

PLR

Yitong Liu
Spring 2019

Xe1T PLR

- Download location:
 - Laidbax (Limits codes): <https://github.com/XENON1T/laidbax>
 - Blueice (Likelihood functions & analysis): <https://github.com/JelleAalbers/blueice>
Used in the limit code, creating the likelihood functions, finding the limits
 - PAX (the processor for analyzing XENON, do digital processing and other data processing on the Xe1T raw data): <https://github.com/XENON1T/pax>
Used in the base model, import units (m,ms...)

Summary

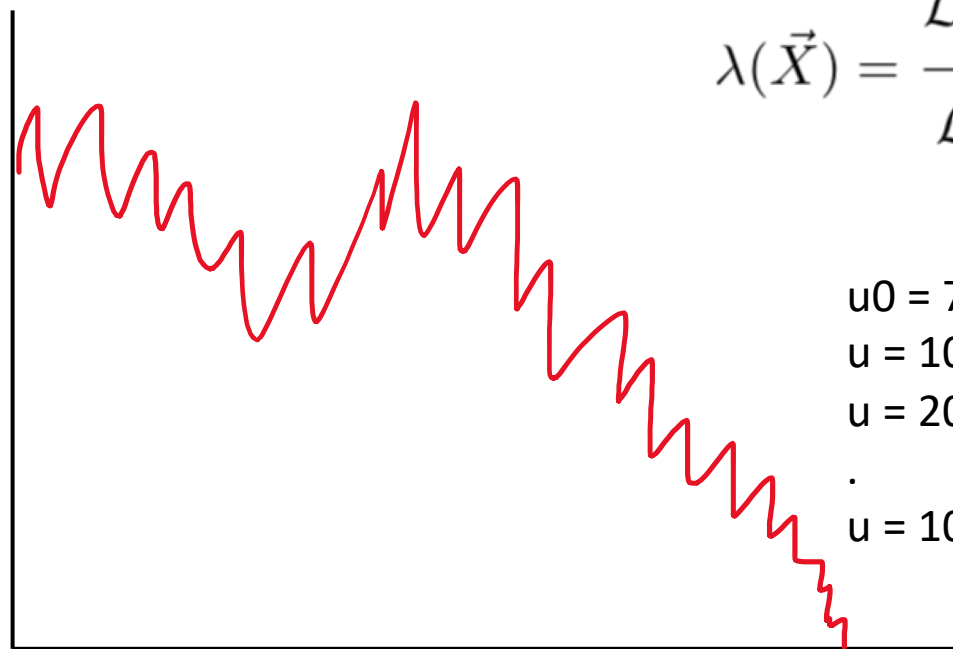
- The structure of the code

Test wimp masses (7, 10, 20, 50, 100, 200, 1000) →
Create the configuration (base model removes wimp sources) →
Add the wimp source from the CSV files →
Create the likelihood function (blueice object) →
Add rate parameter (WIMP mass) & shape parameter (Leff) →
Simulate background-only data based on the 50Gev likelihood function →
Some test plots on 50 GeV mass (background data sets, likelihood ratios) →
Set each likelihood function of different WIMP mass to the background data →
Find the limit (blueice, 90% CL, upper kind) * scale ($1e-45$) →
Plot the median for the limits

PDF

$$q = -2\ln(\lambda)$$

Numbers



$$\lambda(\vec{X}) = \frac{\mathcal{L}_{M+P} \left((\mu_0, \hat{\theta}) \middle| \vec{X} \right)}{\mathcal{L}_{M+P} \left((\hat{\mu}, \hat{\theta}) \middle| \vec{X} \right)}$$

Best fit

Best fit

u0 = 7GeV, float theta double hat (.....)

u = 10GeV, float theta hat (....)

u = 20GeV, float theta hat (....)

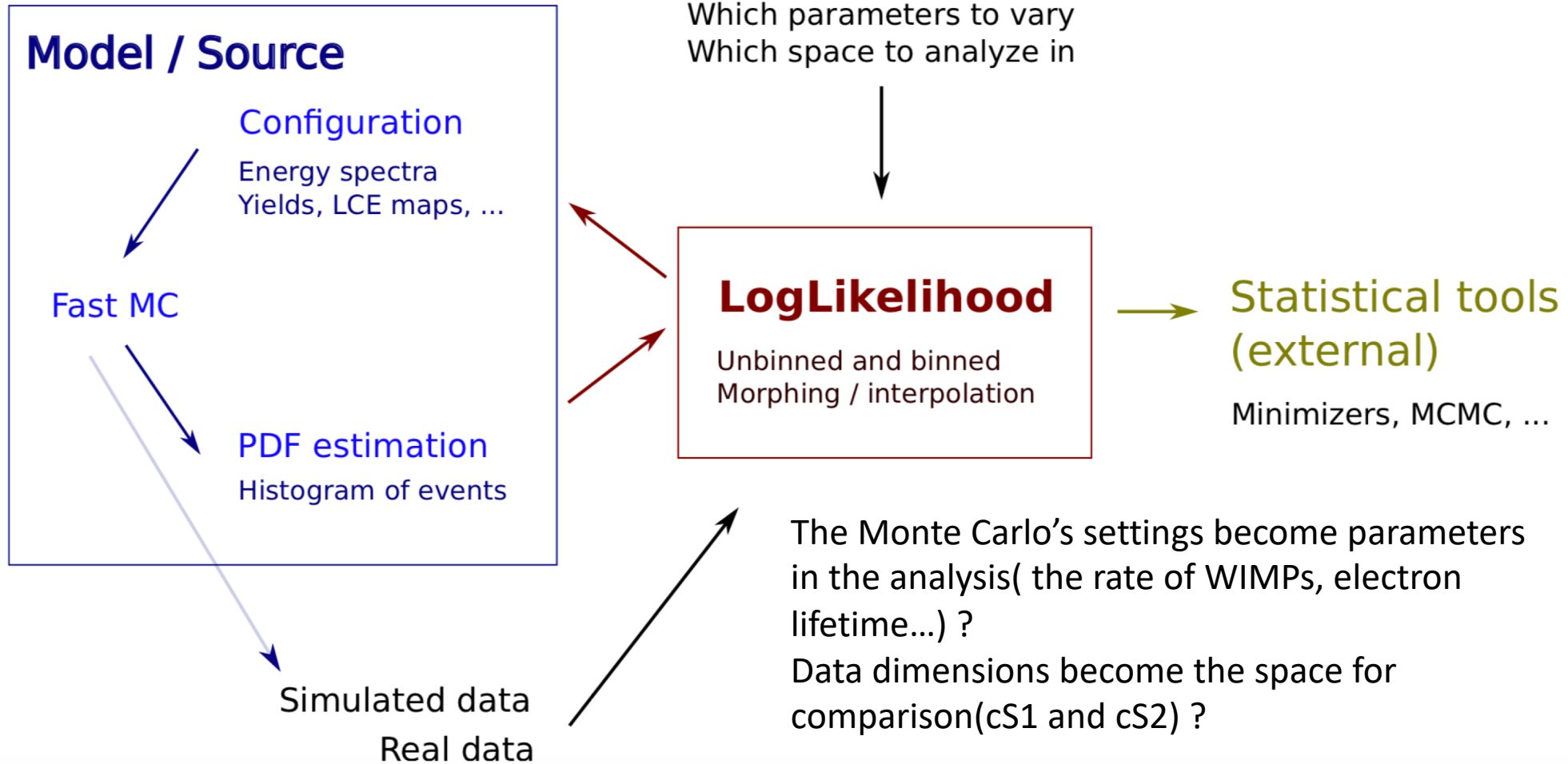
.

u = 1000GeV, float theta hat (....)

Test statistics (q)

Monte Carlos

```
lf = LogLikelihood(c)
```



Configuration

- Sources:
 - ER background($1e7$)
 - CNNS(coherent neutrino-nucleus scattering, $5e6$)
 - radiogenic neutrons($55e6$)
 - accidental coincidences
 - wall leakage
 - anomalous flat component
 - 50 GeV WIMP($5e6$)

WIMP parameters

	XE1T	LZ
Detection mechanism	elastic nr	
Interaction	SI	
nr_response_cutoff	1 keV	0.1 keV
Er_response_cutoff	0.18 keV	0.02kev
WIMP sigma nucleon	$1e^{-45}$	$1e^{-45} \rightarrow 10^{-47}$
Wimp mass	50 GeV	50 GeV

Detector parameter

	XE1T	LZ
S1_area_threshold (efficiency operates on coincidence)	0	
S2_area_threshold	150	415*1.2
S1_min_photons_detected (coincidence requirement on S1)	3	S1 coincidence level: 3-fold
S1, S2 bias (primarily due to self-trigger)	**csv files	
e_lifetime	452 us	850 us
v_drift	1.44 m/ms	1.88 m/ms
Drift field	117 V/cm	310 V/cm
Photon-detection-efficiency	0.1442/1.15 ~0.125	gi_liquid = 0.119 gi_gas = 0.102
double_pe_emission_probability	0.15	0.23
Electron extraction efficiency	1	0.95
S2 gain	?11.52/1.15 ~10.02	79
Single phe width	0.4181	0.38

For sampling the light and charge yield in space

	XE1T	LZ
Fiducial volume radius	39.85cm	69cm
Fiducial volume zmax	-83.45 cm	
Fiducial volume zmin	-13.45 cm	
Fiducial_mass	1042 kg	5.6 tonne = 5600kg
N_samples_for_source_positions	Int(1e5)	Equally sampled for LZ
s1_relative_ly_map	file	



$$\text{np.pi} * \text{rmax}^{**2} * (\text{zmax} - \text{zmin}) * \text{density}$$

S1/S2 generation parameters

	Xe1T	LZ
NEST's basic quanta yield	73	
Fano factor for smearing the base quanta yield(From the NEST code)	0.03	
S1 peak detection efficiency(number of detected photons)	(0. , 0. , 0. , 0.5135, 0.787 , 0.851 , 0.9165, 0.9255.....1)	

ER

	Xe1T	LZ
Er reference energy	5 KeV	
Er max response energy	12 KeV	
Er polynomial order	3	
Er_qy_0, Er_qy_1, Er_qy_2	0	
the coefficients for the PCA and the normalization before and after applying the PCA		
p_er_electron_fluctuation	0.045	

NR global fit parameters

	Xe1T	LZ
Nr alpha.....nr lambda		
Lindhard_k		
p_nr_electron_fluctuation	0.02	

Add wimp source to the configuration (c)

energy distribution: wimp Xgev csv files

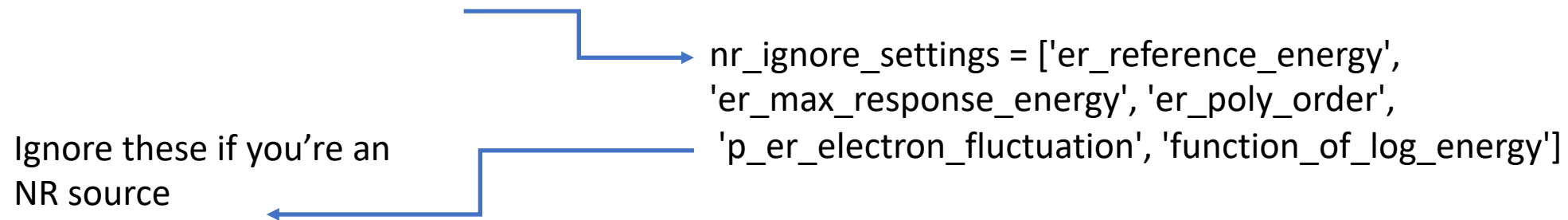
color: red

recoil type: nr

name: wimp Xgev

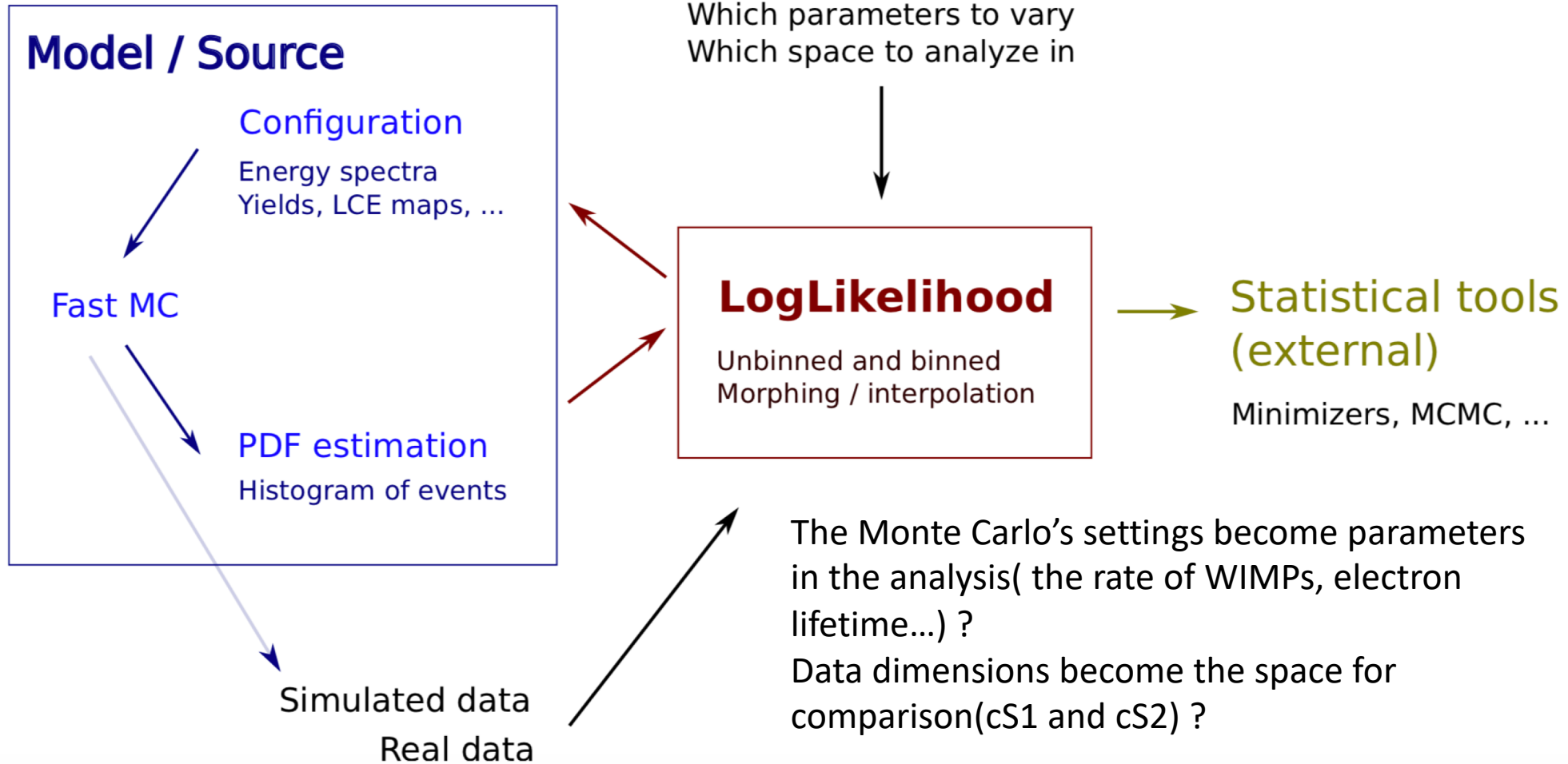
events for pdf: the amount of events to simulate to create the PDF.

ignore setting: nr ignore settings



PDF from blueice

```
lf = LogLikelihood(c)
```



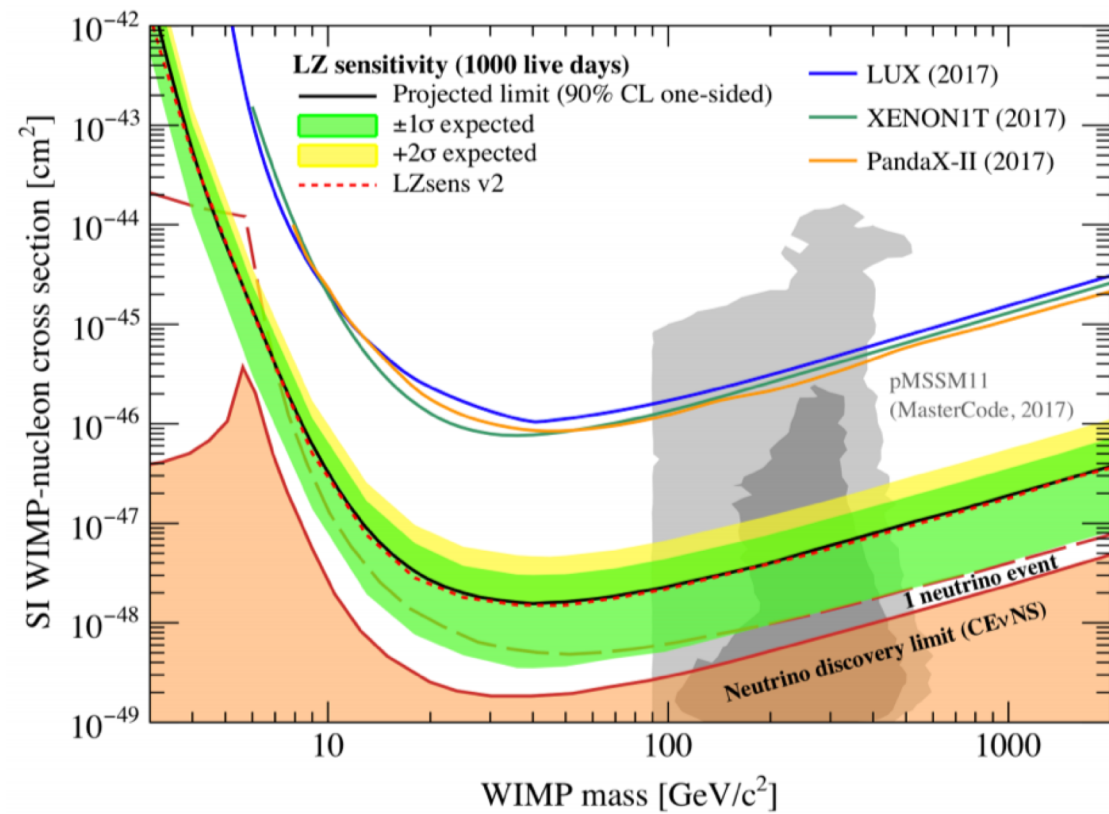
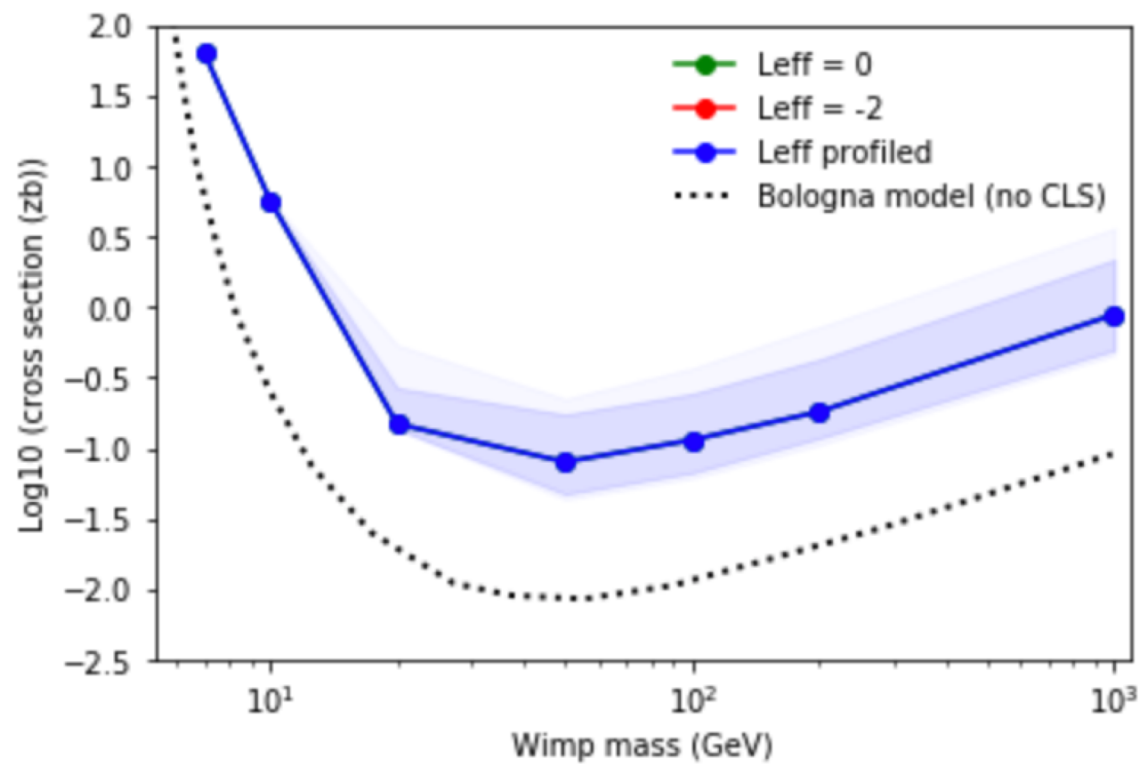
How to solve it?

- First, how to create a problem like this?
- Start with the mass of 10 GeV
 - # events for pdf
 - # trials
 - Shape parameter of Leff
- Playing with numbers set # trials = 100, no Leff shape parameter

to have an error or not to have an error									
1 x	8 x	15 x	22 x	29 o	36 o	43 o	50 o	63 o	82 o
2 o	9 x	16 x	23 x	30 o	37 o	44 o	51 o	66 x	84 x
3 x	10 x	17 x	24 o	31 o	38 o	45 x	52 o	67 o	90 o
4 x	11 o	18 x	25 x	32 x	39 o	46 o	53 x	70 o	99 o
5 x	12 x	19 x	26 x	33 o	40 o	47 o	54 o	72 o	100o
6 x	13 o	20 x	27 x	34 o	41 o	48 x	55 o	75 o	107x
7 x	14 x	21 x	28 o	35 x	42 x	49 x	60 o	77 o	135x

o means no error
x means an error

- From patterns, it is random not to have an error with small numbers of events, however, from 140~160, they are all error-free, one can conclude that up to a large number, it is not a random anymore, take the average, 150 is the number.
- Back to the csv files, for 10 GeV, 1000 events in total, with 86 events that are not zero, for 7 GeV, 1000 events in total, with 40 events that are not zero, if 150 for 10 GeV, around 322 for 7 GeV
- Try 325



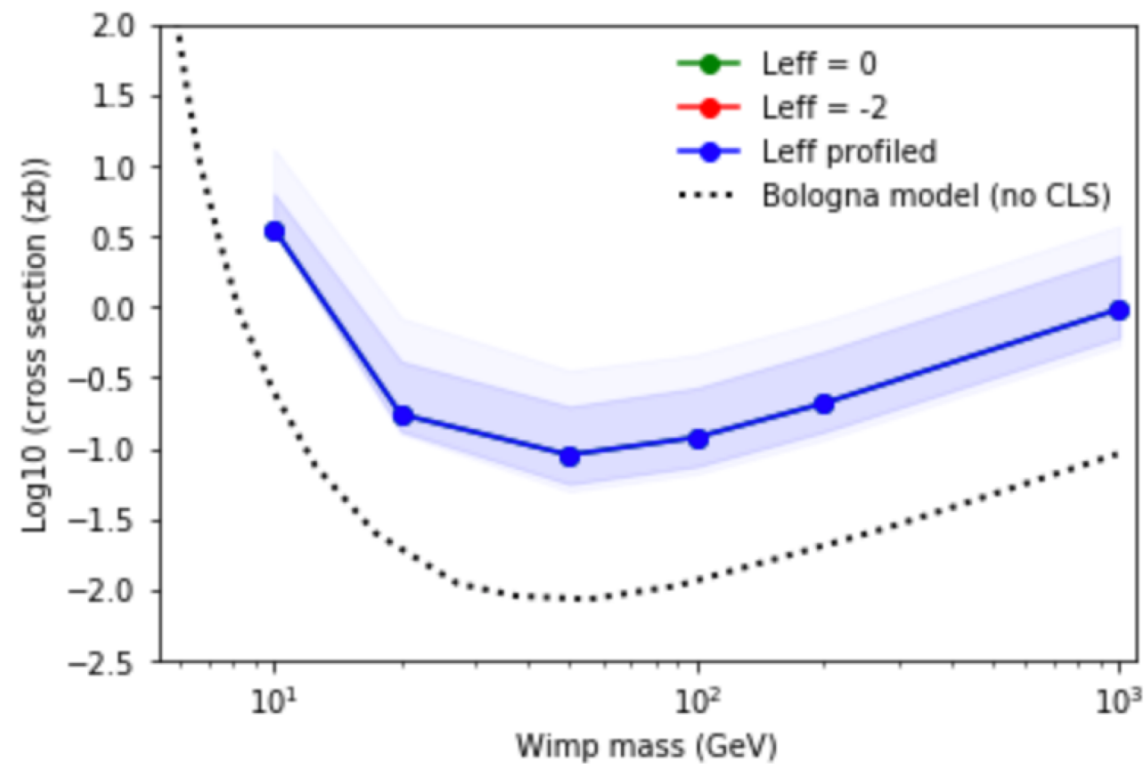
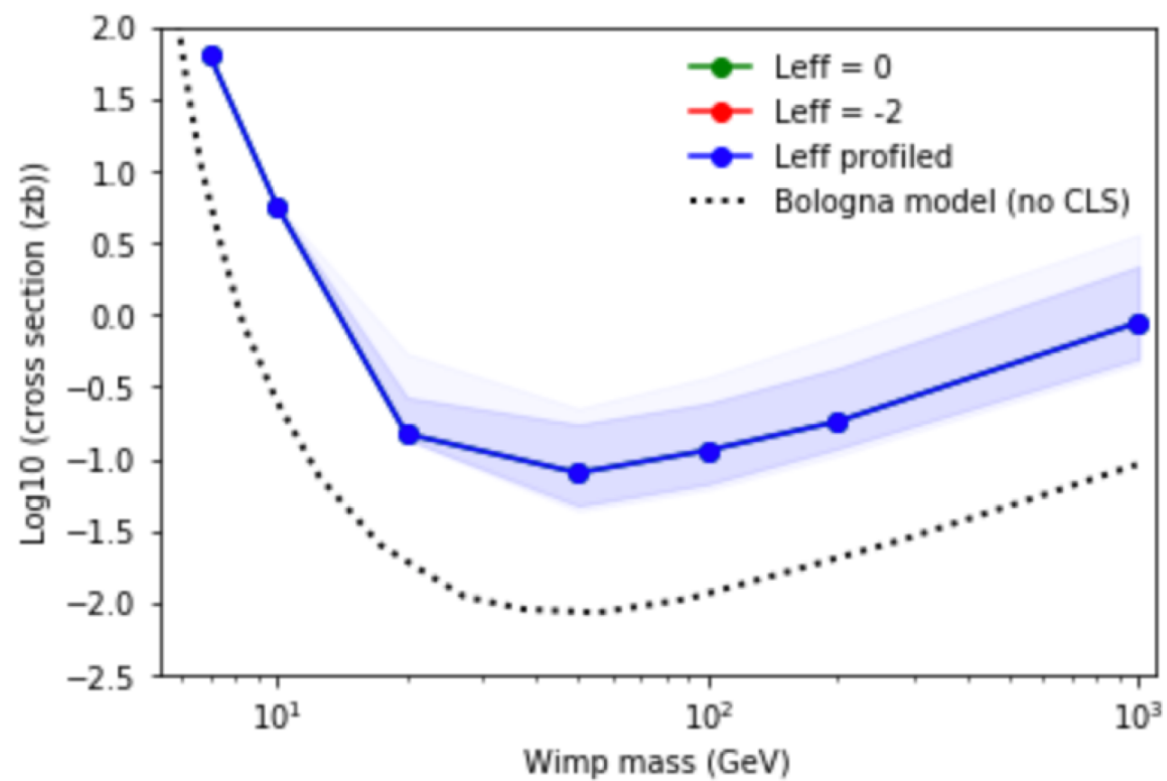


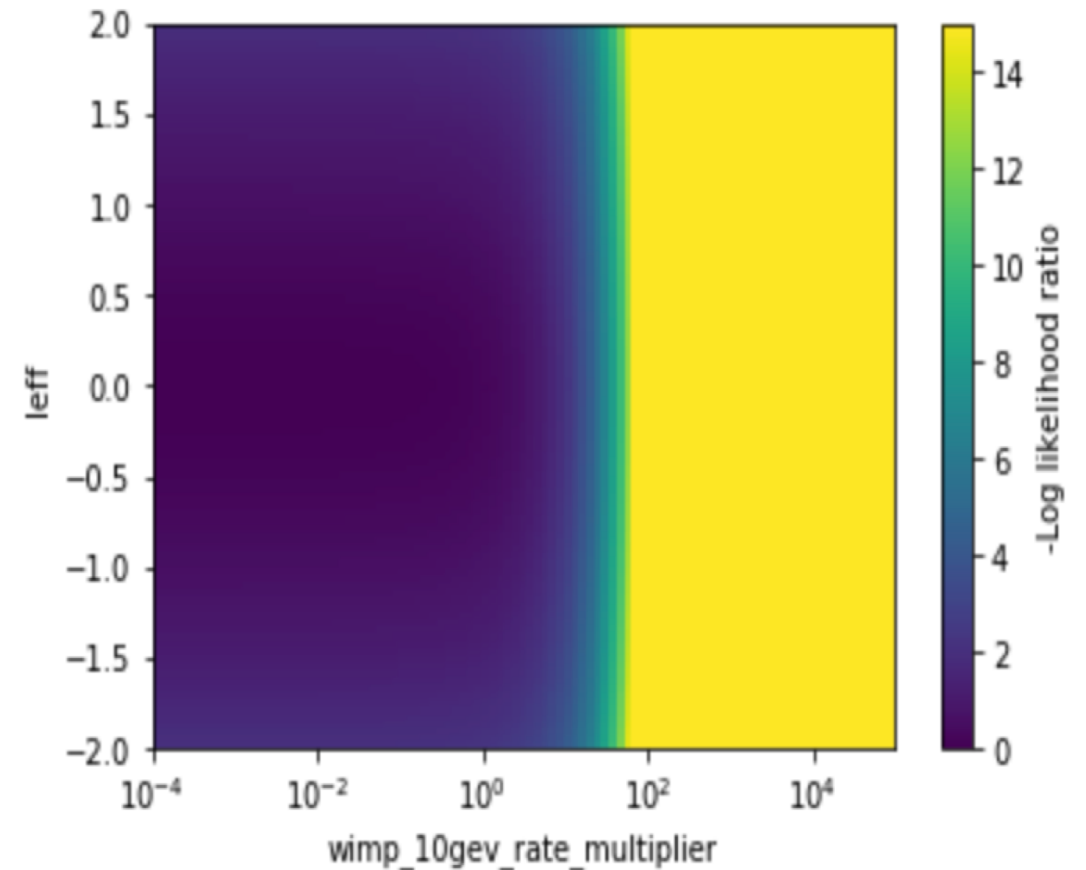
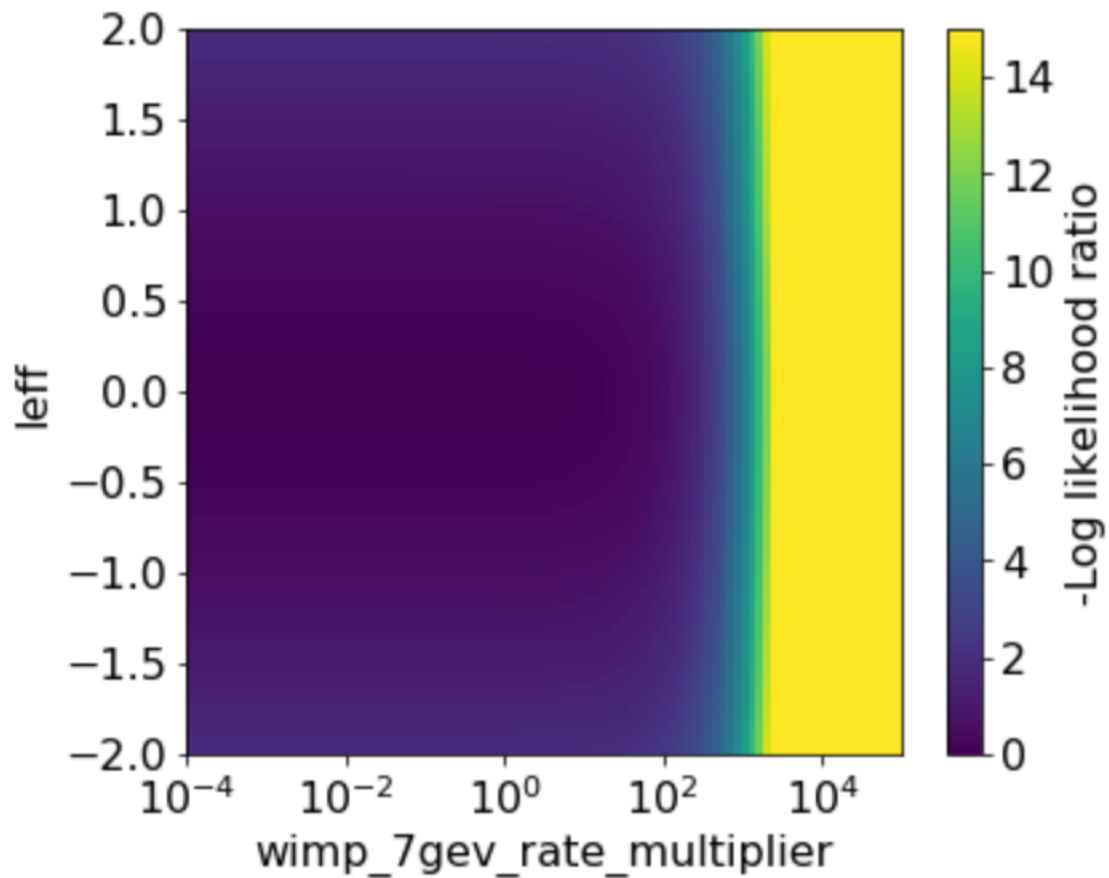
Table of Failures

trials

	100		Shape parametes of Leff	100	1000			
Below 150	Randomly							
Above 150	No errors							
			fail					
				Mostly				
					Fails			

Change the scale $(-4,1) \rightarrow (-4,5)$

Thanks, Jonathan !



Test on 50 GeV WIMP

1000 trials

Simulate(blueice) the background-only data using the likelihood function of 50 GeV and multiply by the rate of 50 GeV

A list of background data [0....999]

Change the rate of the source

Source name

multiplier

```
bg_dsets = [lfs[50].base_model.simulate(rate_multipliers=dict(wimp_50gev=0))  
            for _ in tqdm(range(n_trials))] #a list blueice.model
```

Choose ONE background dataset from the list

Set the likelihood function(50 GeV in this case) to the background data

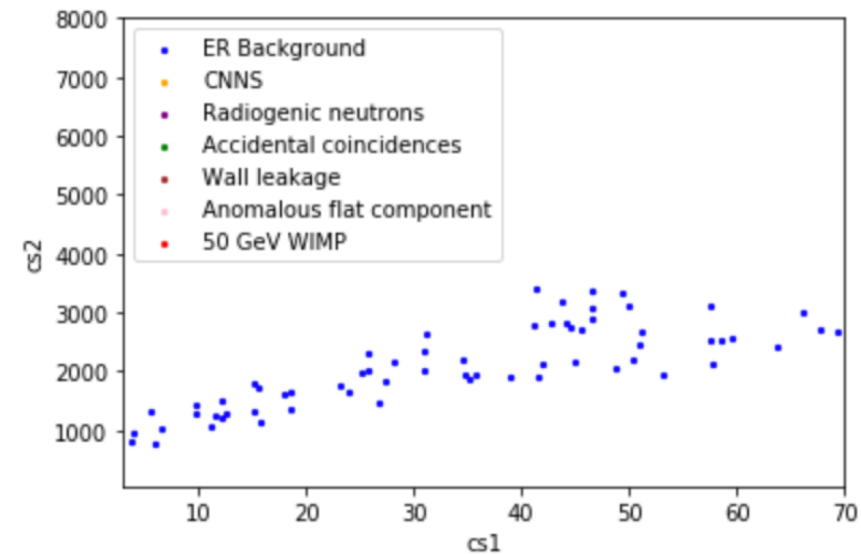
Plot ONE set of background+signal data

Set the log likelihood function of 50 GeV to the background dataset

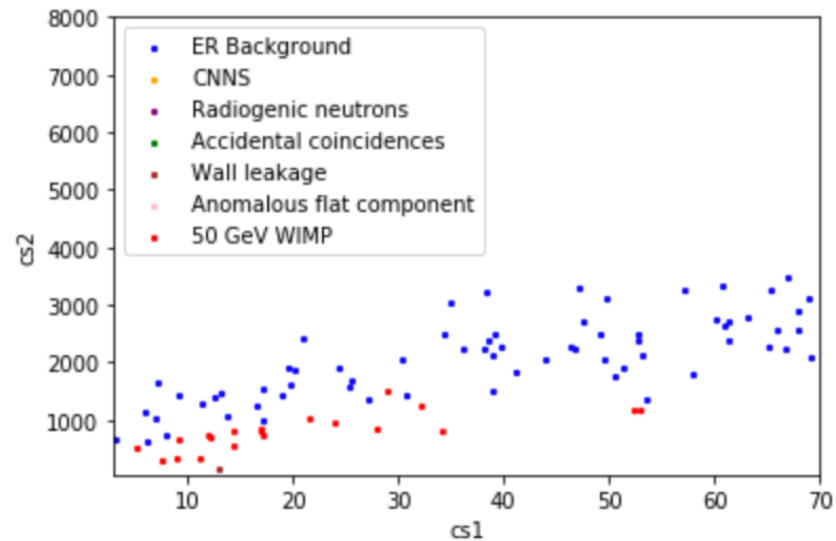
Use the syntax from blueice, `plot_likelihood_ratio`, plots the loglikelihood ratio with two changing parameters, `Leff` and `rate_multipliers`.

Rate_multipliers in background

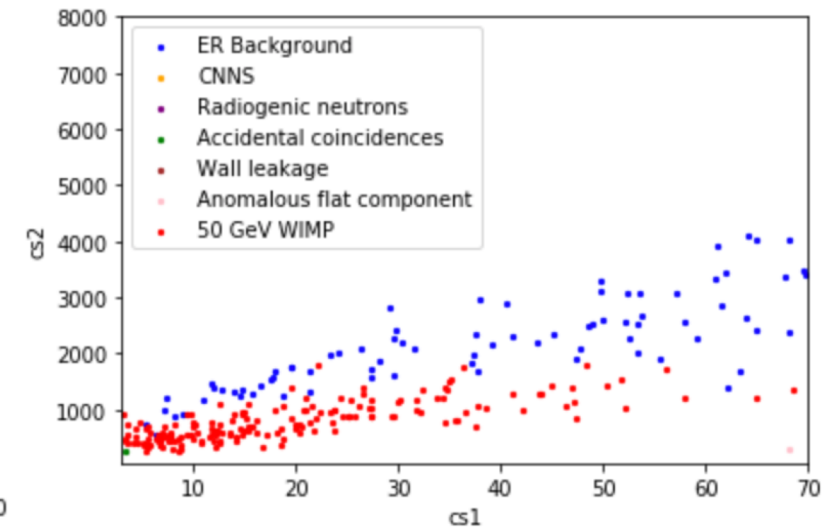
From step 2, 50gev=0



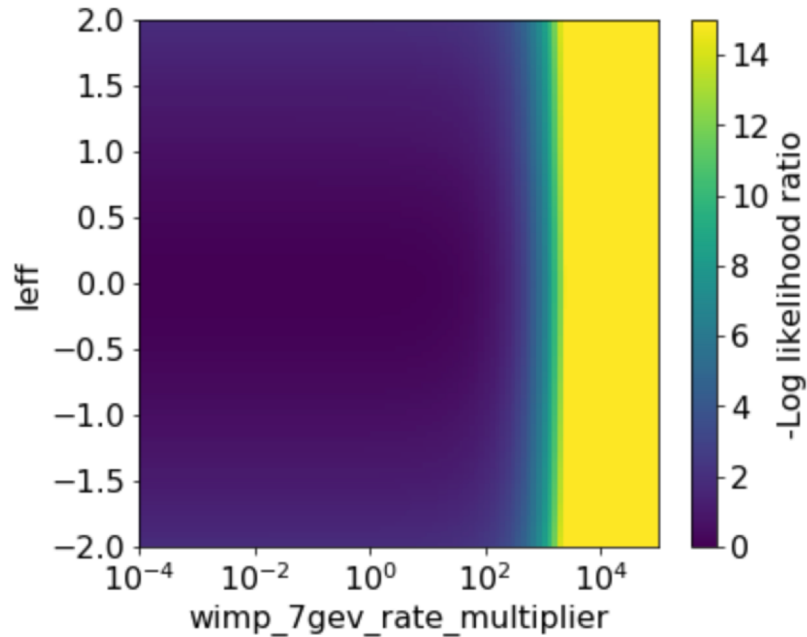
50gev = 1



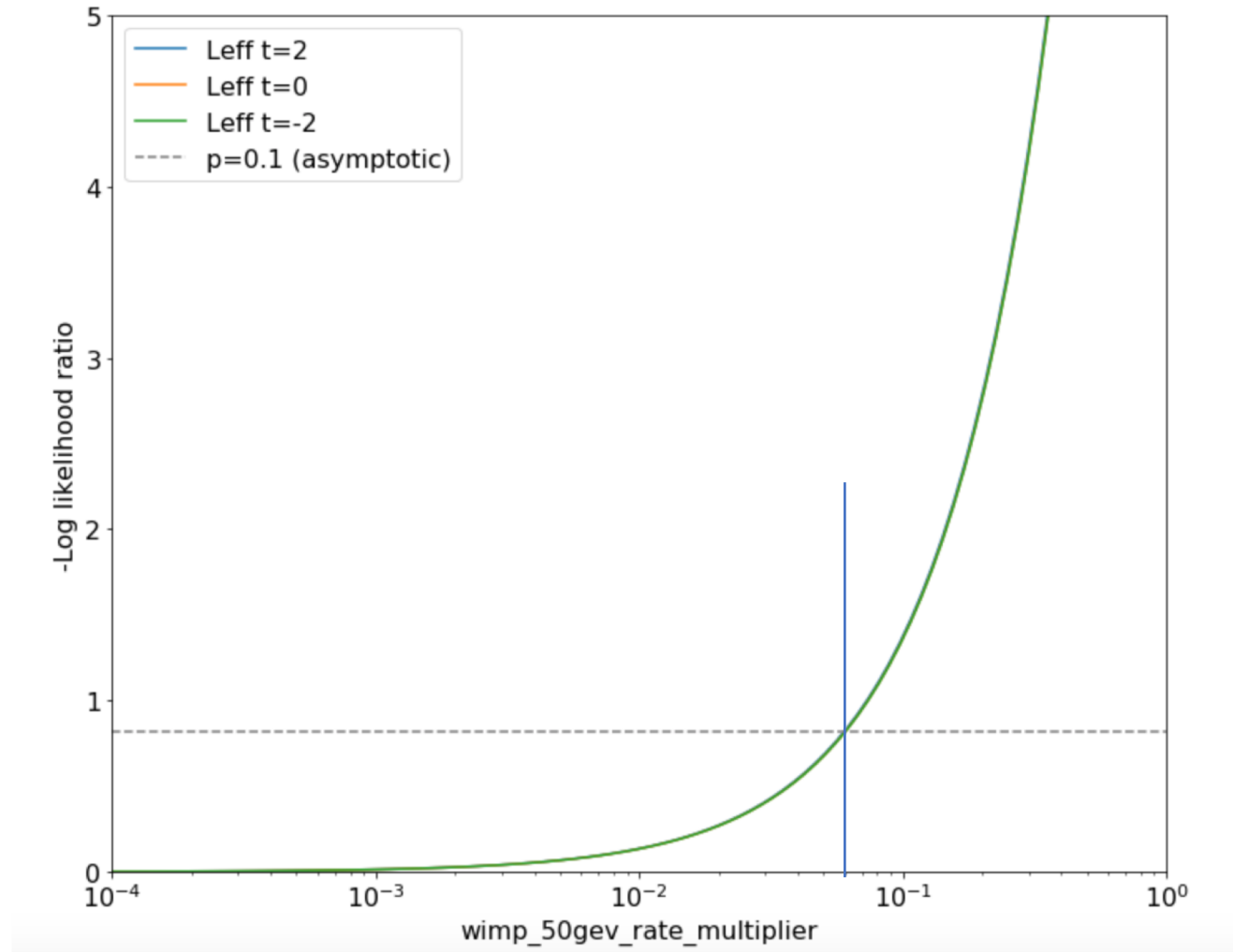
50gev = 10



Plotting in another way:

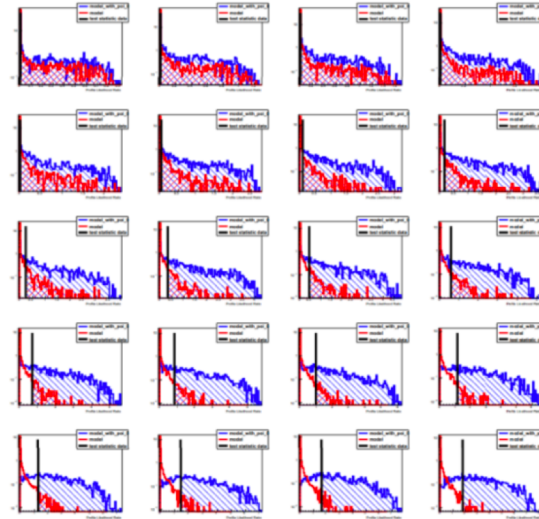


rate_multipliers is a list of rate multipliers for each source in self.source_name_list



Signal_events = rate_multiplier * number_expected_events

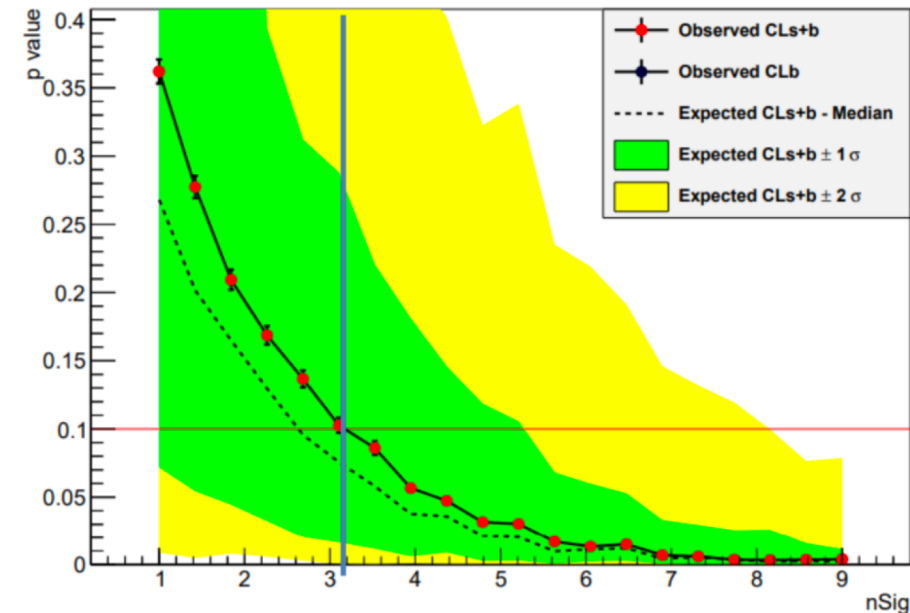
Limit Setting for One Mass Illustration



$$\bullet \lambda = \frac{L(u=u_{\text{test}}, L_{\text{eff}}, \theta)}{L(u \neq u_{\text{test}}, L_{\text{eff}}, \theta)}$$

$\longrightarrow$ B+S
 $\longrightarrow$ Only B

Different run than before



Intersects at ~3.2 signal events so
our limit is the value of POI that
yields an expected 3.2 signal events

Setting the limits

1. create 3 dictionaries: {m: empty N-dimensional arrays(number of trials, in this case, it is 1000)} m is the wimp mass.

- Limits on likelihood ratio
- Limits on profile likelihood ratio
- Limits on likelihood ratio lowleff

2. Two for loops

for m in wimp_masses:

 lf = log likelihood function[m]

 for i, d in the background data sets:

 set lf into the backgournd data d

 target = wimp_mgev_rate_multiplier

 using three one_parameter_interval() to get the three limits

limits[m][i] = set a 90% confidence level interval of kind(upper)? on the parameter target of lf, with leff=0

limits_lowleff[50][i] = same as limits except that leff = -2

limits_plr = same as limits except no restrictions on leff

```
one_parameter_interval(lf, target, bound=100, confidence_level = 0.499, kind='upper', leff=0) * scale
```

Tests on the three limits: run 10 events, print them out

- Limits: {20: array([.....10 numbers]), 50 array([.....10 numbers]).....}
- Limits_lowleff: {20: array([0,0,0,...0,]), 50 array([same 10 numbers as in limits]), all zeros for 100, 200, 1000}
- Limits_plr {20: array([10 numbers slightly larger than limits]).....}

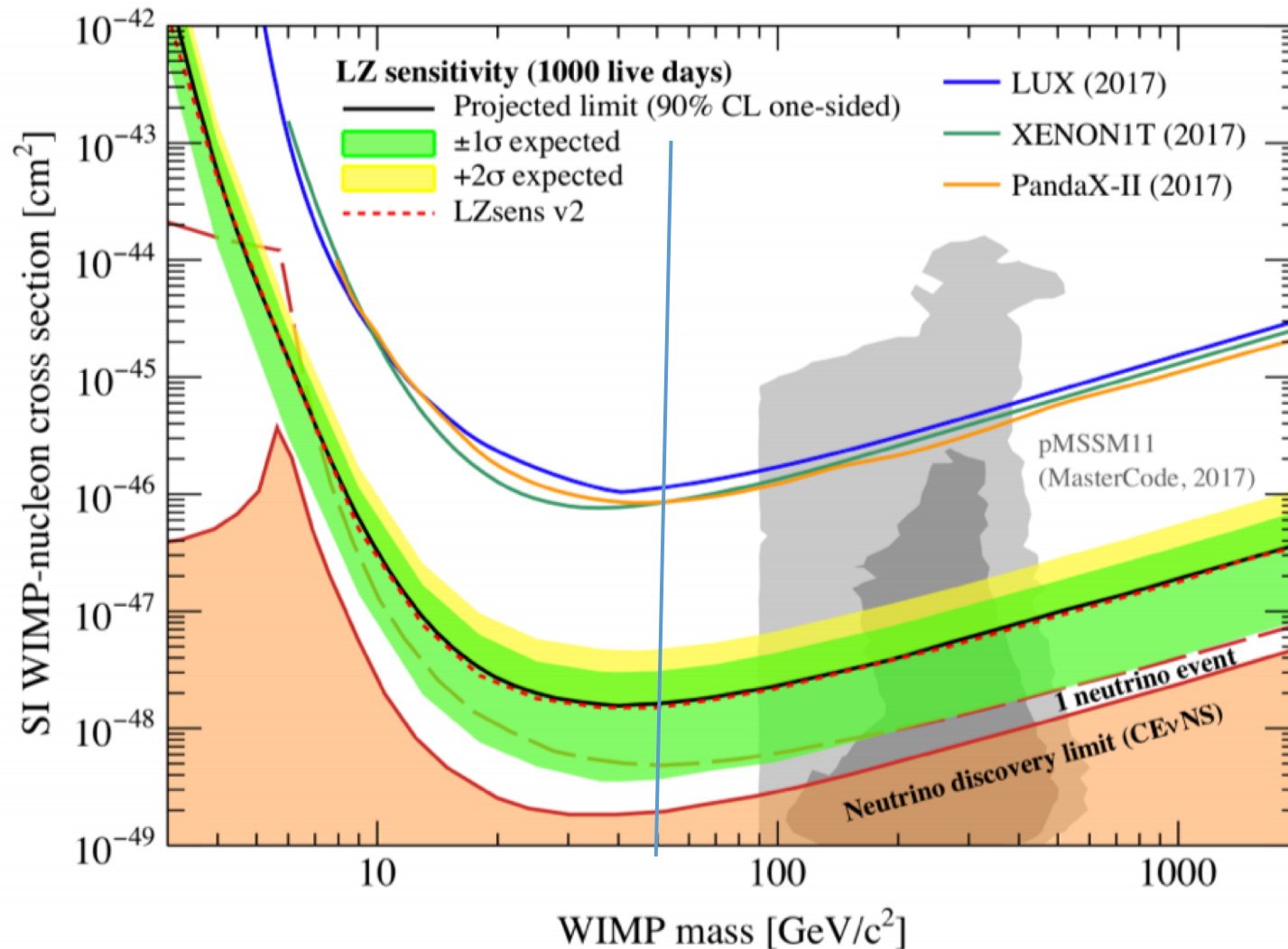
Comparison:

- In my code, in one background +signal data, varying rate_multiplier, get a number at $p=0.1$, 1000 limits for one mass, then, find the median of 1000 limits, and plot.
- In Shaun's slide, varying the datasets, get the number at $p=0.1$, 1 limit for one mass.

Same problems as before

- Errors for 7GeV, cannot run the code since the number is too low.
- Increasing the trails to 100000, it doesn't help.
- For 7GeV, it works when I changed the confidence level from 0.9 to 0.5, and only numbers around 0.5(testing from 10 ~10000 trials) not working below or above 0.5 too much
- Understand more how the confidence level affects the result

Why 50GeV?

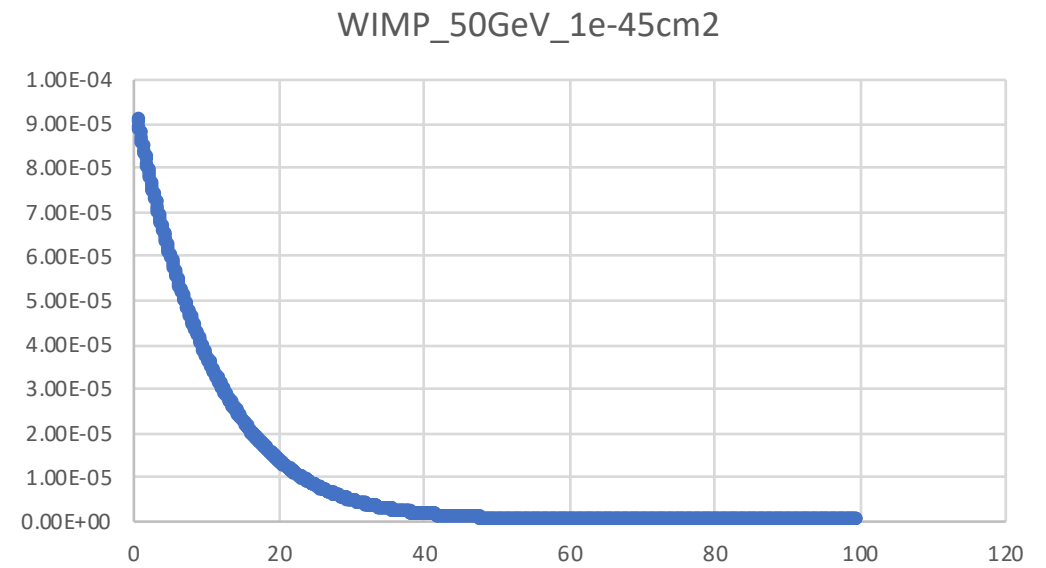
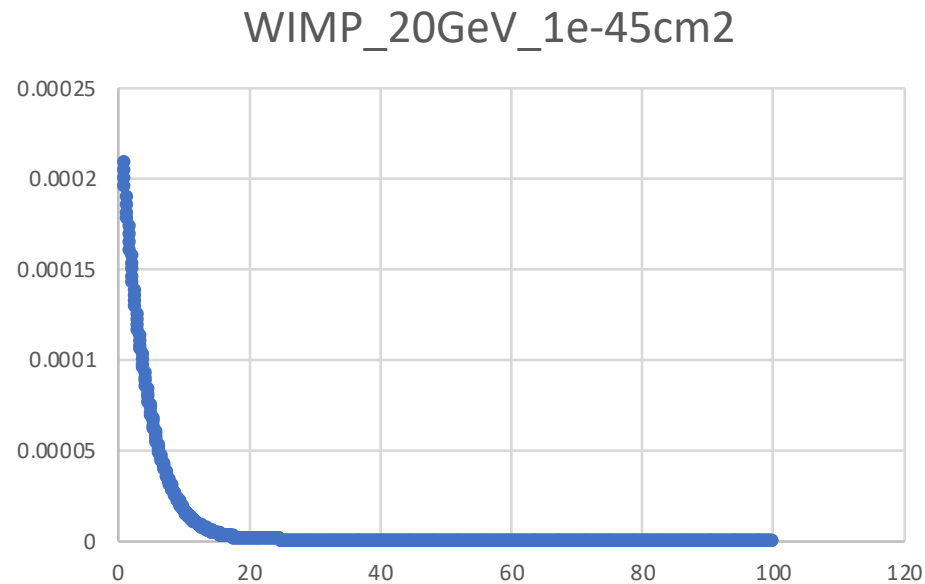
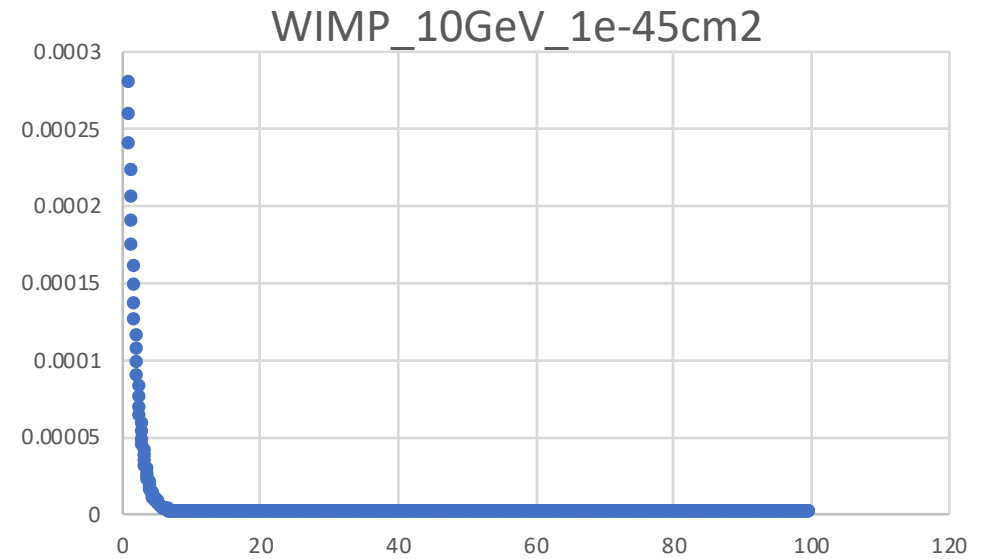
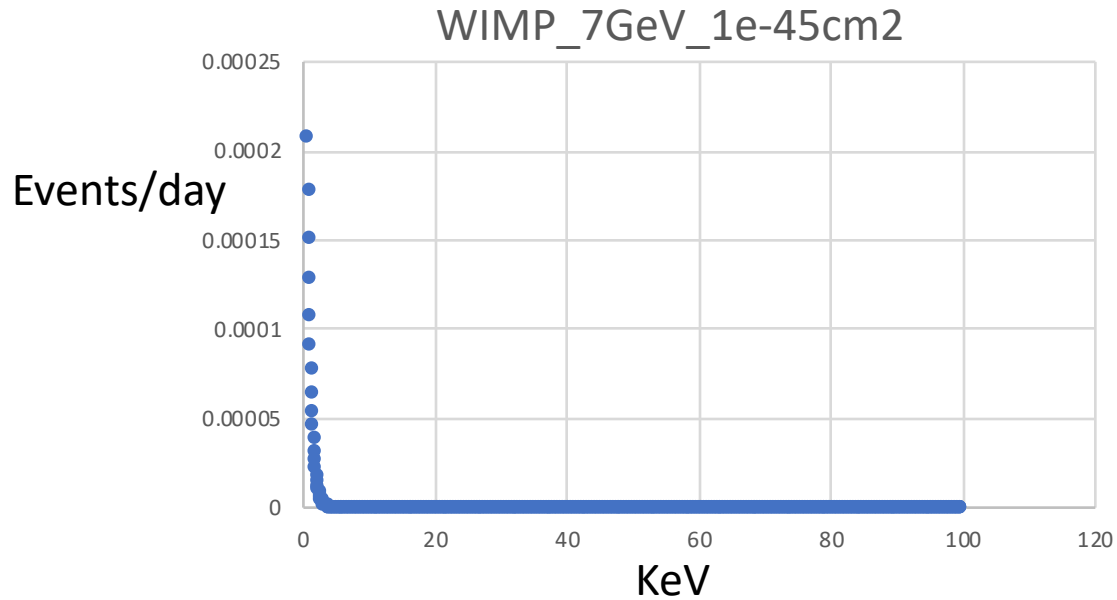


Just testing on 50 GeV? Change it to other values?

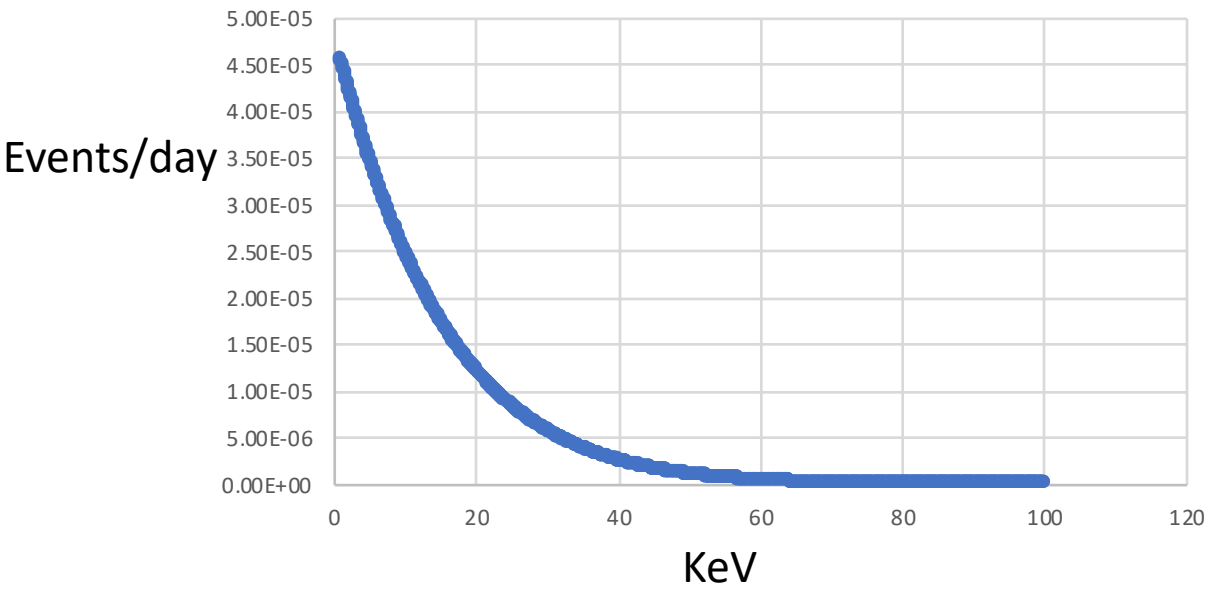
Understanding of the Code

- Create the likelihood function for different WIMP masses
 - Use the configuration from `base_model` and remove WIMPs source
 - Add each WIMP mass with cross section = $1\text{e-}45\text{cm}^2$ from the list(csv files) to the configuration

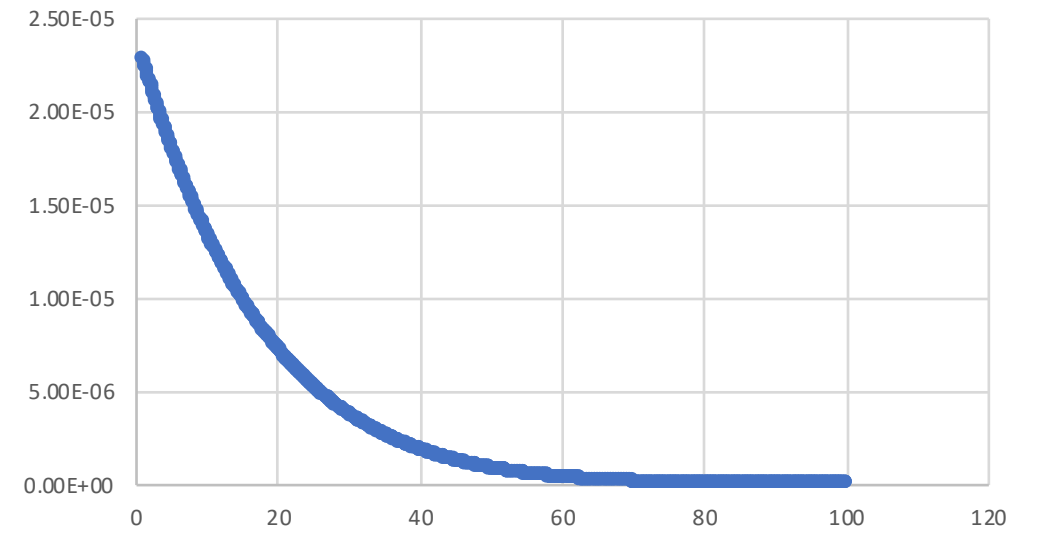
Plotted CSV files used for WIMP:



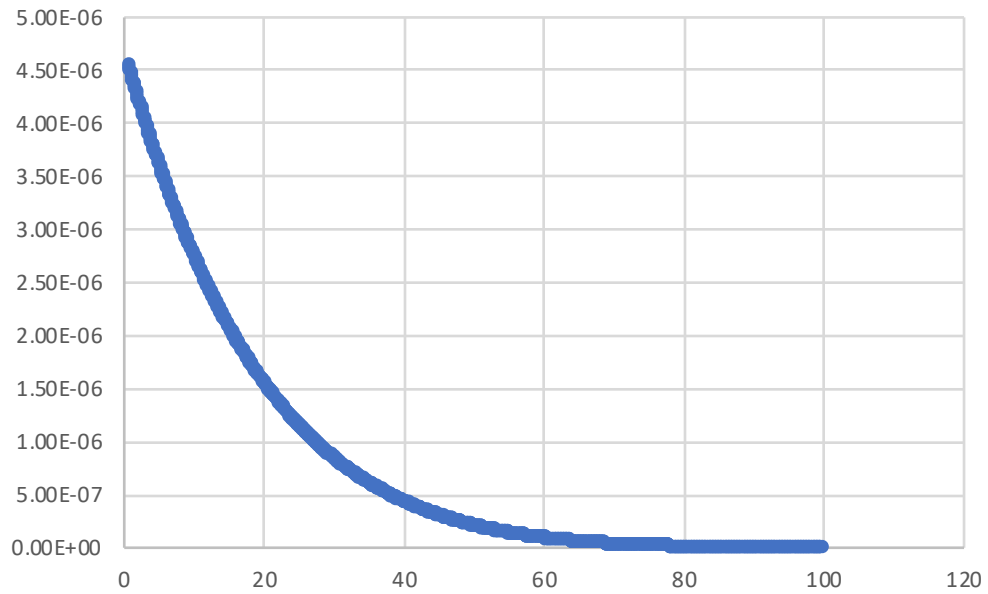
WIMP_100GeV_1e-45cm2



WIMP_200GeV_1e-45cm2



WIMP_1000GeV_1e-45cm2



Rate multiplier

- `rate_multipliers` is a list of rate multipliers for each source in `self.source_name_list`
- The rate parameter to the likelihood function is a multiplier for the *total* rate of WIMPs, (not just the rate of detectable events). This was done so we can easily convert to a cross section: just multiply by the reference cross section ($1\text{e-}45\text{ cm}^2$) for which the WIMP spectrum was generated.

- $\lambda = \frac{L(u=u_{test}, L_{eff}, \theta)}{L(u \neq u_{test}, L_{eff}, \theta)}$

$\longrightarrow$ B+S
 $\longrightarrow$ Only B

- Floating u_{test} and nuisance parameters to maximize the ratio

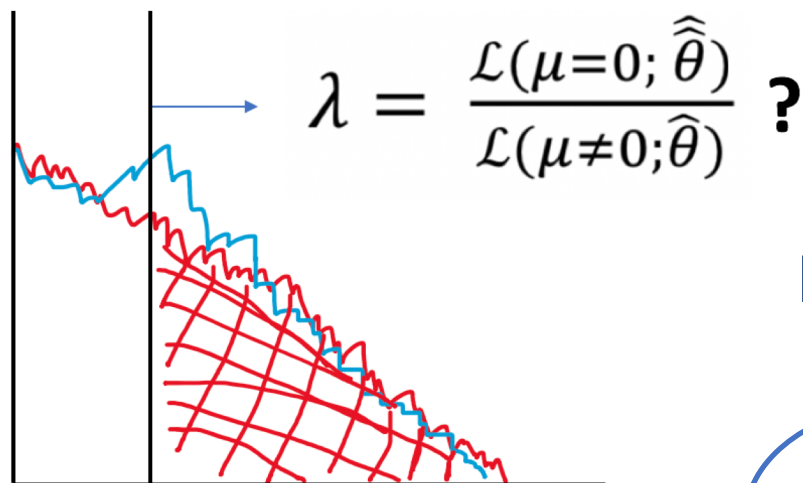
Background

LZ

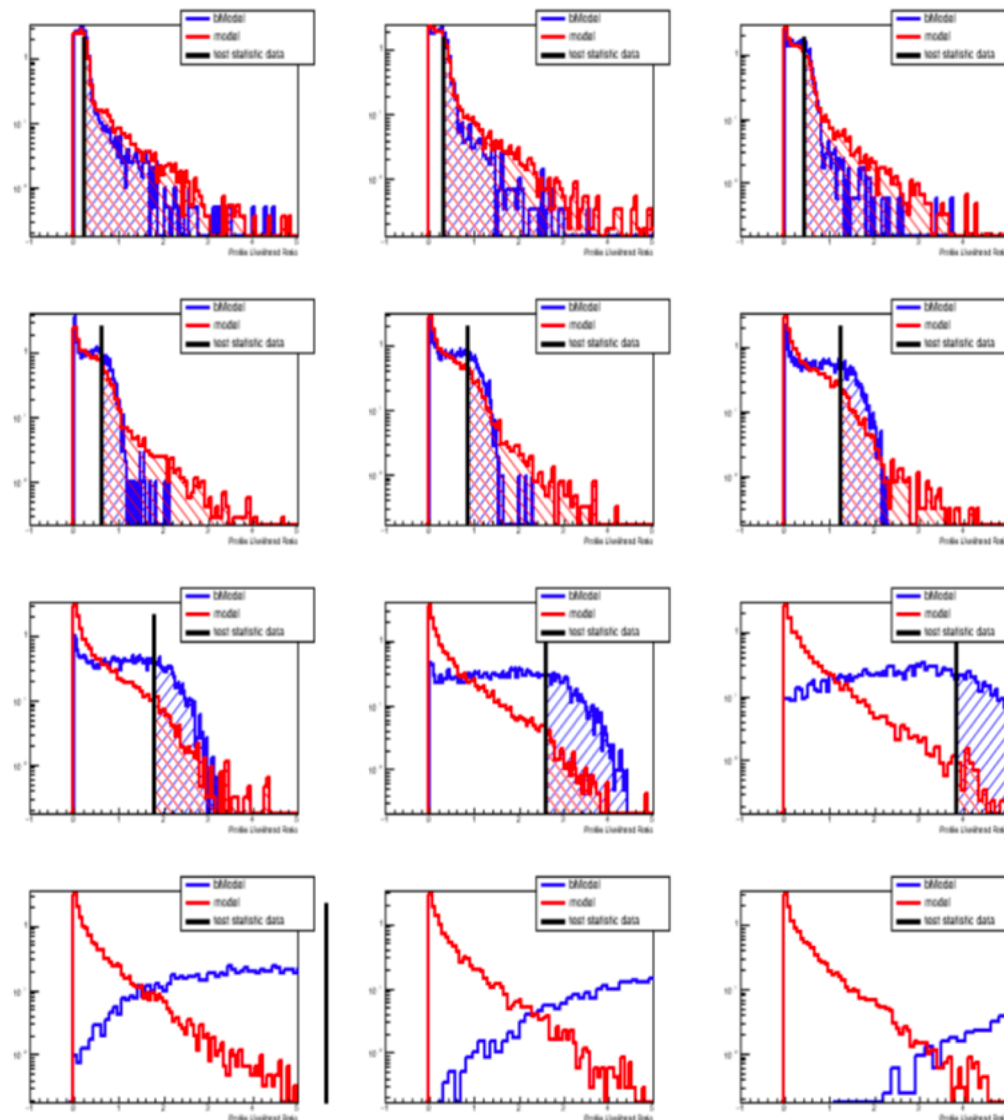
ER background		NR background	
Background		Background	σ/N
^{222}Rn (ER)	10%	^8B solar ν (NR)	15%
$pp+^7\text{Be}+^{14}\text{N}$ ν (ER)	2%	Det. + Env. (NR)	20%
^{220}Rn (ER)	10%	Atmospheric ν (NR)	25%
^{136}Xe $2\nu\beta\beta$ (ER)	50%	hep ν (NR)	15%
Det. + Env. (ER)	20%	DSN ν (NR)	50%
^{85}Kr (ER)	20%		

Xe1T

using the likelihood function of 50 GeV and multiply by the rate of 50 GeV



Varying POI (cross section in this case?)



Problem

- Set cross section = $1e^{-45}$, wimp interaction = SD, got an error.

```
~/Desktop/blueice/blueice/inference.py in one_parameter_interval(lf, target, bound, confidence_level, kind, bestfit_routine, t_ppf, **kwargs)
    376         return brentq(t, bound, global_best, args=[1 - confidence_level])
    377     elif kind == 'upper':
--> 378         return brentq(t, global_best, bound, args=[confidence_level])
    379
    380

/anaconda3/lib/python3.7/site-packages/scipy/optimize/zeros.py in brentq(f, a, b, args, xtol, rtol, maxiter, full_output, disp)
    508     if rtol < _rtol:
    509         raise ValueError("rtol too small (%g < %g)" % (rtol, _rtol))
--> 510     r = _zeros._brentq(f,a,b,xtol,rtol,maxiter,args,full_output,disp)
    511     return results_c(full_output, r)
    512
```

ValueError: f(a) and f(b) must have different signs

What I think.....

- Using Brent's method to find roots of the function and it requires $f(a)$ and $f(b)$ have different signs.
- Brent's Method: iteration of (bisection method + secant method)(I could make a slide on this method, if anyone is interested.....)
- When set CS to $1e^{-45}$, the likelihood is too small, and the function curve is around zero-axis.
- Solution: increase the numbers