Summary of Wall Test

- **Non-Separated Model**
  - **Method**
  - **Problem**
    - Create Workspace:
      - Interpolation order
      - Initial mu_sig value
      - Data file
    - LZStats:
      - Calculator type
  - **Output**
    - Parameter
    - Constraint function
    - Graph
      - Constrained vs. non-constrained
      - Varying constraint sigma, S1
    - Conclusion

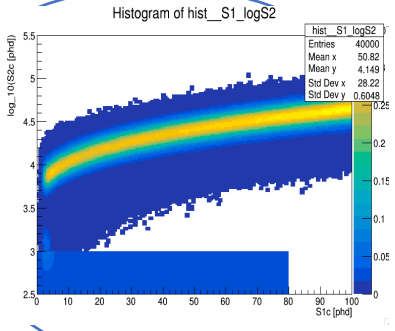
- **Separated Model**
  - **Method**
  - **Problem**
    - obs_data line
    - Initial fit
    - Failed toy
  - **Output**
    - Parameter
    - Constraint function
    - Graph
      - Limit vs. logS2 vs. rate

Non-Separated Model Method

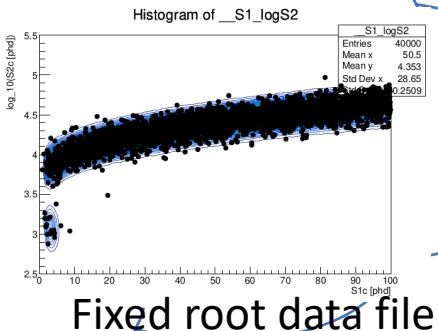
Test on the sensitivity mode on the workspace that does not have wall model added

Test on the sensitivity mode on the workspace that has wall model added (wall data generated randomly from LZStats)

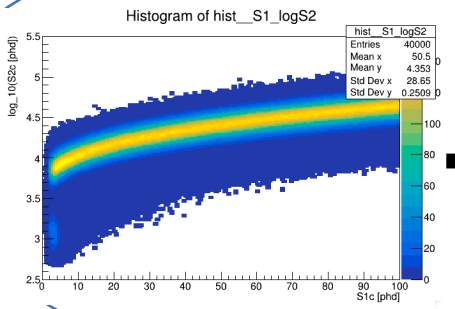
No wall model	No wall data
With wall model	With wall data
With wall model	No wall data
No wall model	With wall data



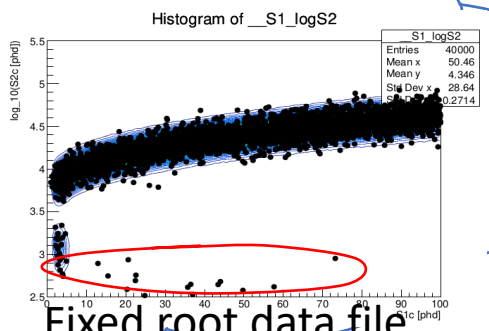
+



Fixed root data file with no wall data



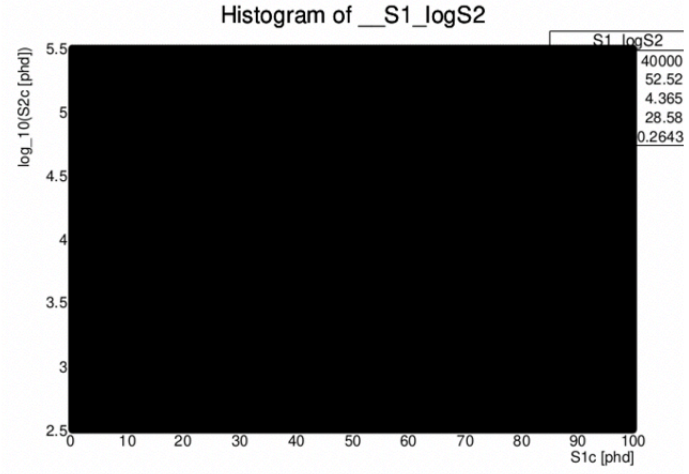
+



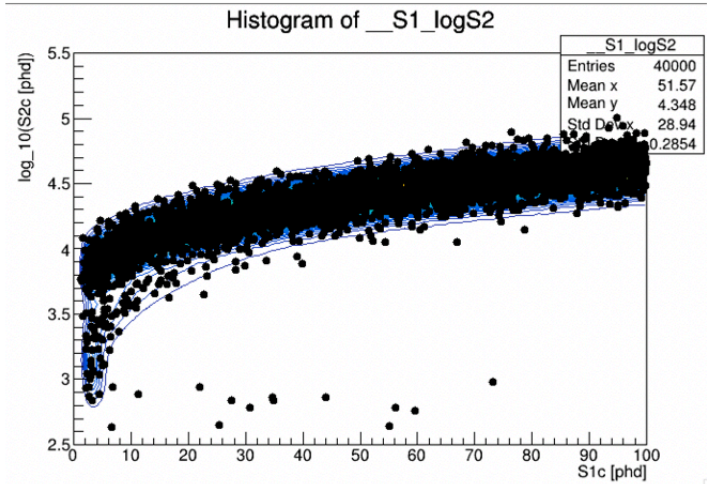
Fixed root data file with wall data

Problem: 0^{th} interpolation

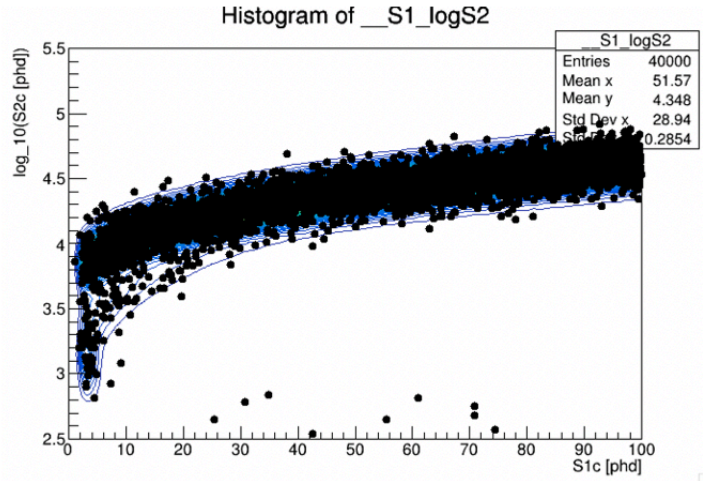
Data file generated from the workspace is totally black for the 0^{th} interpolation order



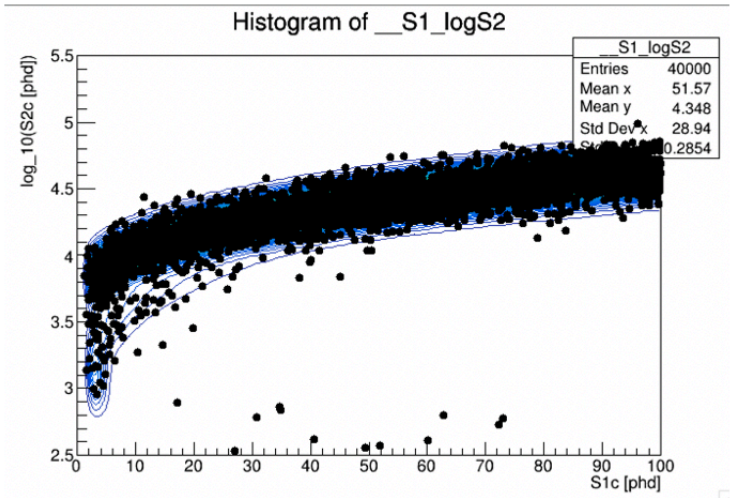
Interpolation order = 0



Interpolation order = 1



Interpolation order = 2

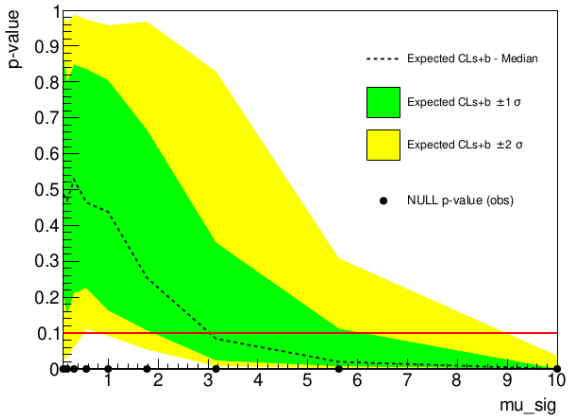


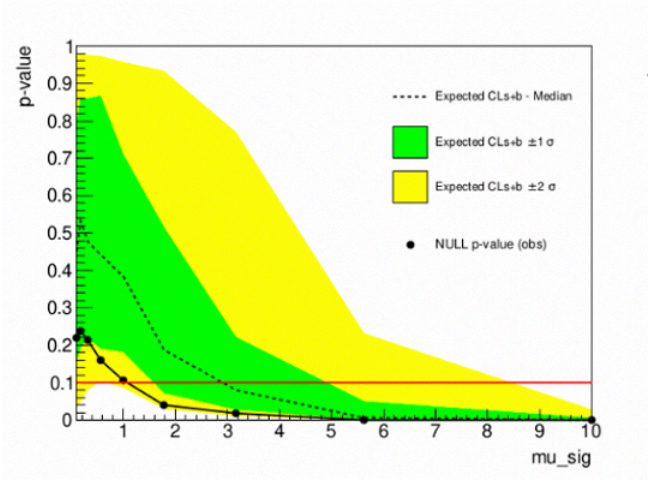
Interpolation order = 3

Initial mu_sig

- **Problem:** For the with wall model, no wall data case, the obs_data line is constant at zero
- **Analysis:** since there is no wall data, but there is a wall model, so when running LZStats, it creates more mu_sig events to compensate for the difference
- **Solution:** set initial mu_sig value to 0 in the create_workspace code

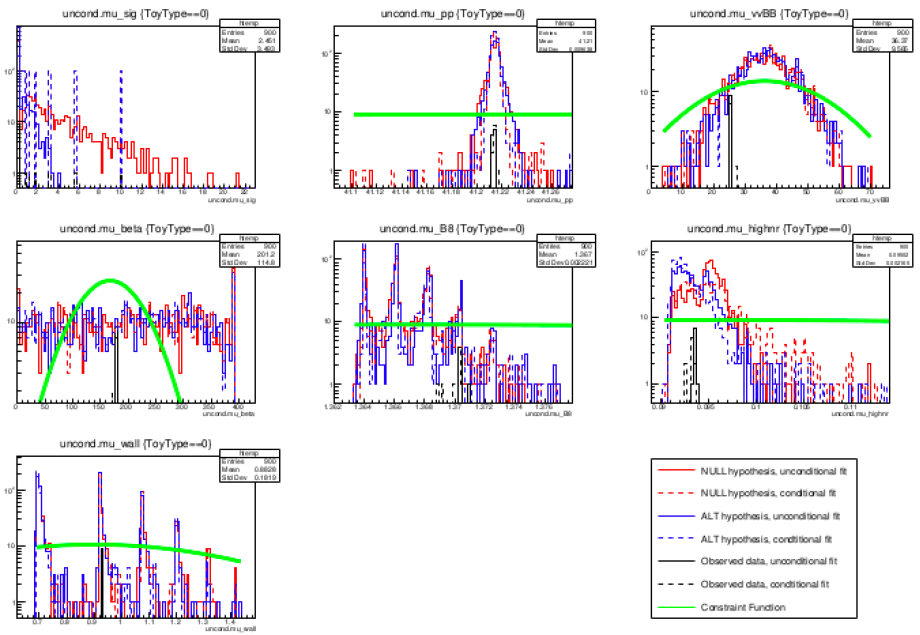
```
mu_B8      = 1.36605      +/-  0.0556527 (limited)
mu_beta    = 163.847      +/-  24.5343   (limited)
mu_highnr          = 29.9979      +/-  1.95864   (l
mu_pp      = 0.00842288   +/-  0.000168463       (l
mu_sig     = 105.319      +/-  10.2295   (limited)
mu_vvBB    = 40.251       +/-  21.0395   (limited)
mu_wall    = 0.697606     +/-  0.383162  (limited)
```



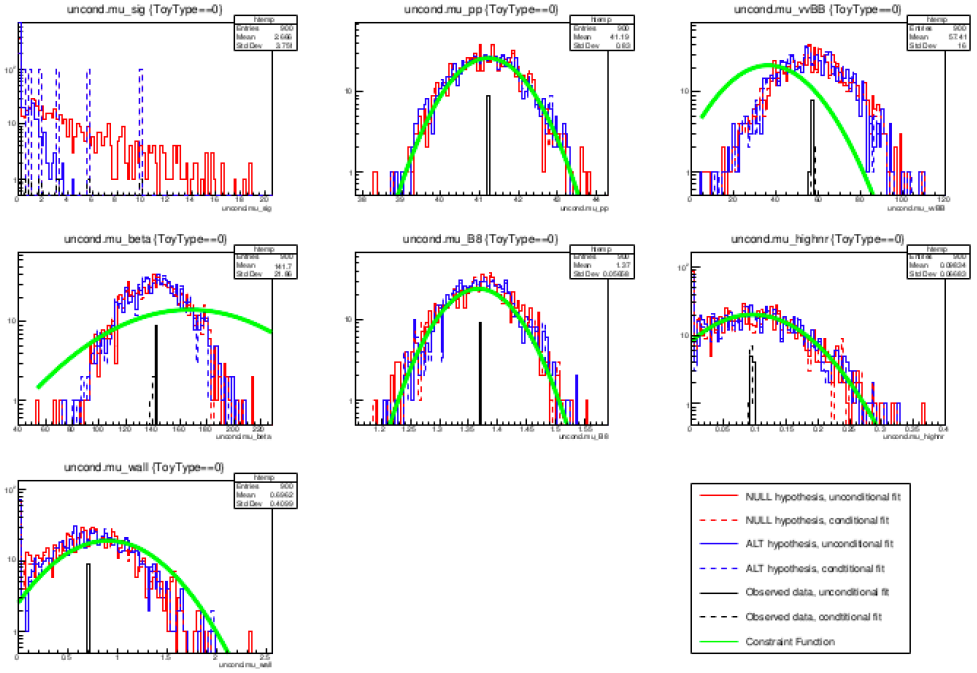


Calculator Type

- Run LZStats with calculator type 1, with high toys (~500) crashed in the middle, works fine for lower toy numbers, 100 toys



With Model, with data
Calculator Type = 1



With model, with data
Calculator Type = 0

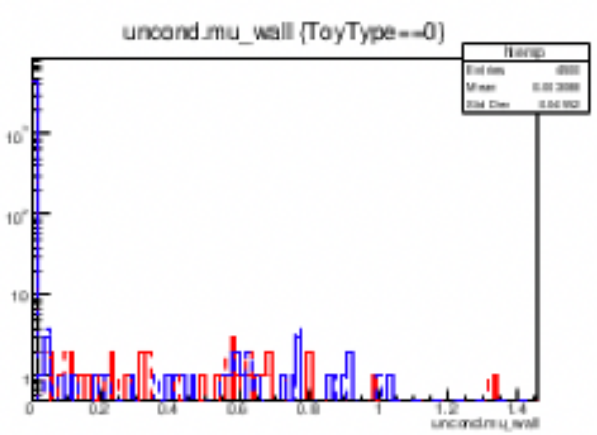
Output: Parameter

- **Workspace:**
  - Livedays = 60d
  - WIMP_mass = 40 GeV
  - Interpolation order = 1
  - Wall model:
    - Rate = 0.015 events/day
    - S1 = 80
    - logS2 = 3

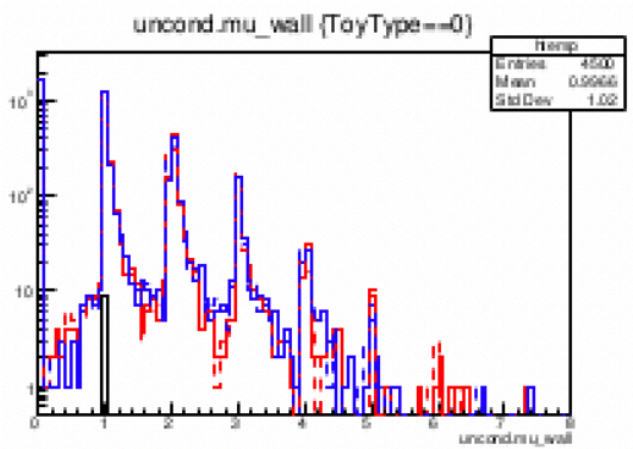
LZStats:

- MIN_POI = 0.1
- MAX_POI = 10
- Ntoys = 500
- LogScan
- N_POI_Scan = 9
- Calculator Type = 0
- Two-sided test

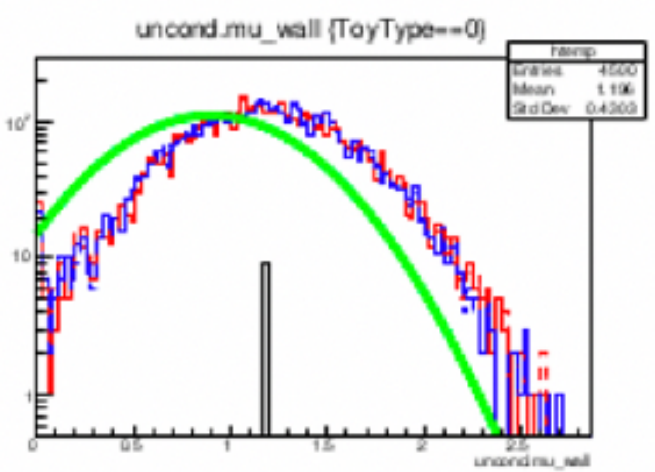
Constraint



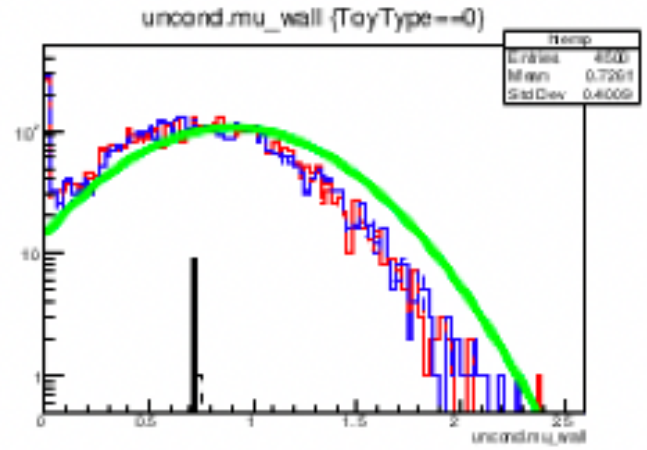
No constraint
NDWM



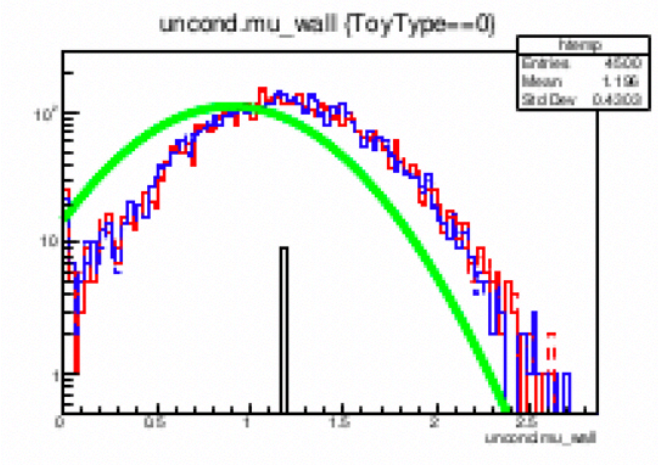
No constraint
WDWM



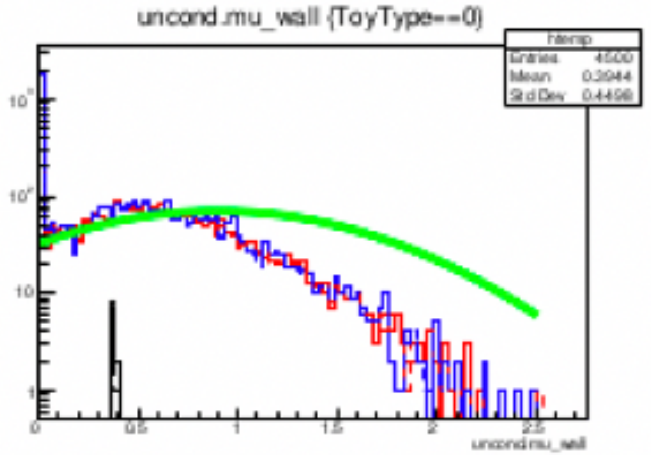
Sigma = 0.5



With constraint
NDWM



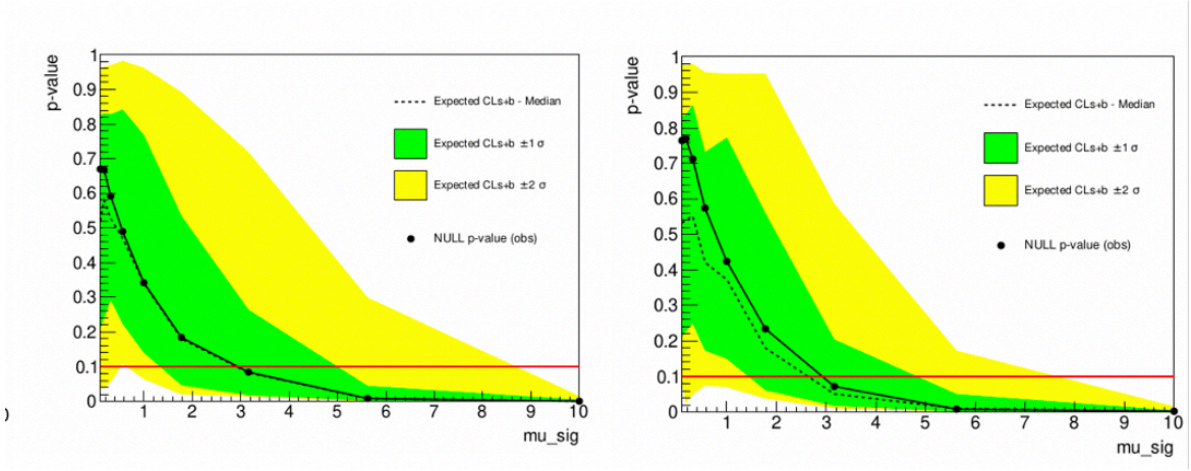
No constraint
WDWM



Sigma = 0.8

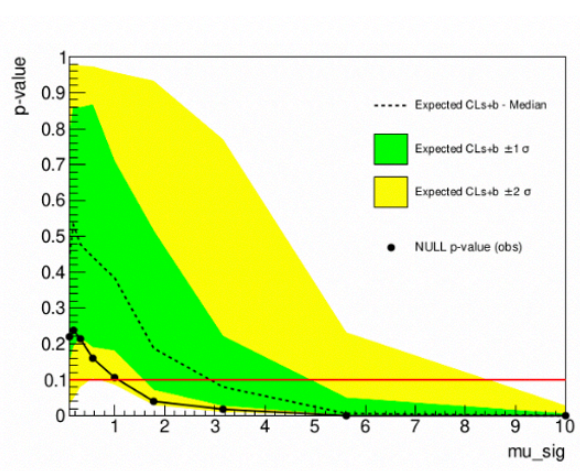
Graph

- For **WDWM, NDNM**, obs_data gives **similar limit** as the expected median line
- For **NDWM**, obs_data gives a **better limit** than the expected median line
  →The datafile does not have wall data in it, but since the workspace has a wall model, each toy has wall data in it
- For **WDNM**, obs_data gives a **worse limit** than the expected median line
  → The wall data in the data file will be treated as signal data

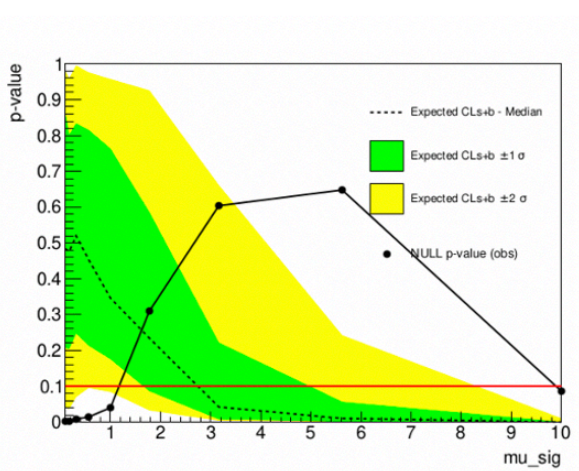


With wall data
With wall model(WDWM)

No wall data
No wall model(NDNM)



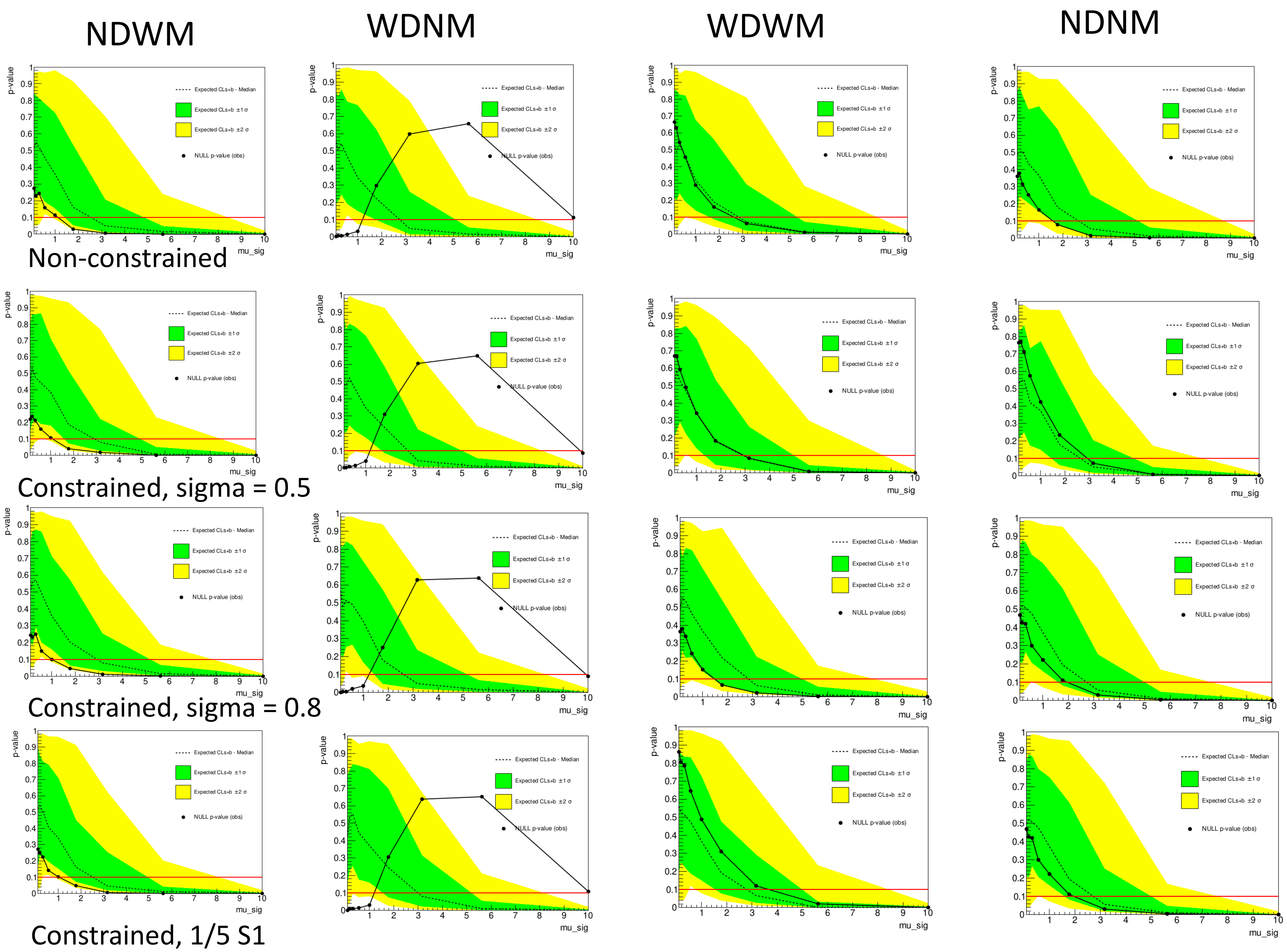
No wall data
With wall model(NDWM)



With wall data
No wall model(WDNM)

10 runs for each case and each plot is randomly chosen from the 10 outputs

Comparison of the output on each case with different conditions



Conclusion

The obs_data line gives similar upper limit as the best-fit line and fluctuates within the 1-sigma band

No wall model	No wall data
With wall model	With wall data
With wall model	No wall data
No wall model	With wall data

1. The obs_data line gives a better limit than the expected median line and fluctuates at the lower bound (2-sigma band)
2. The datafile does not have wall data in it, but since the workspace has a wall model, each toy has wall data in it

1. If the wall data is around the ^{8}B region, get a significantly worse limit; if the wall data is far from the ^{8}B region, get an error running LZStats
2. However, the problems can be solved by adding a wall model :back to the case of with both wall model and wall data case → gives a better limit and runs LZStats successfully

Separated Model Method

- Motivation: With low rate, sometimes no generated data in the area of interest; only tests on one data file, the result is not very trustworthy
- Separate the model that generates dataset and the model that is used for fitting
- Same process for NDNM & WDWM cases in non-separated model cases
- For WDNM & NDWM cases, the data file will be generated randomly for fitting instead of fixed data file,

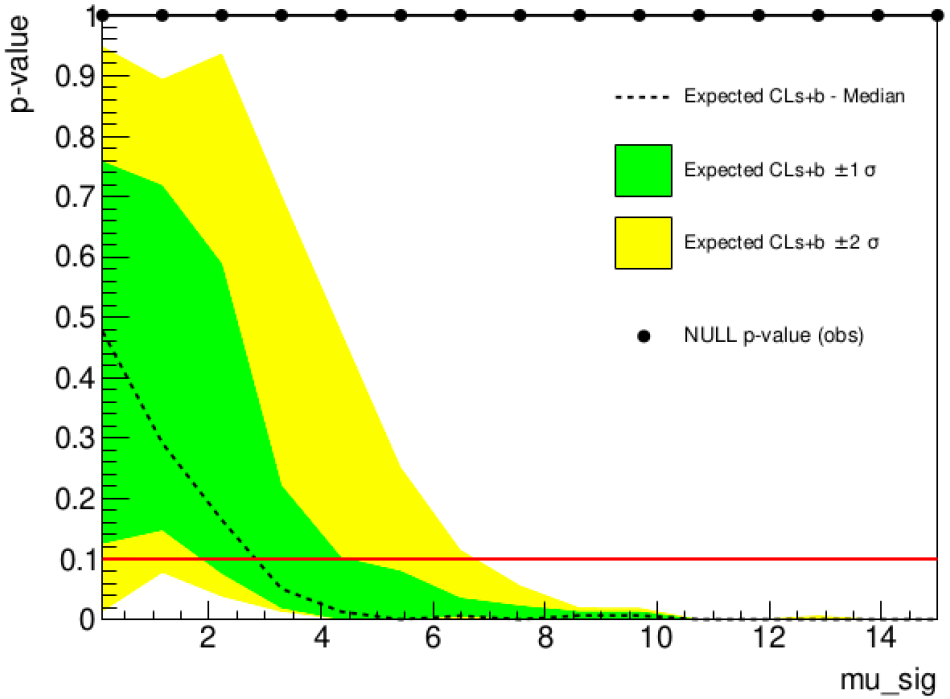
Problem: obs_data line

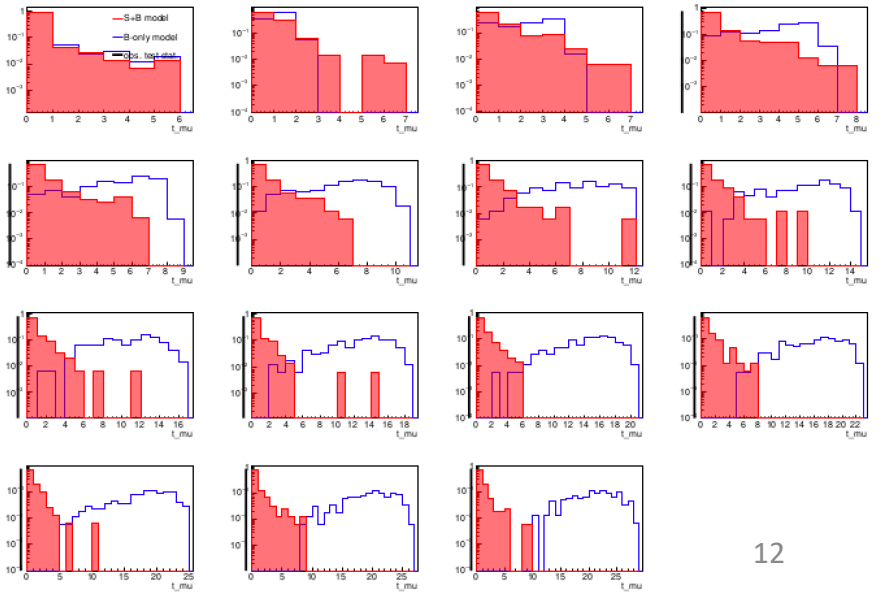
Problem:

1. Constant Obs_data line at p-value = 1
2. Negative Obs_data line in TS plot

Solution:

Set *bool use_observed_data = false* in analyse_output.C



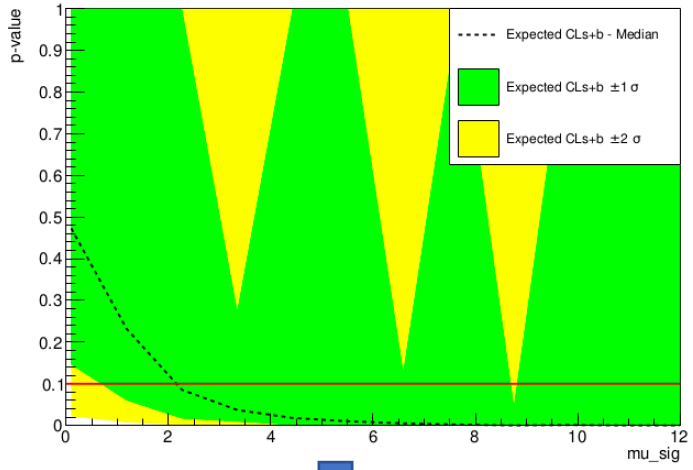


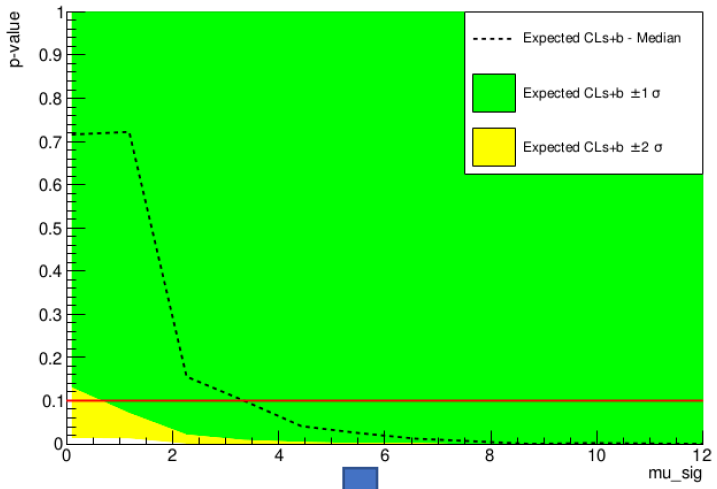
Initial fit

- Crash → Set *RUN_INITIAL_FIT: false* in Parameters.yaml

```
------------------------------------------------------------
There was a crash.
This is the entire stack trace of all threads:
============================================================
#0  0x00007f74eae7046c in waitpid () from /lib64/libc.so.6
#1  0x00007f74eadedf62 in do_system () from /lib64/libc.so.6
#2  0x00007f74f0d5ea3d in TUnixSystem::StackTrace() () from /cvmfs/lz.opensciencegrid.org/external/ROOT/6.20.00/x86_64
-centos7-gcc7-opt/lib/libCore.so
#3  0x00007f74f0d612d4 in TUnixSystem::DispatchSignals(ESignals) () from /cvmfs/lz.opensciencegrid.org/external/ROOT/6
.20.00/x86_64-centos7-gcc7-opt/lib/libCore.so
#4  <signal handler called>
#5  0x00007f74ebc45b07 in ProfileLikelihoodTestStatMod::EvaluateProfileLikelihood (this=0x59c7e60, type=0, data=..., p
aramsOfInterest=...) at src/ProfileLikelihoodTestStatMod.cxx:194
#6  0x00007f74ebf895ed in RooStats::ToyMCSampler::EvaluateAllTestStatistics(RooAbsData&, RooArgSet const&, RooStats::D
etailedOutputAggregator&) () from /cvmfs/lz.opensciencegrid.org/external/ROOT/6.20.00/x86_64-centos7-gcc7-opt/lib/libR
ooStats.so
#7  0x00007f74ebf898b2 in RooStats::ToyMCSampler::EvaluateAllTestStatistics(RooAbsData&, RooArgSet const&) () from /cv
mfs/lz.opensciencegrid.org/external/ROOT/6.20.00/x86_64-centos7-gcc7-opt/lib/libRooStats.so
#8  0x00007f74ebf189aa in RooStats::HypoTestCalculatorGeneric::GetHypoTest() const () from /cvmfs/lz.opensciencegrid.o
rg/external/ROOT/6.20.00/x86_64-centos7-gcc7-opt/lib/libRooStats.so
#9  0x00007f74ebf1b3e6 in RooStats::HypoTestInverter::Eval(RooStats::HypoTestCalculatorGeneric&, bool, double) const (
) from /cvmfs/lz.opensciencegrid.org/external/ROOT/6.20.00/x86_64-centos7-gcc7-opt/lib/libRooStats.so
#10 0x00007f74ebf1c3cd in RooStats::HypoTestInverter::RunOnePoint(double, bool, double) const () from /cvmfs/lz.opensc
iencegrid.org/external/ROOT/6.20.00/x86_64-centos7-gcc7-opt/lib/libRooStats.so
#11 0x00007f74ebf1d01d in RooStats::HypoTestInverter::RunFixedScan(int, double, double, bool) const () from /cvmfs/lz.
opensciencegrid.org/external/ROOT/6.20.00/x86_64-centos7-gcc7-opt/lib/libRooStats.so
#12 0x00007f74ebf1fb20 in RooStats::HypoTestInverter::GetInterval() const () from /cvmfs/lz.opensciencegrid.org/extern
al/ROOT/6.20.00/x86_64-centos7-gcc7-opt/lib/libRooStats.so
#13 0x0000000000419545 in RooStats::HypoTestInvTool::RunInverter(RooWorkspace*, char const*, char const*, char const*,
 int, int, bool, bool, int, double, double, int, bool, char const*, bool, bool) ()
#14 0x000000000041bb3f in StandardHypoTestInv(char const*, double, int, double, double, int, int, int, char const*, ch
ar const*, char const*, bool, bool, bool, char const*, char const*, bool, bool, char const*) ()
#15 0x000000000041e38d in run_lzstats(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >,
 char const*, int, float, int, char const*) ()
#16 0x000000000040f9e0 in main ()
============================================================
```

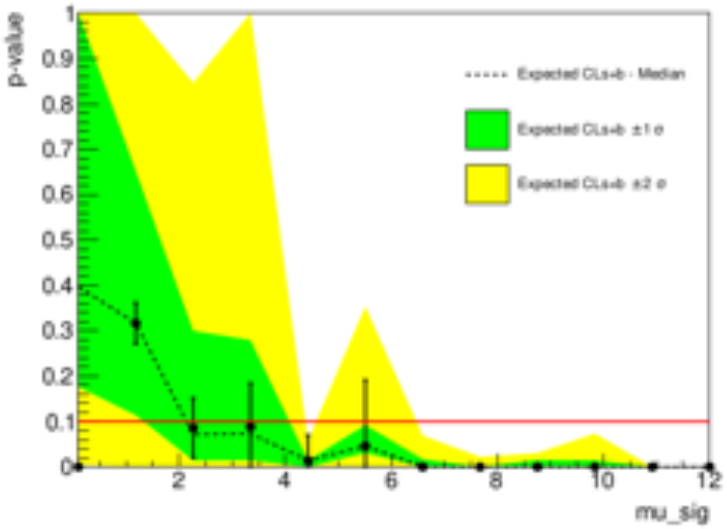
High failed event (WDNM)

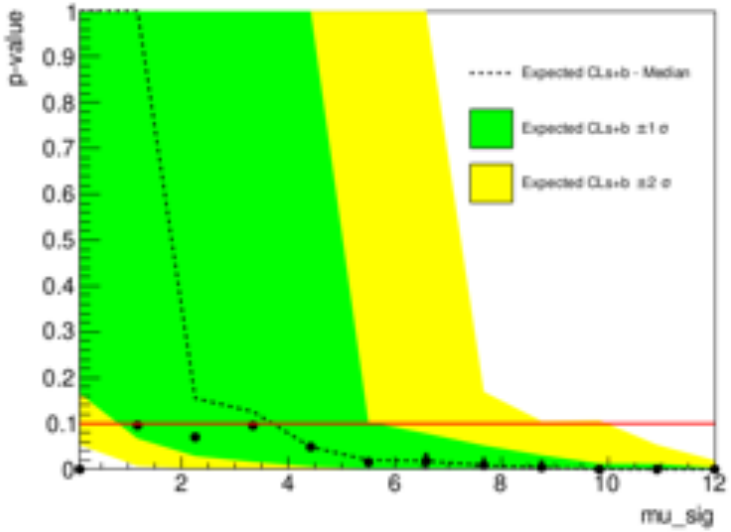




Raw output from LZStats
10 % ~ 17% failed

After run_analyse_output.sh
→ ignore failed events





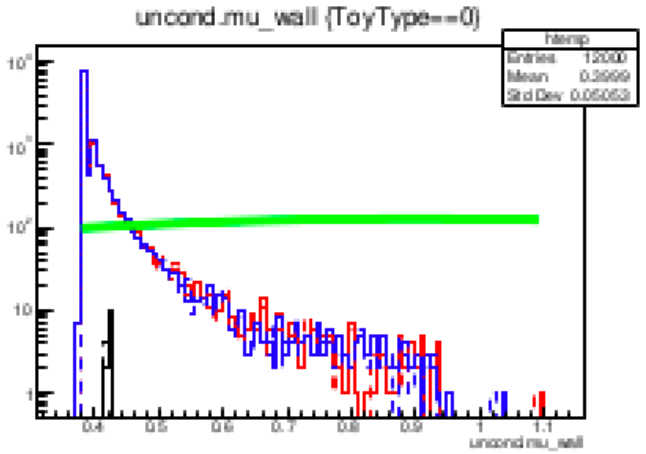
Output: Parameter

- **Workspace:**
  - Livedays = 60d
  - Rate = 0.015 events/day
  - WIMP_mass = 40 GeV
  - Interpolation order = 1
  - Wall model:
    - Data will generate from the overlapping region with B8
    - Rate = 0.015 events/day
    - S1 = 16
    - logS2 = 3

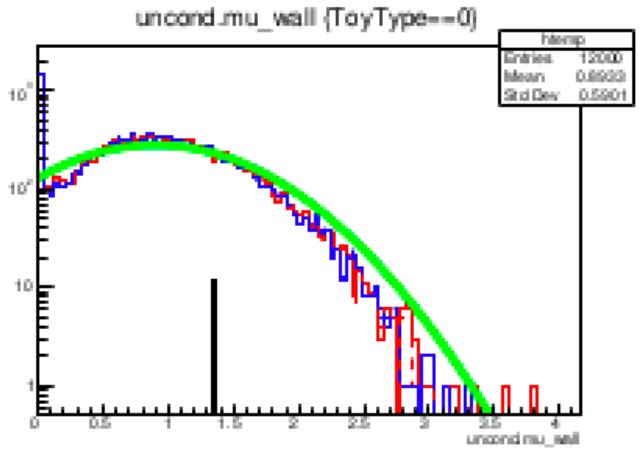
- **LZStats:**
  - MIN_POI = 0.1
  - MAX_POI = 12
  - Ntoys = 1000
  - N_POI_Scan = 12
  - LogScan
  - Calculator Type = 9
  - Two-sided test

Constraint

No constraint
NDWM



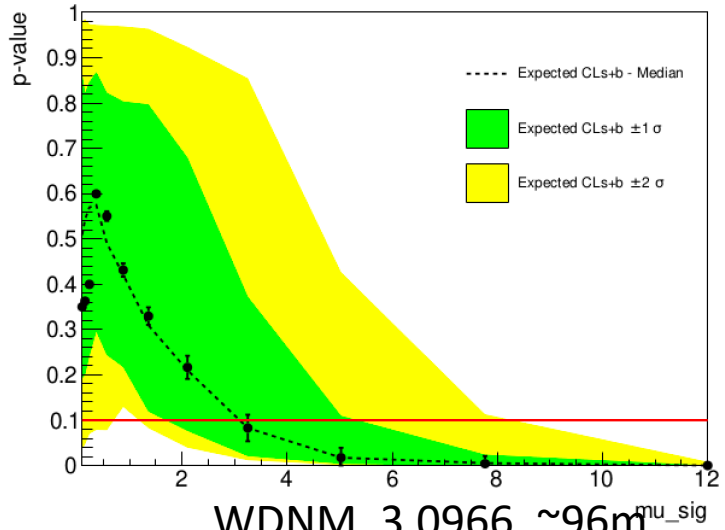
No constraint
WDWM



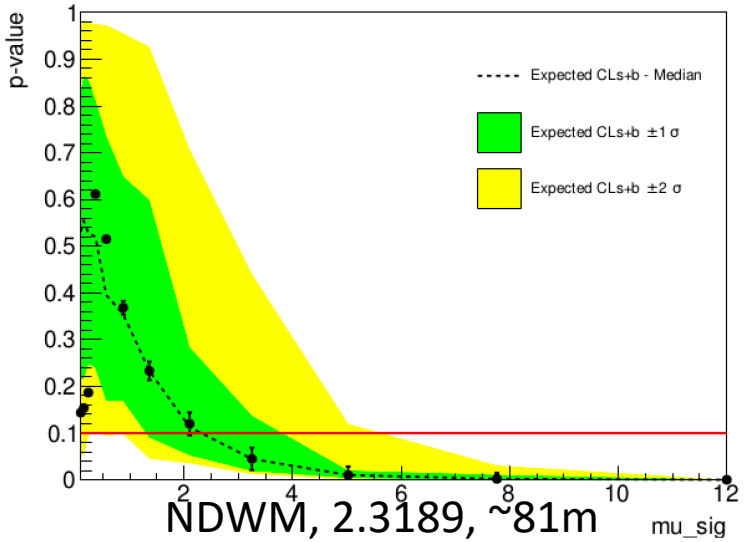
With constraint
NDWM

No constraint
WDWM

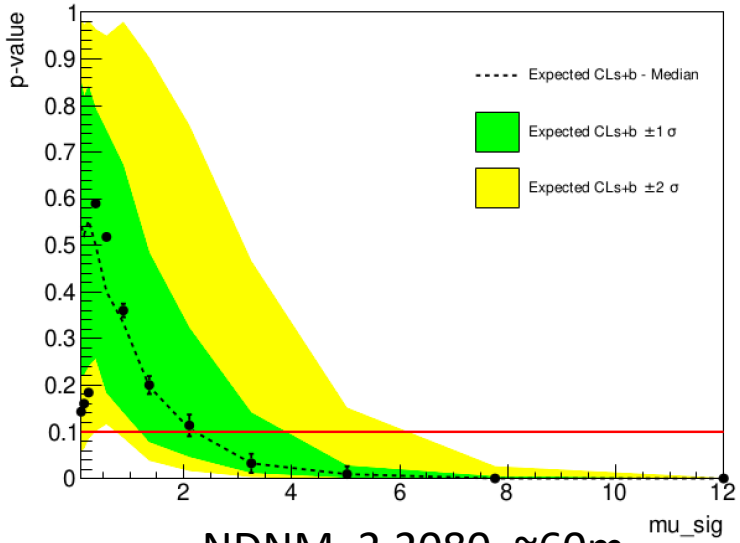
Graph



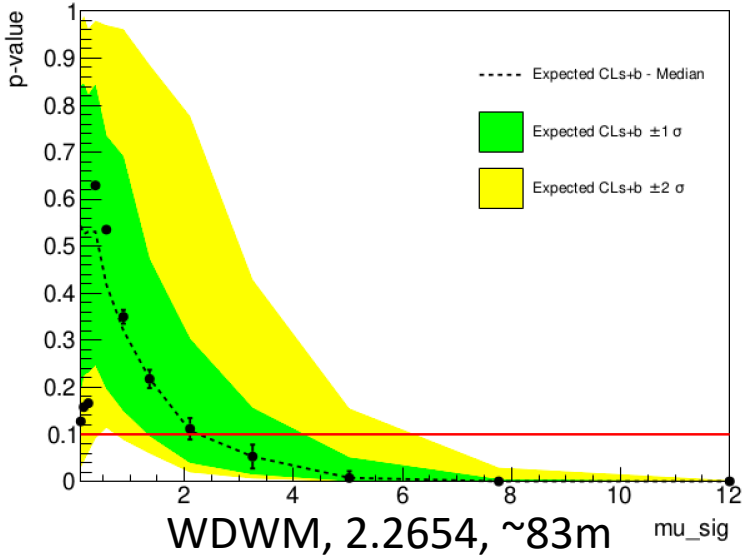
WDNM, 3.0966, ~96m



NDWM, 2.3189, ~81m



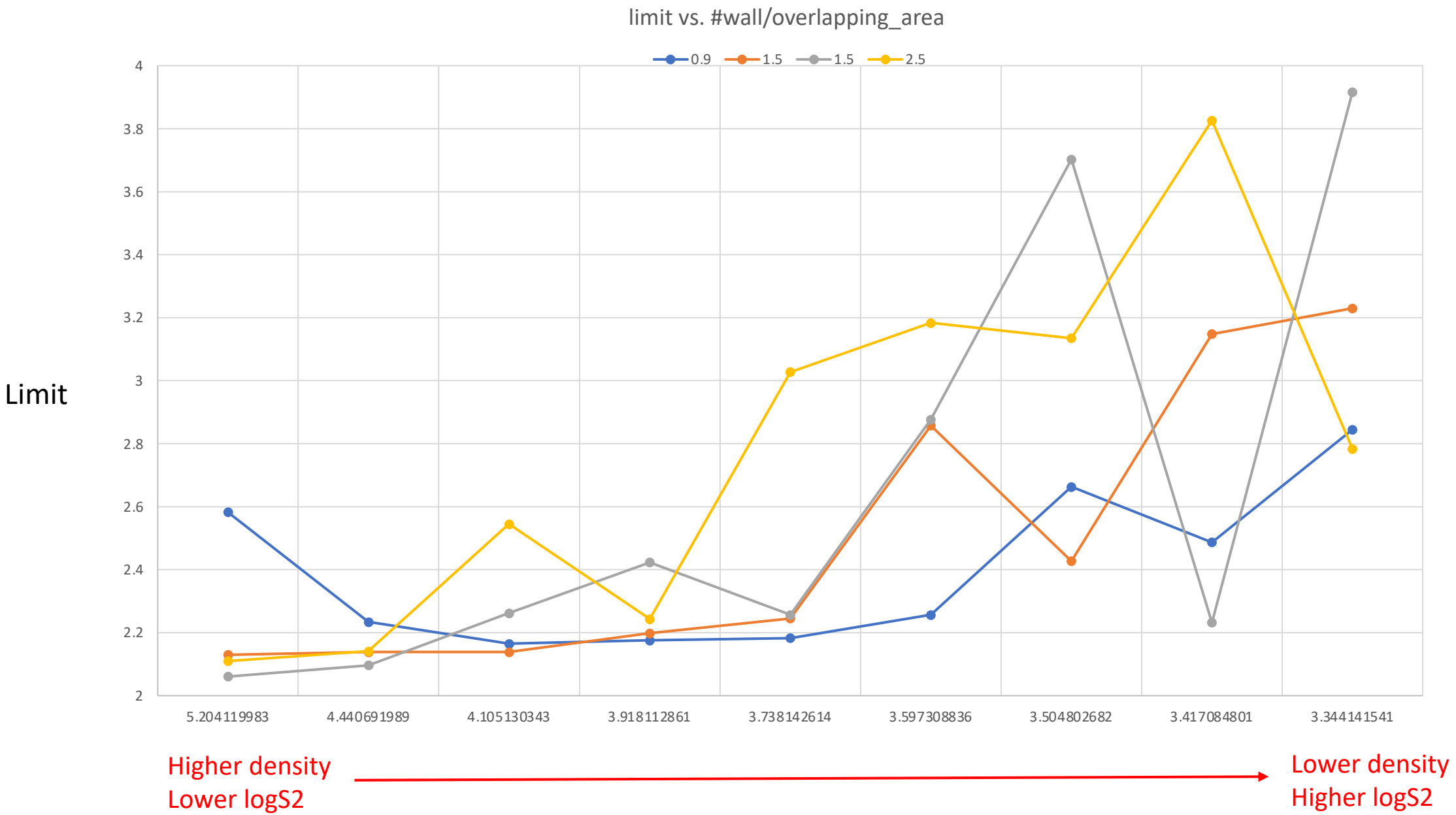
NDNM, 2.2080, ~60m



WDWM, 2.2654, ~83m

WDNM → significantly worse limit
NDWM → slightly worse limit (not sure, fluctuation, low toy number…)
Longer run time for WDNM case → trying other fitting methods

Similar limit


limit vs. #wall/overlapping_area
0.9
1.5
1.5
2.5
4
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
5.204119983
4.440691989
4.105130343
3.918112861
3.738142614
3.597308836
3.504802682
3.417084801
3.344141541
Limit
Higher density
Lower logS2
Lower density
Higher logS2
#WE/Area

To do

- Run no wall constraint cases, compare wall model