Fake trigger background simulation – status

- Blahoslav Pastirčák, Pavol Bobík,
- Franceso Fenu, Kenji Shinozaki

10th JEM-EUSO Intern. meeting, RIKEN Tokyo Japan,
7th December 2011

Outline

- Motivation
- SW and configuration
- Rates
- Stored information
- Issues since last simulation meeting

Motivation

- Very high statistics of bckg needed to be sure when filtering events → 10^5 events
- It corresponds to 10^14 GTU's
- Not possible to be simulated by ESAF which is 10^3 slower then presented code and cannot be runned by parallel computing (memory share)
- Fast and standalone code written in C++ developed by Francseco Fenu

The code

- Trigger algorithm implemented (the same as in ESAF)
- One PDM simulated
- PTT alg. → 2nd level → 1Hz/PDM
- LTT alg. → 3rd level → 1 mHZ/PDM
- Bckg source → Poisson ditribution of avg 500 photons/(m^2 s sr)
- Code fast but since to produce huge statistics has to be run in parallel (on Kosice cluster)

JEM-EUSO Kosice cluster

- Actually used and available for also for collaboration 7*32 cores @ 2.3 Ghz; 25 TB
- Fedora Core 14 1.2.5-2.fc14
- kernel 2.6.35.13-91.fc14.x86_64
- Gcc 4.5.1-4
- Disk space shared by nfs
- ROOT v30.00, ESAF trunk, GEANT4 9.4

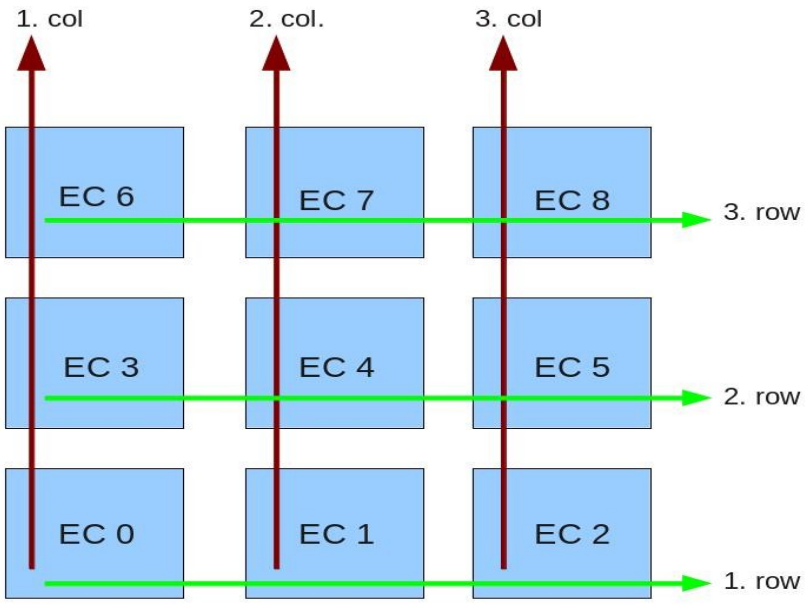
Configuration

M36

BG = 2.1 ph/pix/GTU

PTT_integr = 43

LTT integration = 145

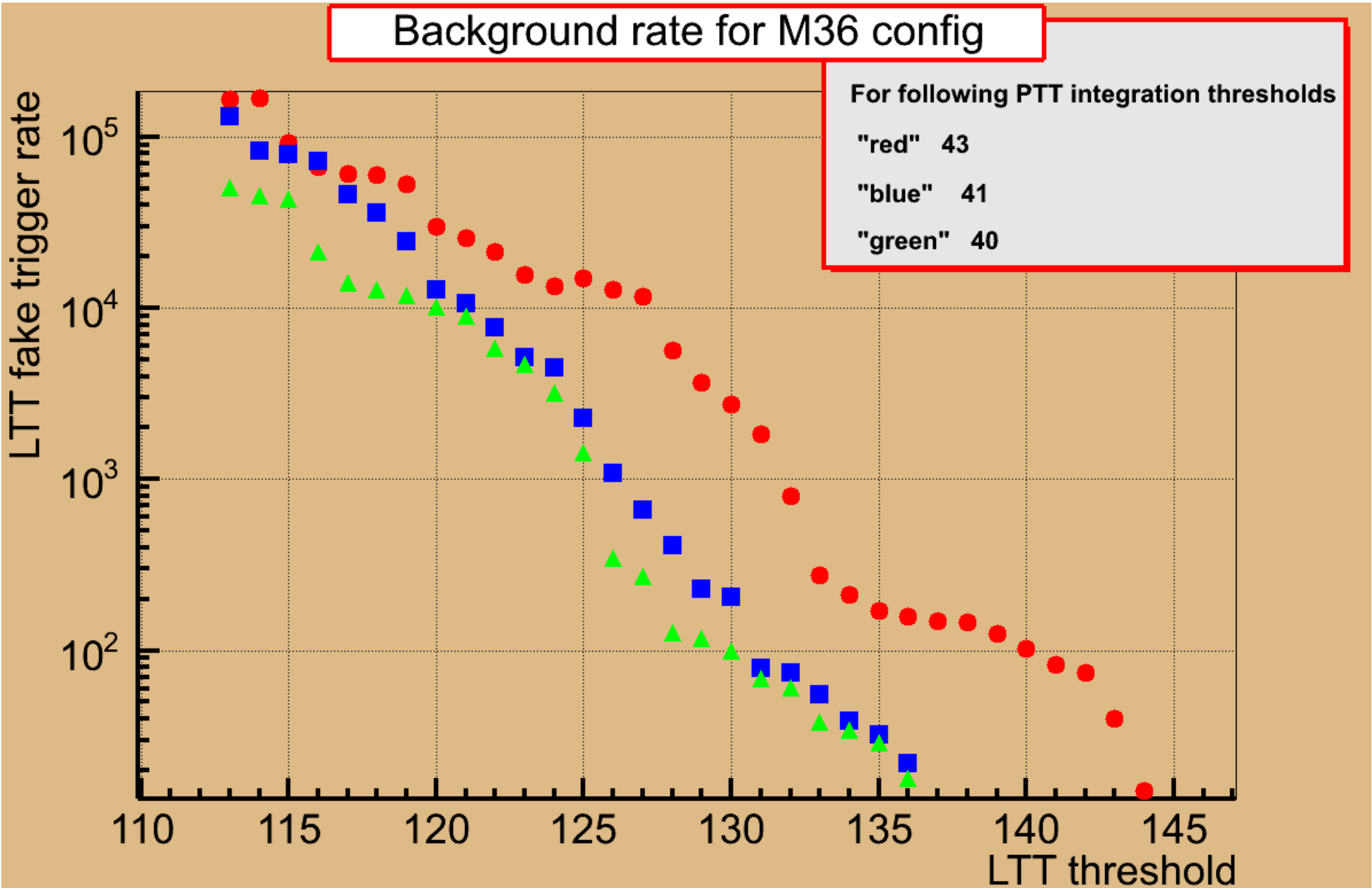


1 PDM = 9 EC = 1296 pixel

1 EC = 4 x PMT = 144 pixel

1 PMT = M 36 = 36 pixel (6 x 6)

Background rate for M36 configuration



Stored information

- Two files written when thresholds reached:
- PTT_SECOND_OUT → (x,y,pers, ecid,counts)

  12x12x5 =720 lines/pdm
- LTT_SECOND_OUT → (x,y,time,counts)

  36x36x31 = 40196 lines/pdm
- Information for which PPT, LTT dumped (reached threshold)
- Analysis of only pixels contributed to LTT going on

Older result

The information of accumulated LTT triggers stored to ascii file in the 4 column format:

row in EC (0-35) : column in EC (0-35) : time (0-30) : counts/pixel

1 LTT trigger = 36x36x31 lines

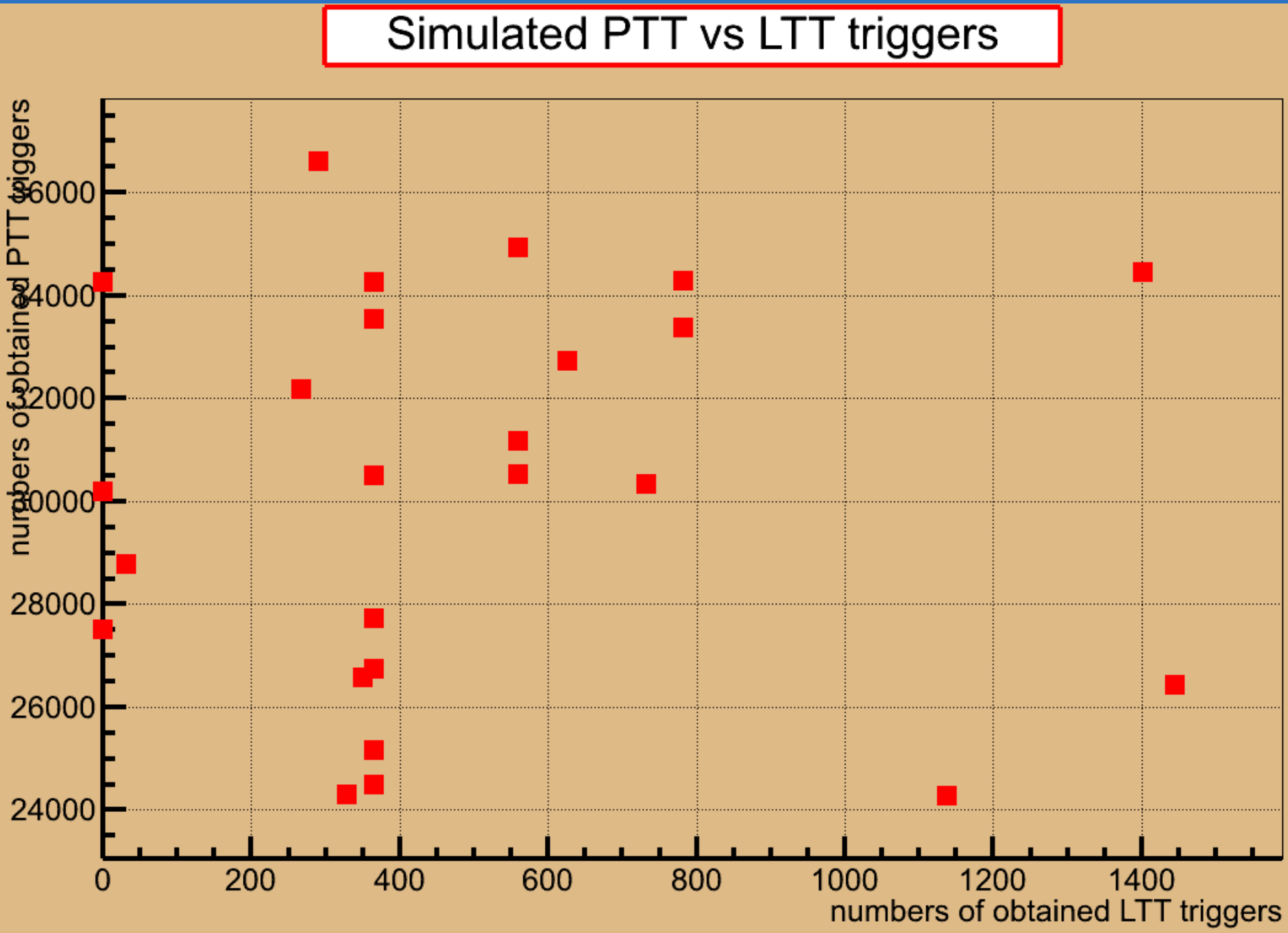
Average size of the LTT output : 250 MB/ 1.e9 Gtu's

We reprocessed it to store like root ntuples: 10 MB/1.e9 GTU's

	PTT (MB)	LTT (MB)	LTT lines	LTT triggers	ltt.ascii (MB)	ltt.root (MB)
Run.8	290	539	42988320	1070	828	48
Run.14	232	316	25150176	626	481	30
Run.25	322	135	10767168	268	204	13
Run.36	304	369	29408832	732	563	34
Run.37	246	185	14704416	366	280	18
Run.51	385	1417	107993088	2688	2246	116
Run.93	347	1281	97627680	2430	2026	105
Run.94	288	1281	97627680	2430	2026	105
Run.95	263	816	65085120	1620	1336	71
Run.100	285	898	71673984	1784	1476	78
Run.109	328	8554	651895776	16226	14000	682
Run.110	307	539	42988320	1070	828	48
Run.111	316	539	42988320	1070	828	48
Run.112	308	881	70308000	1750	1447	77
Run.114	232	539	42988320	1070	828	48
Run.116	298	172	13659840	340	260	16

Present statistics

- New 10^12 GTU's simulated during cca 6 weeks on PC cluster (power consumption and not clarified configuration limited higher st.)
- 12 000 LTT triggers → 1mHz/PDM
- 750000 PTT triggers → 0.1 Hz/PDM


Simulated PTT vs LTT triggers
numbers of obtained PTT triggers
36000
34000
32000
30000
28000
26000
24000
0
200
400
600
800
1000
1200
1400
numbers of obtained LTT triggers

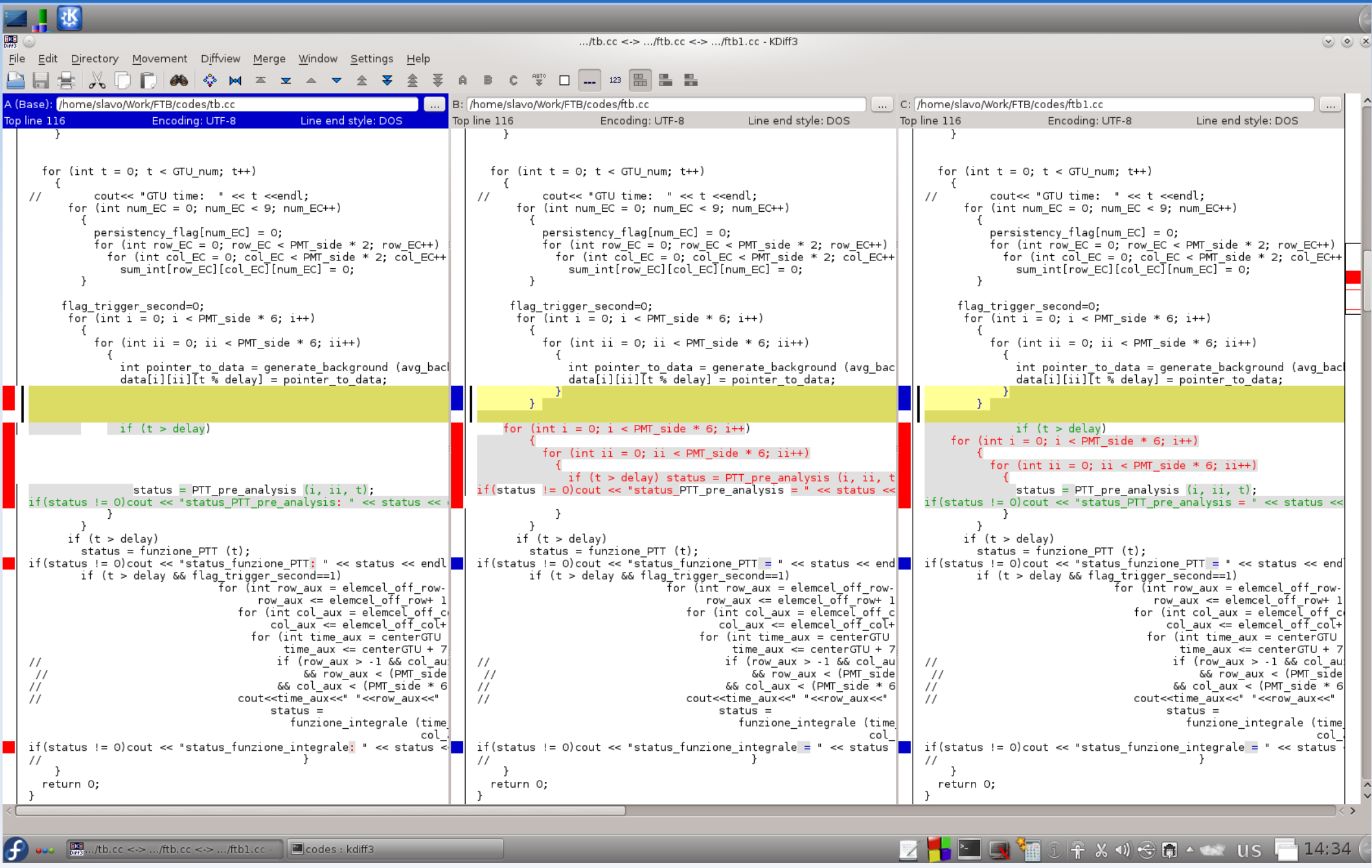
Issues since last simulation meeting

- Checked of the influence of observed bug on obtained data - done
- Random number generation check - done
- Preparation the code for M64 configuration – partially done
- Pattern recognition of the obtained result – under study (contact with Svetlana, possibly Thomas)

Possible problem in the code

- The trigger code exctrated from ESAF for determining appropriate threshold values affected: → generate_background and PPT_pre_analysis status executed in the same loop, but has to be accomplished separately
- Results with and without modification of the code were compared at the same initial conditions
- no influence on obtained LTT rates up to 10^9 GTU's

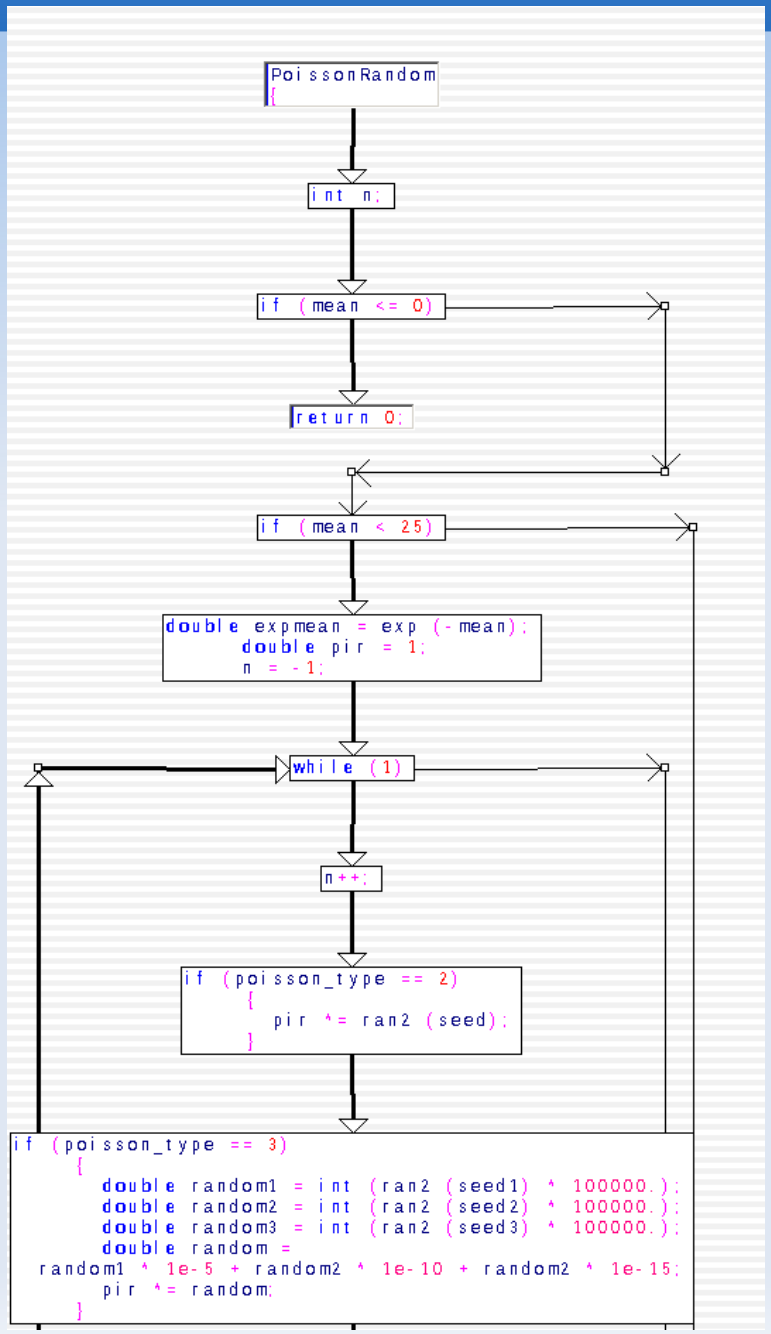
Possible problem in the code



Random number generation

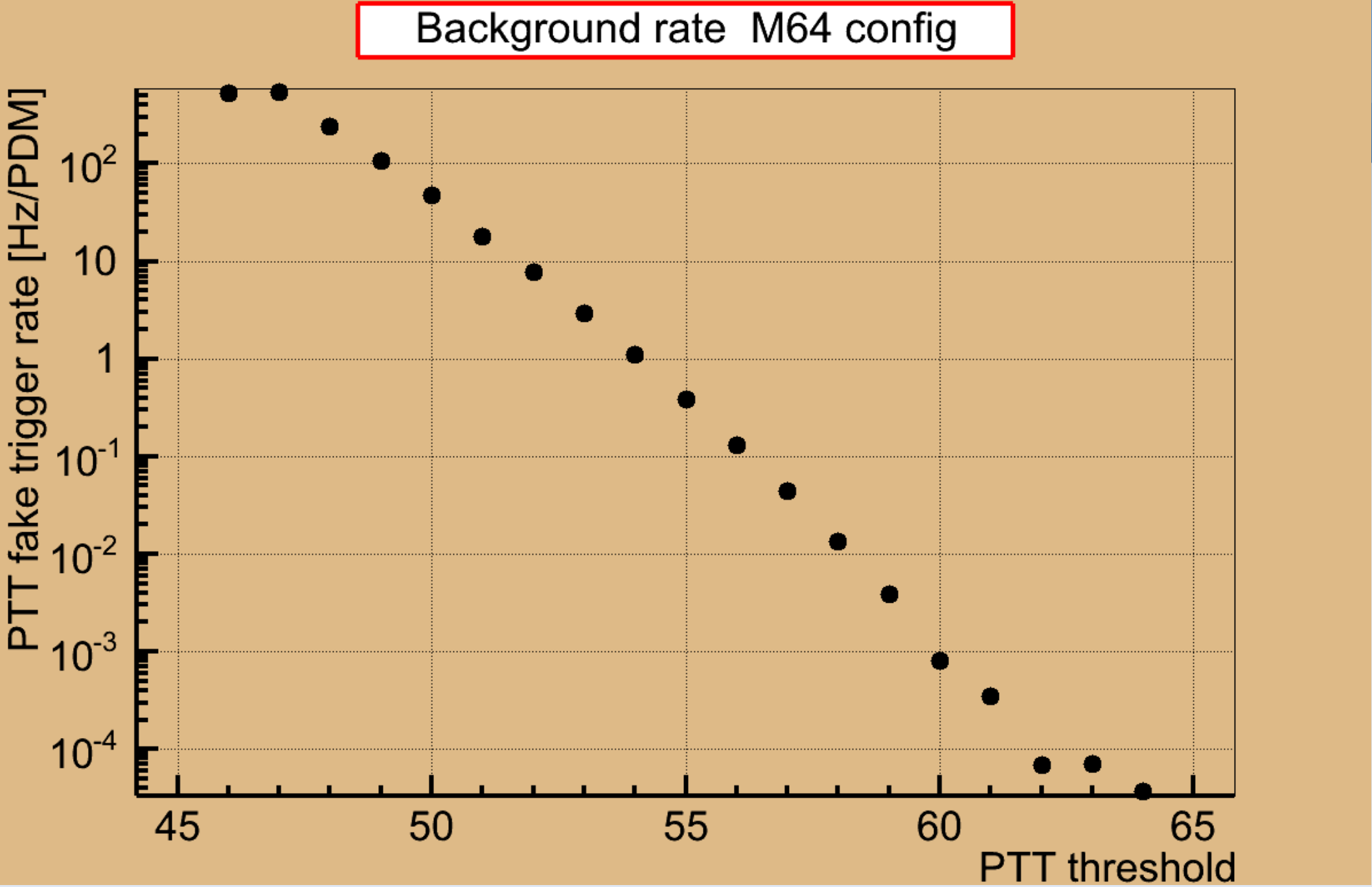
- Due to very high generated statistics possible problems with random number generation
- Previously 4 different types of rnd generation acording to predefined Poisson distributions investigated from ROOT investigated
- Folowing schema used for improving
- Check the random seeds -> succesfully follow Poisson

Random number generation

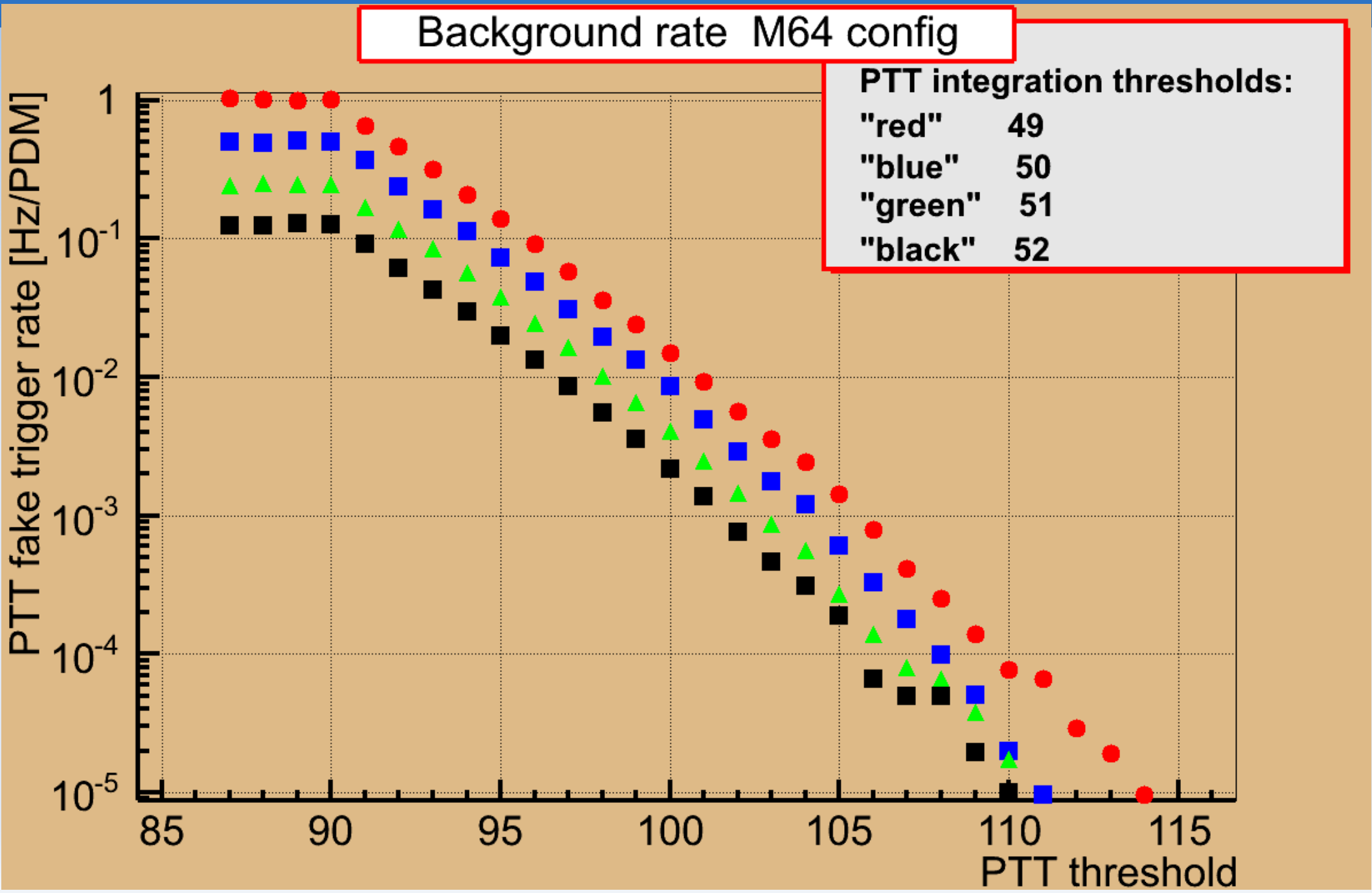


Preparation fo M64 config simulations

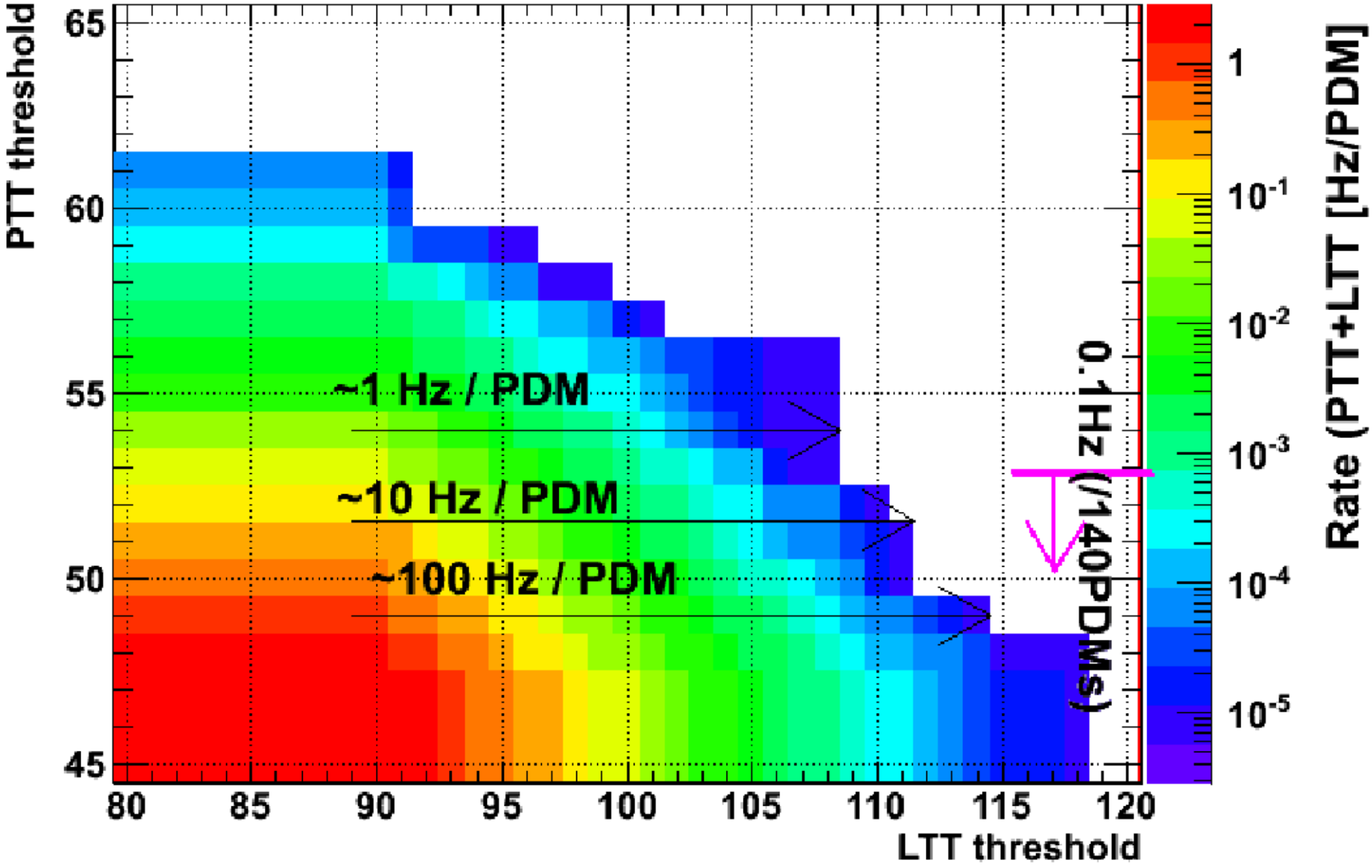
- Modification of BG – scaled according (36/64)^2
- Modify PTT and LTT integration thresholds
- Following background rates for M64 obtained and compared with the result provided by Kenji


Background rate M64 config
PTT fake trigger rate [Hz/PDM]
10^{2}
10
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
45
50
55
60
65
PTT threshold

M64 config


Background rate M64 config
PTT integration thresholds:
"red" 49
"blue" 50
"green" 51
"black" 52
PTT fake trigger rate [Hz/PDM]
1
10-1
10-2
10-3
10-4
10-5
85
90
95
100
105
110
115
PTT threshold

M64 config


PTT threshold
65
60
55
50
45
80
85
90
95
100
105
110
115
120
LTT threshold
~1 Hz / PDM
~10 Hz / PDM
~100 Hz / PDM
0.1Hz (/140PDMs)
Rate (PTT+LTT [Hz/PDM]
1
10-1
10-2
10-3
10-4
10-5

Summary, todo

- Checked trigger rates obtained from the code compatible with expectation
- Visualisation at present level don't show structures
- Negligible influence of the observed bug in the code to the obtained results
- Improved random number generation OK
- The code modified for M64 configuration and prepared to start massive simulation
- Pattern recognition of the obtained data under study