,e3}, L3={N3,E3}.

let q1={u,d}, Q1={U,D}, q1a={u,Vud*d+Vus*s+Vub*b}, Q1a={U,Vud*D+Vus*S+Vub*B}.
let q2={c,s}, Q2={C,S}, q2a={c,Vcd*d+Vcs*s+Vcb*b}, Q2a={C,Vcd*D+Vcs*S+Vcb*B}.
let q3={t,b}, Q3={T,B}, q3a={t,Vtd*d+Vts*s+Vtb*b}, Q3a={T,Vtd*D+Vts*S+Vtb*B}.
```

$$q_3 = \begin{pmatrix} t \\ b \end{pmatrix}$$

File.mdl structure

$$\begin{pmatrix} W_3 \\ B_1 \end{pmatrix} = \begin{pmatrix} \cos\theta_W & \sin\theta_W \\ -\sin\theta_W & \cos\theta_W \end{pmatrix} \begin{pmatrix} Z \\ A \end{pmatrix}$$

```
let B1= -SW*Z+CW*A, W3=CW*Z+SW*A, W1=('W+'+'W-')/Sqrt2,
      W2 = i*('W+'-'W-')/Sqrt2.

do_if gauge_fixing==Feynman.

let gh1 = ('W+.c'+'W-.c')/Sqrt2, gh2= i*('W+.c'-'W-.c')/Sqrt2,
        gh3= CW*'Z.c'+SW*'A.c', gh={gh1,gh2,gh3}.

let Gh1 = ('W+.C'+'W-.C')/Sqrt2, Gh2=i*('W+.C'-'W-.C')/Sqrt2,
        Gh3= CW*'Z.C'+SW*'A.C', Gh={Gh1,Gh2,Gh3}.

end_if.

let WW1 = {W1,  W2 , W3}, WW = {'W+',W3,'W-'}.

let g=EE/SW, g1=EE/CW.
```

It will be used for D_μ

Generates part of Lagrangian, in this case

$$-\frac{1}{4}F^{\mu\nu}F_{\mu\nu}$$

```
% Self-interaction of gauge bosons

lterm -F**2/4   where
    F=deriv^mu*B1^nu-deriv^nu*B1^mu.

lterm -F**2/4  where
    F=deriv^mu*G^nu^a-deriv^nu*G^mu^a+i*GG*f_SU3^a^b^c*G^mu^b*G^nu^c.

lterm -F**2/4  where
F=deriv^mu*WW1^nu^a-deriv^nu*WW1^mu^a -g*eps^a^b^c*WW1^mu^b*WW1^nu^c.
```

File.mdl structure

```
% left fermion interaction with gauge fields

lterm   anti(psi)*gamma*(1-g5)/2*(i*deriv-g*taupm*WW/2-Y*g1*B1)*psi
        where
            psi=l1,  Y=-1/2;
            psi=l2,  Y=-1/2;
            psi=l3,  Y=-1/2;
            psi=q1a, Y= 1/6;
            psi=q2a, Y= 1/6;
            psi=q3a, Y= 1/6.
```

$\bar{\psi}\gamma^{\mu}P_L D_{\mu}\psi$

$\bar{\psi}\gamma^{\mu}P_R D_{\mu}\psi$

```
% right fermion interaction with gauge fields

lterm   anti(psi)*gamma*(1+g5)/2*(i*deriv - Y*g1*B1)*psi
        where
            psi=e1,Y= -1;
            psi=e2,Y= -1;
            psi=e3,Y= -1;
            psi=u, Y=  2/3;
            psi=c, Y=  2/3;
            psi=t, Y=  2/3;
            psi=d, Y= -1/3;
            psi=s, Y= -1/3;
            psi=b, Y= -1/3.
```

Yukawa couplings

```
lterm  -M/MW/Sqrt2*g*(anti(pl)*(1+g5)/2*pr*pp + anti(pr)*(1-g5)/2*pl*PP )
    where
    M=Vud*0,  pl=q1a, pr=d;          % 0 stands for Md
    M=Vus*Ms, pl=q1a, pr=s;
    M=Vub*Mb, pl=q1a, pr=b;
    M=Vcd*0,  pl=q2a, pr=d;
    M=Vcs*Ms, pl=q2a, pr=s;
    M=Vcb*Mb, pl=q2a, pr=b;
    M=Vtd*0,  pl=q3a, pr=d;
    M=Vts*Ms, pl=q3a, pr=s;
    M=Vtb*Mb, pl=q3a, pr=b.
```

File.mdl structure

Higgs Doublet

```
do_if gauge_fixing==Feynman.

let pp = { -i*'W+.f',  (vev(2*MW/EE*SW)+H+i*'Z.f')/Sqrt2 },
    PP = {  i*'W-.f',  (vev(2*MW/EE*SW)+H-i*'Z.f')/Sqrt2 }.

do_else.

let pp = { 0,  (vev(2*MW/EE*SW)+H)/Sqrt2 },
    PP = { 0,  (vev(2*MW/EE*SW)+H)/Sqrt2 }.

end_if.
```

$$-2\lambda(\phi^\dagger\phi - v^2/2)^2$$

```
% Scalar sector

lterm -2*lambda*(pp*PP-v**2/2)**2  where
    lambda=(g*MH/MW)**2/16, v=2*MW*SW/EE.

let Dpp^mu^a = (deriv^mu+i*g1/2*B1^mu)*pp^a +
     i*g/2*taupm^a^b^c*WW^mu^c*pp^b.

let DPP^mu^a = (deriv^mu-i*g1/2*B1^mu)*PP^a
    -i*g/2*taupm^a^b^c*{'W-'^mu,W3^mu,'W+'^mu}^c*PP^b.

lterm DPP*Dpp.
```

$$D_\mu\phi$$

$$(D^\mu\phi)^\dagger(D_\mu\phi)$$

Standard model + Singlet

Scalar field is defined as DM candidate with ~

```
%%--BSM-- Singlet scalar

scalar S0/S0:(Singlet, pdg 1000025, mass MS0=100, width wS0 = auto).
```

$$V = \mu_s(S_0S_0) + \frac{1}{2}\lambda_s(S_0S_0)(\phi^\dagger\phi)$$

```
%%%%%%%%%%--- BSM --- Scalar Potential --- Lagrangian Term
%%%%

lterm -lambdas*pp*PP*S0*S0-mus*S0*S0
    where mus=-MS0**2-(1/2)*lambdas*v**2.

let DS0^mu^a = (deriv^mu+i*g1/2*B1^mu)*S0^a+i*g/2*taupm^a^b^c*WW^mu^c*S0^b.

let DS0^mu = (deriv^mu)*S0.

lterm DS0*DS0.
```

$$(D_\mu S_0)(D^\mu S_0)$$

J. McDonald, Phys. Rev. D 50, 3637 (1994)

Inert Doublet Model

$$V = \mu_1^2|\phi_1|^2 + \mu_2^2|\phi_2|^2 + \lambda_1|\phi_1|^4 + \lambda_2|\phi_2|^4 + \lambda_3|\phi_1|^2|\phi_2|^2 + \lambda_4|\phi_1^\dagger\phi_2|^2 + \left[\frac{\lambda_5}{2}(\phi_1^\dagger\phi_2)^2 + h.c.\right].$$

```
parameter mu2=MHX**2-laL*(2*MW/EE*SW)**2.
parameter la3=2*(MHC**2-mu2)/(2*MW/EE*SW)**2.
parameter la5=(MHX**2-MH3**2)/(2*MW/EE*SW)**2.
parameter la4=2*laL-la3-la5.
```

$$m_{H^\pm}^2 = \mu_2^2 + \lambda_3 v^2/2,$$
$$m_X^2 = \mu_2^2 + (\lambda_3 + \lambda_4 + \lambda_5)v^2/2,$$
$$m_A^2 = \mu_2^2 + (\lambda_3 + \lambda_4 - \lambda_5)v^2/2.$$

```
scalar h/h:('Higgs', pdg 25, mass Mh = 200, width wh = auto).
scalar '~H3'/'~H3':('odd Higgs',pdg 36, mass MH3, width wH3 = auto).
scalar '~H+'/'~H-':('Charged Higgs',pdg 37, mass MHC, width wHC=auto).
scalar '~X'/'~X'  :('second Higgs', pdg 35,  mass MHX, width wHX=auto).
```

```
let pp = { -i*'W+.f',  (vev((2*MW/EE*SW))+h+i*'Z.f')/Sqrt2 },
    PP = {  i*'W-.f',  (vev((2*MW/EE*SW))+h-i*'Z.f')/Sqrt2 }.

let qq = { -i*'~H+',  ('~X'+i*'~H3')/Sqrt2 },
    QQ = {  i*'~H-',  ('~X'-i*'~H3')/Sqrt2 }.
```

$$\phi_2 = \begin{pmatrix} H^+ \\ (X + iH_3)/\sqrt{2} \end{pmatrix}$$

E. M. Dolle and S. Su, Phys. Rev. D 80, 055012 (2009).
L. L. Honorez, C. E. Yaguna, JHEP 1009 (2010) 046.
L. L. Honorez, C. E. Yaguna, JCAP 1101 (2011) 002.

Inert Doublet Model

$(D^\mu \phi_1)^\dagger (D_\mu \phi_1)$

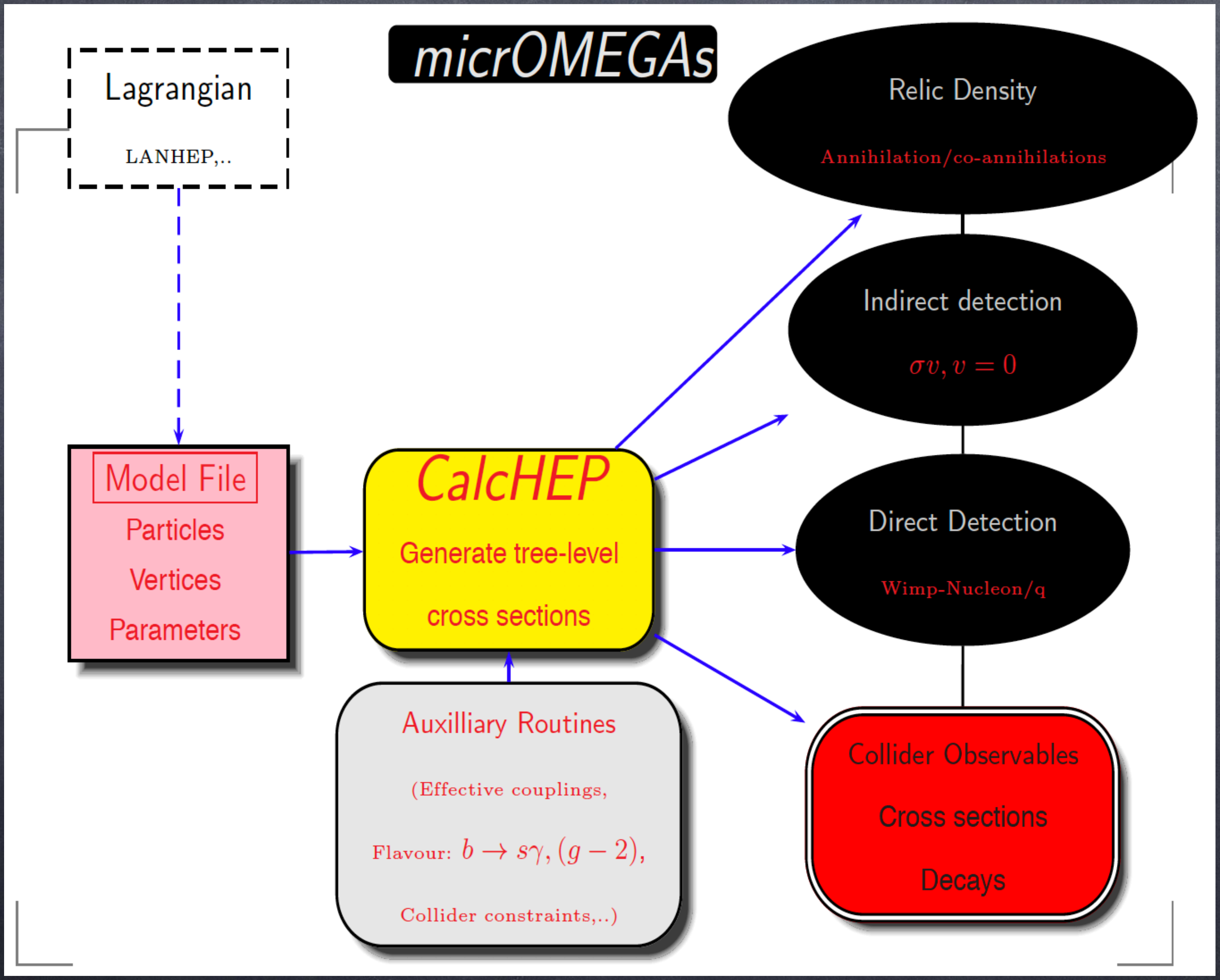
$(D^\mu \phi_2)^\dagger (D_\mu \phi_2)$

```
let Dpp^mu^a = (deriv^mu+i*g1/2*B1^mu)*pp^a +
      i*g/2*taupm^a^b^c*WW^mu^c*pp^b.

let DPP^mu^a = (deriv^mu-i*g1/2*B1^mu)*PP^a
     -i*g/2*taupm^a^b^c*{'W-'^mu,W3^mu,'W+'^mu}^c*PP^b.

lterm DPP*Dpp.

let Dqq^mu^a = (deriv^mu+i*g1/2*B1^mu)*qq^a +
      i*g/2*taupm^a^b^c*WW^mu^c*qq^b.

let DQQ^mu^a = (deriv^mu-i*g1/2*B1^mu)*QQ^a
     -i*g/2*taupm^a^b^c*{'W-'^mu,W3^mu,'W+'^mu}^c*QQ^b.

lterm DQQ*Dqq.
```

$$V = \mu_1^2 |\phi_1|^2 + \mu_2^2 |\phi_2|^2 + \lambda_1 |\phi_1|^4 + \lambda_2 |\phi_2|^4 + \lambda_3 |\phi_1|^2 |\phi_2|^2 + \lambda_4 |\phi_1^\dagger \phi_2|^2 + \left[\frac{\lambda_5}{2}(\phi_1^\dagger \phi_2)^2 + h.c.\right].$$

```
lterm -mu2*qq*QQ.
lterm -la2*(qq*QQ)**2.
lterm -la3*(pp*PP)*(qq*QQ).
lterm -la4*(pp*QQ)*(PP*qq).
lterm -la5/2*(pp*QQ)**2 + AddHermConj.
```

MicrOMEGAs



G. Belanger, F. Boudjema, A. Pukhov, arXiv:1402.0787

MicrOMEGAs

Download and unpackaged the file micromegas_5.0.8.tgz

lapth.cnrs.fr/micromegas/

MicrOMEGAs: a code for the calculation of Dark Matter Properties
including the **relic density**, direct and **indirect rates**
in a general supersymmetric model
and other models of New Physics

Geneviève Bélanger, Fawzi Boudjema, Alexander Pukhov and Andrei Semenov

MicrOMEGAs 5.0 (Generic Model)

Introduction

Documentation

Download and Install

Previous versions

Installation of Micromegas_5.0 (Freeze-in and Freeze-out)

The compressed File micromegas_5.0.8.tgz is about 20Mb. DOWNLOAD

micromegas_5.0.8 was tested for Linux and Darwin.

First unpack the file, for example tar -zxvf micromegas_5.0.8.tgz

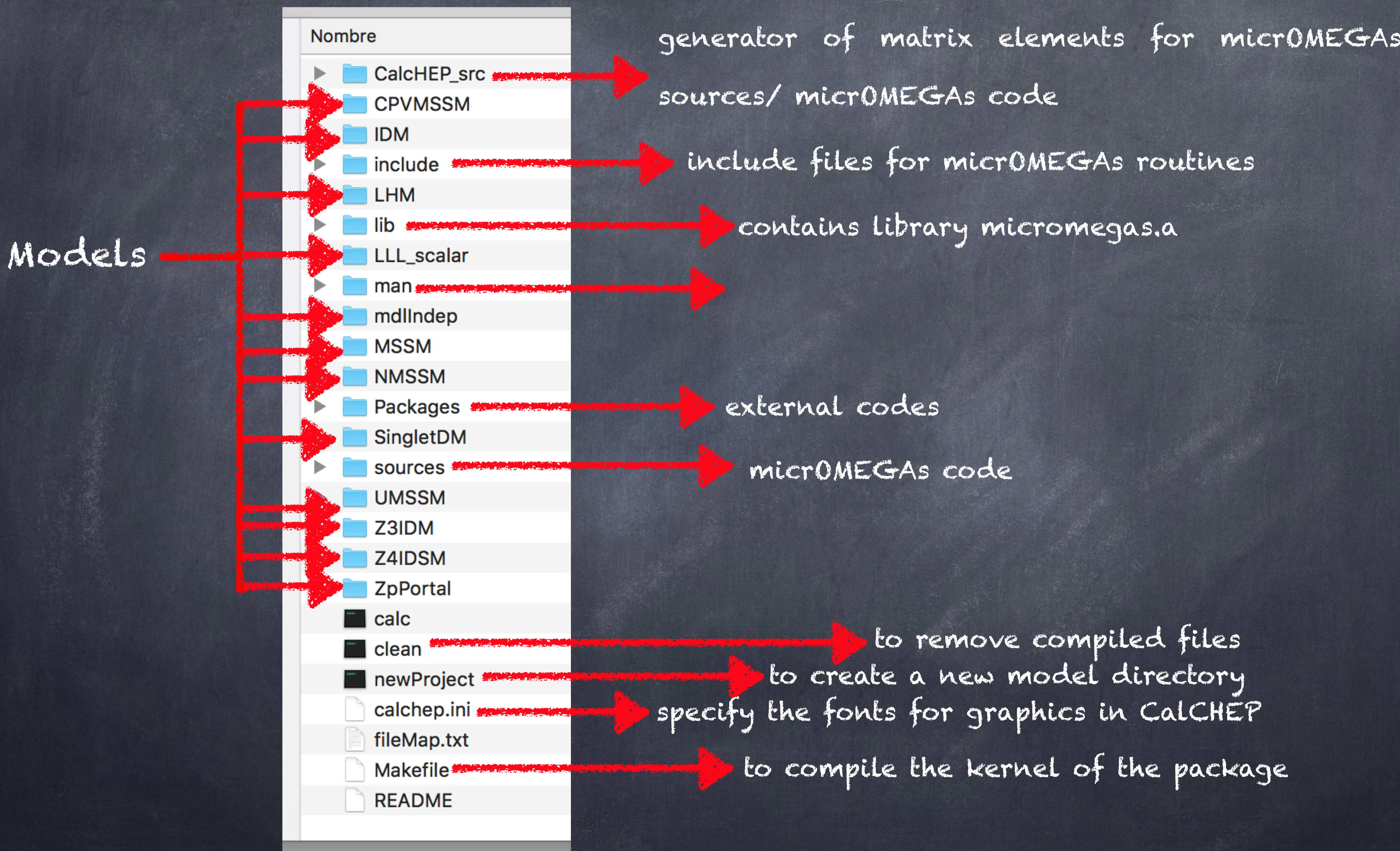
This will create a directory called micromegas_5.08

Installation: Compile with "make" inside the directory

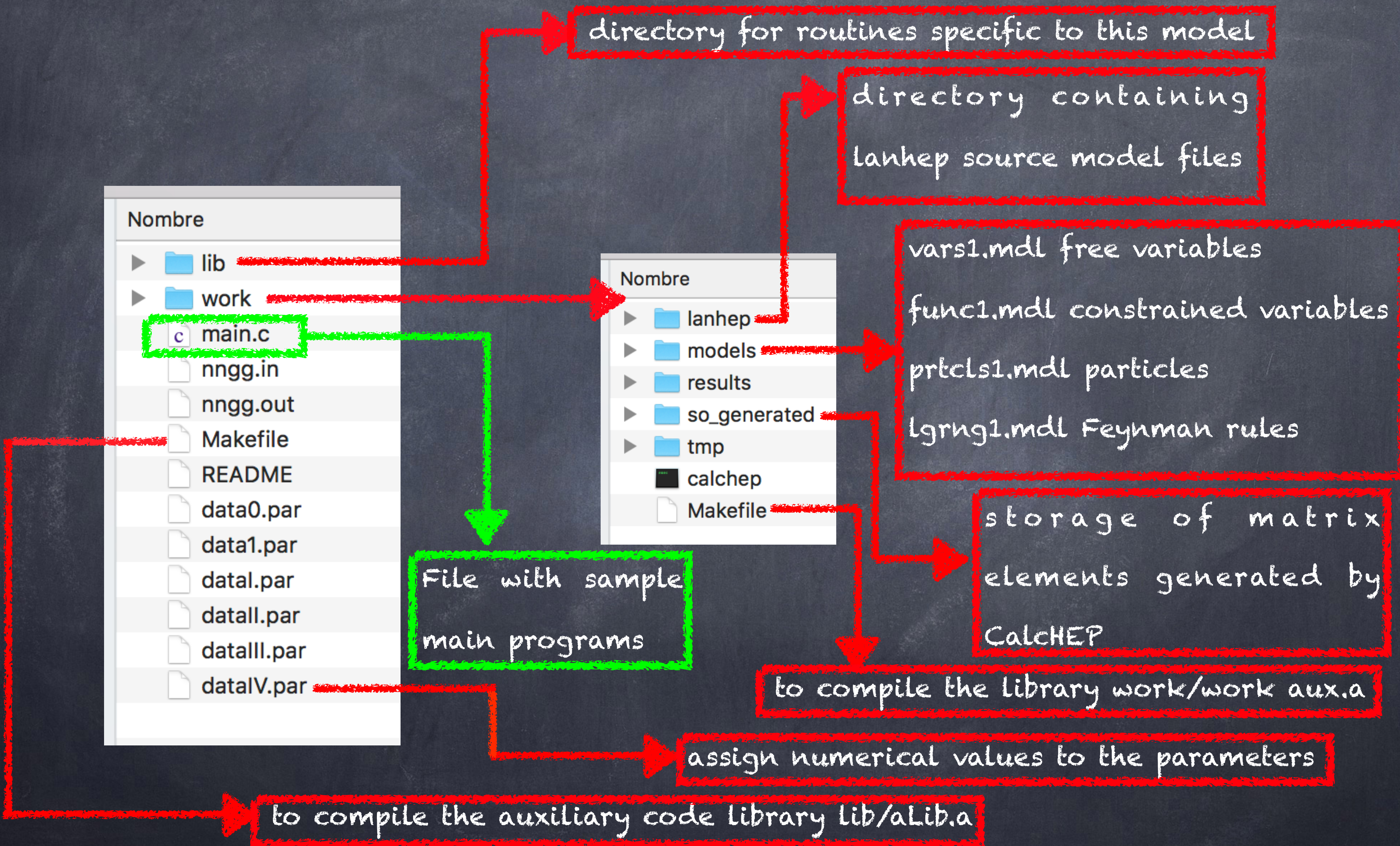
```
MacBook-Pro-de-Jose:micromegas_5.0.8 josehalim$ make
echo \#define micrO \"/Users/josehalim/Desktop/HEP-Tools/micromegas_5.0.8\"  > include/microPath.h
/Applications/Xcode.app/Contents/Developer/usr/bin/make -C CalcHEP_src  MICROMEGAS=MICROMEGAS
./getFlags
MAKE= make
UNIX Darwin
C compiler detected
: shared liblary generation OK
```

continue compiling . . .

MicrOMEGAs files



IDM directory



The main.c file

main.c

main.c 〉 No Selection

```c
/*====== Modules ===============
   Keys to switch on
   various modules of micrOMEGAs
================================*/

#define MASSES_INFO
  /* Display information about mass spectrum  */

#define CONSTRAINTS

//#define HIGGSBOUNDS
//#define HIGGSSIGNALS
#define LILITH
//#define SMODELS


#define OMEGA
  /* Calculate relic density and display contribution of  individual channels */
#define INDIRECT_DETECTION
  /* Compute spectra of gamma/positron/antiprotons/neutrinos for DM annihilation;
     Calculate <sigma*v>;
     Integrate gamma signal over DM galactic squared density for given line
     of sight;
     Calculate galactic propagation of positrons and antiprotons.
  */

//#define LoopGAMMA
    /* Calculate discrete  photon spectrum caused by annihilation of
       DM into two photons and Z-photon
    */
```

The main.c file, cont..

```
/*#define RESET_FORMFACTORS*/
  /* Modify default nucleus form factors,
  */
#define CDM_NUCLEON
  /* Calculate amplitudes and cross-sections for  CDM-mucleon collisions */

#define CDM_NUCLEUS
  /* Calculate number of events for 1kg*day and recoil energy distibution
      for various nuclei
  */

#define NEUTRINO //neutrino telescope

#define DECAYS
//#define CROSS_SECTIONS

/*===== end of Modules  ======*/

/*===== Options ========*/
/*#define SHOWPLOTS*/
     /* Display  graphical plots on the screen */

//#define CLEAN   to clean intermediate files
/*===== End of DEFINE  settings ===== */


#include"../include/micromegas.h"
#include"../include/micromegas_aux.h"
#include"lib/pmodel.h"
```

Start the routine

Modules for main program

```
#define MASSES_INFO          //Displays information about mass spectrum
#define CONSTRAINTS          //Displays B_>sgamma, Bs->mumu, etc
#define HIGGSBOUNDS          //calls HiggsBounds/HiggsSignal to constrain the Higgs sector
#define LILITH               //calls LiLith to constrain the Higgs sector
#define SModelS              //calls SModelS to constrain the new physics sector
#define OMEGA                //Calculates the relic density
#define FREEZEIN             //Calculates the relic density in the freeze-in mechanism
#define INDIRECT_DETECTION //Signals of DM annihilation in galactic halo
#define LoopGAMMA            //Gamma-Ray lines - available only in some models
#define RESET_FORMFACTORS  //Redefinition of Form Factors and other
                             //parameters
#define CDM_NUCLEON          //Calculates amplitudes and cross-sections
                             //for DM-nucleon collisions
#define CDM_NUCLEUS          //Calculates number of events for 1kg*day
                             //and recoil energy distribution for various nuclei
#define NEUTRINO             //Calculates flux of solar neutrinos and
                             //the corresponding muon flux
#define DECAYS               //Calculates decay widths and branching ratios
#define CROSS_SECTIONS       //Calculates cross sections
#define CLEAN                //Removes intermediate files.
```

Relic density

Call the modules

Start the code

main_only_relic

main_only_relic.c 〉 main

```c
/*====== Modules ===============
   Keys to switch on
   various modules of micrOMEGAs
================================*/

#define MASSES_INFO
  /* Display information about mass spectrum  */
#define OMEGA
  /* Calculate relic density and display contribution of  individual channels */
#define DECAYS
//#define CROSS_SECTIONS

/*===== end of Modules  ======*/


#include"../include/micromegas.h"
#include"../include/micromegas_aux.h"
#include"lib/pmodel.h"

int main(int argc,char** argv)
{ int err;
  char cdmName[10];
  int spin2, charge3,cdim;

  ForceUG=0;  /* to Force Unitary Gauge assign 1 */

  if(argc==1)
  {
      printf(" Correct usage:  ./main  <file with parameters> \n");
      printf("Example: ./main data1.par\n");
      exit(1);
  }
```

Conditional statement to ensure the use of the file data.par

Read the values from data.par

Find DM candidates inside the model file

Find DM candidates inside the model file

Compute relic density

```
  err=readVar(argv[1]);

  if(err==-1)     {printf("Can not open the file\n"); exit(1);}
  else if(err>0)  { printf("Wrong file contents at line %d\n",err);exit(1);}

  err=sortOddParticles(cdmName);
  if(err) { printf("Can't calculate %s\n",cdmName); return 1;}

   qNumbers(cdmName, &spin2, &charge3, &cdim);
   printf("\nDark matter candidate is '%s' with spin=%d/2\n",cdmName,spin2);
   if(charge3) { printf("Dark Matter has electric charge %d/3\n",charge3); exit(1);}
   if(cdim!=1) { printf("Dark Matter is a color particle\n"); exit(1);}
#ifdef MASSES_INFO
{
  printf("\n=== MASSES OF HIGGS AND ODD PARTICLES: ===\n");
  printHiggs(stdout);
  printMasses(stdout,1);
}
#endif

#ifdef OMEGA
{ int fast=1;
  double Beps=1.E-4, cut=0.01;
  double Omega,Xf;

// to exclude processes with virtual W/Z in DM   annihilation
   VZdecay=1; VWdecay=1; cleanDecayTable();


//   to include processes with virtual W/Z  also  in co-annihilation
//   VZdecay=2; VWdecay=2; cleanDecayTable();

  printf("\n==== Calculation of relic density =====\n");
  Omega=darkOmega(&Xf,fast,Beps,&err);

//    Omega=darkOmega2(fast,Beps);
  printf("Xf=%.2e Omega=%.2e\n",Xf,Omega);
  if(Omega>0)printChannels(Xf,cut,Beps,1,stdout);
//   VZdecay=1; VWdecay=1; cleanDecayTable();  // restore default

}
#endif

  return 0;
}
```

Ωh^2

Calculates the dark matter relic density.

This routine solves the Boltzmann differential evolution equation

```
darkOmega(&Xf, fast, Beps, &err )
```

$X_f = \frac{M_{DM}}{T_f}$, T_f characterizes the freeze-out temperature

The fast=1 forces the fast calculation. This is the recommended option and gives an accuracy around 1%.

The parameter Beps defines the criteria for including a given coannihilation channel in the computation of the thermally averaged cross-section. The recommended value is Beps = $10^{-6} - 10^{-4}$ whereas if Beps =1 only annihilation of the lightest odd particle is computed.

Non-zero error code means that the temperature where thermal equilibrium between the DM and SM sectors is too large.

Now it's the turn to compile our code

make main=SOME_NAME.c

```
MacBook-Pro-de-Jose:IDM josehalim$ make main=main_only_relic.c
/Applications/Xcode.app/Contents/Developer/usr/bin/make -C work
make[1]: Nothing to be done for `all'.
/Applications/Xcode.app/Contents/Developer/usr/bin/make -C lib
make[1]: Nothing to be done for `all'.
/Applications/Xcode.app/Contents/Developer/usr/bin/make -C ../sources
make[1]: Nothing to be done for `all'.
clang  -g -fsigned-char -std=gnu99 -o main_only_relic main_only_relic.c  lib/aLib.a ../lib/micromegas.a  /Users/josehalim/Desktop/HEP-Tools/micromegas_5.0.8/CalcHEP_src/lib/dynamic_me.a ../lib/micromegas.a work/work_aux.a  lib/aLib.a   /Users/josehalim/Desktop/HEP-Tools/micromegas_5.0.8/CalcHEP_src/lib/sqme_aux.so /Users/josehalim/Desktop/HEP-Tools/micromegas_5.0.8/CalcHEP_src/lib/libSLHAplus.a /Users/josehalim/Desktop/HEP-Tools/micromegas_5.0.8/CalcHEP_src/lib/num_c.a   /Users/josehalim/Desktop/HEP-Tools/micromegas_5.0.8/CalcHEP_src/lib/serv.a /Users/josehalim/Desktop/HEP-Tools/micromegas_5.0.8/CalcHEP_src/lib/ntools.a     /Users/josehalim/Desktop/HEP-Tools/micromegas_5.0.8/CalcHEP_src/lib/dummy.a  -ldl   -lm -lpthread
```

Ready to obtain the DM relic density

```
MacBook-Pro-de-Jose:IDM josehalim$ ls
Makefile        dataI.par       lib             main_only_relic         nngg.out
README          dataII.par      main            main_only_relic.c       work
data0.par       dataIII.par     main.c          main_only_relic.dSYM
data1.par       dataIV.par      main.dSYM       nngg.in
```

Run the compile file with:

./SOME_NAME data.par

dataI.par is used to give different values to the model parameters

```
MacBook-Pro-de-Jose:IDM josehalim$ ./main_only_relic dataI.par

Dark matter candidate is '~X' with spin=0/2

=== MASSES OF HIGGS AND ODD PARTICLES: ===
Higgs masses and widths
      h   500.00 6.41E+01

Masses of odd sector Particles:
~X       : MHX      =  70.000 || ~H3      : MH3      =  76.000 || ~H+      : MHC      = 190.000


==== Calculation of relic density =====
Xf=2.46e+01 Omega=5.49e-02
# Channels which contribute to 1/(omega) more than 1%.
# Relative contributions in % are displayed
   31% ~X ~X ->W+ W-
   10% ~H3 ~X ->d D
   10% ~H3 ~X ->s S
   10% ~H3 ~X ->b B
    8% ~H3 ~X ->u U
    8% ~H3 ~X ->c C
    5% ~H3 ~X ->ne Ne
    5% ~H3 ~X ->nm Nm
    5% ~H3 ~X ->nl Nl
    2% ~X ~X ->Z Z
    2% ~H3 ~X ->e E
    2% ~H3 ~X ->m M
    2% ~H3 ~X ->l L
MacBook-Pro-de-Jose:IDM josehalim$
```

For instance, a data1.par file structure

```
Mtp     173.5
MHX     600
MH3     601
MHC     604
laL     0.001
la2     0.01
Mh      125
```

Start up with new model

In the principal directory

./newProject MODEL NAME

```
[MacBook-Pro-de-Jose:micromegas_5.0.8 josehalim$ ./newProject ExampleCOMHEP
x Makefile
x README
x data.par
x lib/
x lib/Makefile
x lib/pmodel.h
x lib/pmodel.fh
x lib/model.c
x main.c
x main.cpp
x work/
x work/Makefile
x work/so_generated/
x work/lanhep/
x work/lanhep/Makefile
x work/models/
x work/models/vars1.mdl
x work/models/lgrng1.mdl
x work/models/func1.mdl
x work/models/prtcls1.mdl
x work/models/extlib1.mdl
x work/calchep
x work/results/
x work/tmp/
```

Then, the new directory is created

```
[MacBook-Pro-de-Jose:micromegas_5.0.8 josehalim$ ls
CPVMSSM         LHM             NMSSM           UMSSM           calc            include         newProject
CalcHEP_src     LLL_scalar      Packages        Z3IDM           calchep.ini     lib             sources
ExampleCOMHEP   MSSM            README          Z4IDSM          clean           man
IDM             Makefile        SingletDM       ZpPortal        fileMap.txt     mdlIndep
[MacBook-Pro-de-Jose:micromegas_5.0.8 josehalim$ cd ExampleCOMHEP/
[MacBook-Pro-de-Jose:ExampleCOMHEP josehalim$ ls
Makefile        README          data.par        lib             main.c          main.cpp        work
MacBook-Pro-de-Jose:ExampleCOMHEP josehalim$
```

main.c file as the others model directories

The work directory is ready to introduce the model files: var.mdl, parameter.mdl, lng.mdl, func.mdl

Example : SM+Singlet

Again, main file is compiled

make main=MAIN NAME.c

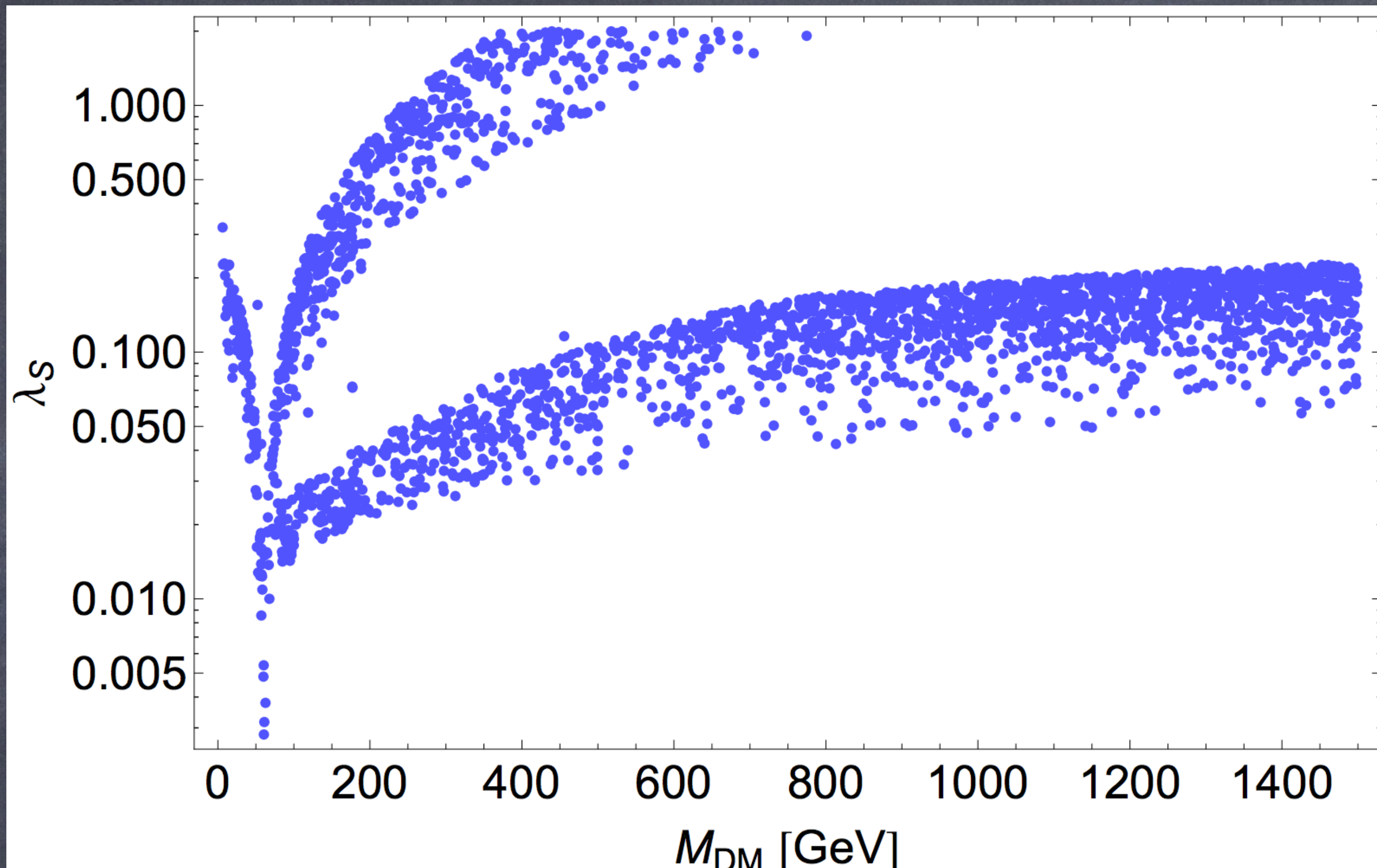
```c
/*====== Modules ===============
   Keys to switch on
   various modules of micrOMEGAs
================================*/

//#define MASSES_INFO
  /* Display information about mass spectrum  */
#define OMEGA
#define CLEAN   to clean intermediate files
/*===== End of DEFINE  settings ===== */


#include"../sources/micromegas.h"
#include"../sources/micromegas_aux.h"
#include"lib/pmodel.h"

int main()
{ int err;
  char cdmName[10];
  int spin2, charge3,cdim;

```

```c

double var = 0.0;
int i;

for (i=0; i<100; i++)
{
var =100 + i*10;
  err=sortOddParticles(cdmName);
  if(err) { printf("Can't calculate %s\n",cdmName); return 1;}

   qNumbers(cdmName, &spin2, &charge3, &cdim);

  err = assignVal( "Mhs", var);
  err = assignVal( "lams", 0.9);

#ifdef OMEGA
{ int fast=1;
  double Beps=1.E-5, cut=0.01;
  double Omega,Xf;

 Omega=darkOmega(&Xf,fast,Beps);
 printf("%.2e\t %.4e\n",var, Omega);
}
#endif

}

  return 0;
}
```

We execute it and we extract the values as text

./MAIN NAME >> EXIT FILE.txt

Example : SM+Singlet



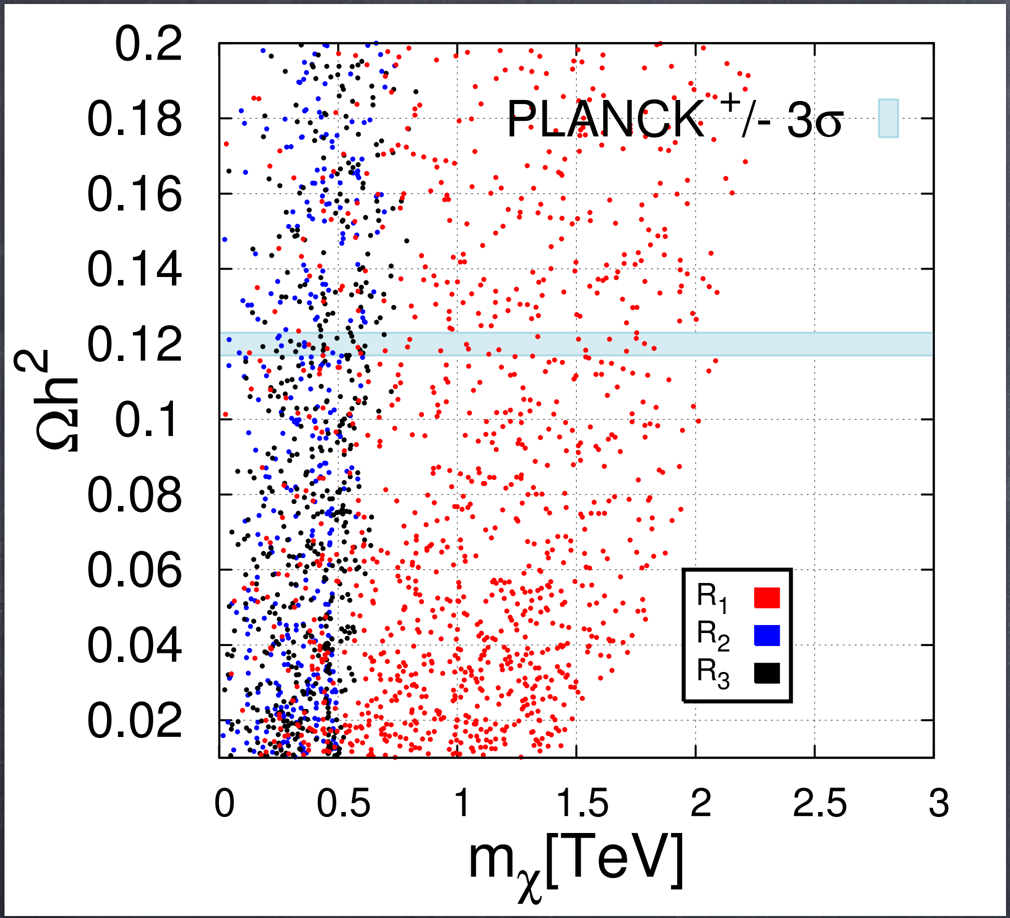
* Use random variable for λ_s and M_{DM}

* Select only the values that satisface $\Omega^2_{DM} = 0.1186 \pm 0.002$ *

*M. Tanabashi et al. (Particle Data Group),(2018)

Another Example: IDM+Singlet

$$G_{SM} \otimes U(1)_X \xrightarrow{\langle \mathcal{S}_X \rangle} G_{SM} \xrightarrow{\langle \Phi_1 \rangle} SU(3)_C \otimes U(1)_{EM}$$



$$\Phi_1 = \begin{pmatrix} \phi_1^+ \\ \frac{1}{\sqrt{2}}(\upsilon + \phi_1 + i\eta_1) \end{pmatrix},$$

$$\Phi_2 = \begin{pmatrix} \phi_2^+ \\ \frac{1}{\sqrt{2}}(\phi_2 + i\eta_2) \end{pmatrix},$$

$$\mathcal{S}_X = \frac{1}{\sqrt{2}}(\upsilon_x + s_x + i\eta_x).$$

$BR(Z' \to l^- l^+)$ excludes with $m_{Z'} < 3TeV$ for $g_x = 0.4$

- $R_1 : \lambda_{2x} \sim \mathcal{O}(10^{-3} - 10^{-2})$,
- $R_2 : \lambda_{2x} \sim \mathcal{O}(10^{-4} - 10^{-3})$,
- $R_3 : \lambda_{2x} = 0 \cup \mathcal{O}(10^{-7} - 10^{-3})$.

M.A. Arroyo-Ureña, R. Gaitan, R. Martinez, J.H. Montes de Oca.
e-Print: arXiv:1907.08231 [hep-ph]

Summary

* LanHep is a program to translate the information of a model into the language of CalcHEP, FeynCalc, MadGraph and Latex.
* LAnHEP is as easy to use as Latex
* MicroMEGAs is a program to study physics and obtain numerically the observables associated with dark matter.
* MicrOMEGAS vs. Work by hand, this is a safe bet.

Thank you for your attention

josehalim@comunidad.unam.mx