1 Manual

The code needs Python 2.7.3 or higher with the standard packages numpy, scipy and the non-standard package ephem, which can be found on http://rhodesmill.org/pyephem/. After all variables have been set, the code may be executed by typing
"python simulation_main.py" into the commandline.

To evaluate a single detector set-up for a given dark matter mass and cross-section range, one needs to set SENSITIVTY_SCAN = 0. If one wants to investigate the scaling behaviour of a dark matter detector for a given (set of) dark matter mass(es) SENSITIVITY_SCAN has to be set to 1 instead (recommended). The variable can be found in simulation_main.py.

The output filename can be specified in the main file via the parameter 'filename_dm' and the output will be stored in the output directory. The outputfile contains the columns detectormass in grams, expected neutrino background rate, the dark matter mass in GeV, the dark matter cross section in cm^2, the expected number of dark matter events, and the α values of the non-directional and directional detector, in total 7 columns. α is the overlap of the distributions of the test statistics for background only and background plus signal hypothesis as discussed in the paper. If $\alpha = 0.00135$ (0.1) we look for a distinction between both hypotheses at 95% (90%) confidence level.

When installing the ephem package, there may be problems with the libastro library. In case it is locally installed, add its path to the path variable via
sys.path.append('/home/*installationFolder*/pyephem-3.7.5.3/libastro-3.7.5'). Try running the code first with commenting this line out. It is found in simulation_main.py after the import statements.

1.1 Physics variables

We need to set the following variables:

variable	where to find	example
detector material (A,Z,N)	modules/constants.py	CF4
M_det0 (detector mass)	modules/constants.py	10^6g
E_thr (energy threshold)	modules/constants.py	5 keV
upper_threshold	modules/constants.py	100 keV
energy_res (energy resolution)	modules/constants.py	0.1
angle_res (angular resolution)	modules/constants.py	0.52
eff_name (energy efficency)	modules/constants.py	'efficiencies/eff_cf4_5keV.dat'
energy efficency datafile	efficencies/	eff_cf4_5keV.dat
nu_sigma (flux uncertainties)	simulation_main.py	np.array([0.08,0.2,0.2])
m_DM_array (DM mass)	simulation_main.py	np.array([6., 40., 1000.])
sigma (DM cross-section)	simulation_main.py	np.logspace(-40,-52, 120)

1.2 Scanning Variables

Here, we can set the following variables:

variable	description	recommended
N_tt	# of 2d-DM-pdf's to capture time-evolution (equally spaced)	10
factor	# of loops for a mass/cross-section pair (avoids memory issues)	10
N_min_nu	# of neutrino events to create 2-d-pdf (per factor)	150000
N_min_dm	# of DM events to create 2-d-pdf (per factor)	150000
source_length	maximal # of neutrino/DM events in event pool (per factor)	5000
N_sim	# of simulated pseudo-experiments (per factor)	1000
N_bins	# bins in histogram of the Q-distributions	50
steps	# steps to integrate the Q-distribution	4000
N_erec	# energy bins in the 2-d-pdfs	36
N_theta	# angular bins in the 2-d-pdfs	16
basenamepdf	defines the name of the 2-d-pdfs, should include detector material	-

Table 1: All parameters above the horizontal line are in simulation_main.py, the last three parameters can be found in constants.py

The creation of the 2-d-pdf's and the calculation of the test statistic Q runs in loops in order to keep the memory demand under control. There are in total 'factor' loops that each create N_min_dm and N_min_nu events for the 2-d-pdfs and calculate the Q-value of N_sim pseudo-experiments, that are stored for the final histogram of Q. For small background rates (smaller than 100) one loop is in principle sufficient. The total number of pseudo-experiments that enter the distribution of the test-statistic is N_sim $\times$ factor.

The 2-d-pdf's should not contain less than 1.5×10^6 events in order to be smooth. This means that N_min_dm $\times$ factor should be greater than that. (the pdf's are created before evaluating Q)

The code creates a new neutrino and dark matter event pool for each dark matter mass/ cross-section pair in each of the factor-loops. This helps to ensure that the pseudo-experiments are independent. For each pseudo-experiment the code then randomly draws events out of this event pool. I assume it would speed up the code if only one set is generated for one dark matter mass. That should work for small event numbers, but if the background rate becomes large (let's say 10^5 events) it is difficult to simulate and store enough events. To get a good statistic for the Q-distribution many pseudo experiments have to be generated (like N_sim = 10^4) and one quickly runs into problems because the number of needed events becomes very big.

For large background rates source_length should be set to a value around the order of background rate $\times$ N_sim to ensure that each simulated event is used only once in a single pseudo-experiment. To avoid memory problems the factor parameter should be increased significantly (For each loop, there will approximately be 'background rate $\times$ N_sim' events. This product should not exceed 10^6 by a lot). If we want to simulate, e.g. 10000 pseudo-

experiments with an expected background of 10^6, we could set factor to 10000 and N_sim to 1 and increase source_length ideally to 10^6. These are background rates for which the code starts to get problems.

The 2-d-pdf's have to be created for each detector material and energy threshold seperately and should carry a unique name defined in basenamepdf,
(E.g. 'pdfs/2dpdf_'+str(Z)+'_'+str(E_thr)+'keV'). They only have to be generated once and the code checks whether a pdf for the target material and energy threshold exists in the folder 'pdfs/'.

1.3 Known Issues

If the detector exposure is too small or the energy threshold too large such that no background event is expected, the code will get stuck in an infinite loop. So, one has to make sure that the exposure is large enough and at least one background event is expected. I couldn't be bothered fixing it, because we want to have background.

There is an old parameter in the code called N_Q which should be set to 1 and is redundant.

Also, in the constants.py module there are some variables in a section 'options' towards the end of the module. I have not tested whether they give reliable results for what they say they should do. So, they should not be changed.