CLAS-NOTE 2005-14
CVS: docs/clas_notes/g11_cooking
Version: 1.3

g11 data processing

M. Ungaro, R. De Vita, L. Elouadrhiri

1 Introduction

This document summarizes the work done in processing the g11 data for calibration and analysis purposes. In order to achieve optimal calibration, several "pass 0" were performed. For "pass 1" there were 16 output files for each input file, between processed data, skims, ntuple, root files, monitoring histograms, trip files. The mysql database was used to monitor the quality of the reconstruction both during the pass 0 to check and improve the detector calibration and during pass 1 to identify the golden runs for the physics analysis.

2 The g11 data

The data was taken between May the 17th and July the 29th 2004. A total of 20 billion triggers were recorded in 421 production runs for a total of 10,500 bos files. The silo path of the files location is:

`/mss/clas/g11a/data`

Figure 1 shows the integrated luminosity accumulated during data acquisition.

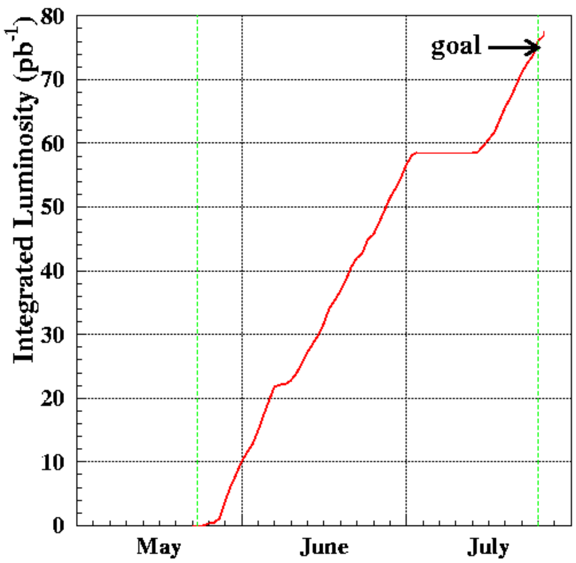

Figure 1: Integrated Luminosity for g11

The run range is $43490 - 44133$. The beam energy was 4018.6 MeV up to run number 44107, then it was set to 5021.14 MeV.[1] Since each file is about 2 GB in size, g11 accumulated a total of about 21 TB of data.

3 Software

The package chosen for the data processing was *recsis*. The first pass 0 was performed using the release 4-9, while the final pass 1 used sofwtare based on release 4-11. In between several changes were made to implement the new start counter in the reconstruction, add new variables for online monitoring, and improve the tracking efficiency. The final software packages use for the data processing are stored in.

`/home/clasg11/top_dir`

This differs from the standard release-4-11 in the:

[1] These numbers represent the nominal energy. The actual energy delivered in CLAS might differ of up to few MeV.

- ana, c_sql,and bankdefs packages modified to include the new monitoring variables,
- st and include packages for the implementation of the new start counter calibration constants,
- seb and pid packages for the optimization of the photon selectina algorithms,
- recsis and user_ana packages for the definition of the output banks and skimmed bos files.

3.1 The new Start Counter

The new Start Counter, shown in Figure 2 differs from the original one for the higher azimuthal segmentation. The new device consits of 24 scintillator paddles (4 per sector) surrounding the the target. The higher segmentation results in a batter time resolution for high multeplicity events.

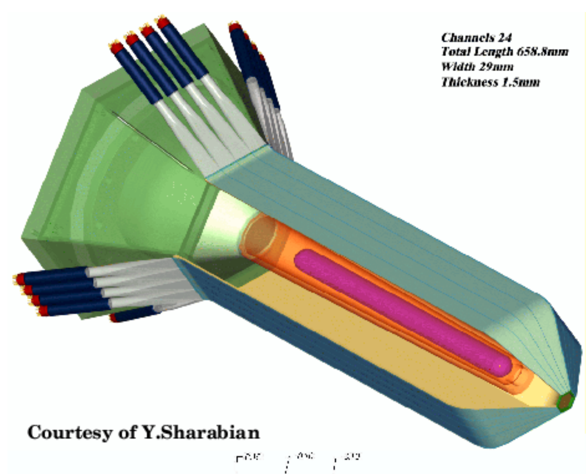

Figure 2: The new Start Counter

The original st package was modified to implement the new geometry. Backward compatibility was mantained using the variable beam_type of the Run_control map to toggle between the old (beam_type=1) and new (beam_type=4) configuration[2]. The raw ST infromation (ADC and TDC values) is read from the bos banks STN0 and STN1(stread_from_bos.F) and converted into energy and time (stn_calibrate.F) based on the conversion constants stored in the STN_CALIB system of the calibration databsed. Only the hits within a time window set by the calibration constants

[2]See the file include/utilities.h for mode detailed information on the definition of this variable.

TDC_minand TDC_max (system ST_CALIB) are retained for reconstruction to reduce accidentals between the ST and the Tagger. The reconstructed hits (ST1 bank) are matched to the tracks in the same sector. This operation is performed first at the hit Based and then at the Time base level. For each track the closer ST hit in the transverse direction is identified and its time corrected for the hit position along the paddle (stn_find)rek.F). The hit time is further correct for time-walk effects (time_walk_stn.F)and finally stored in the STR bank.

3.2 New mysql variables

The Mysql database was used to store information related to the detector calibration and event reconstruction for monitoring purposes. Among the monitored quantities:

- Number of various reconstructed particles and tracks
- Timing: Start Time - RF Time, Vertex Time - RF time for various particles. EC timing.
- Transverse event vertex position from MVRT bank.
- DC Residuals, per sector, per superlayer.
- Missing Masses

See App. A for the complete list of variables used and their description. Such information are collected by the ana package directly (see the routines fill_seb_ntn.F and fill_part_hist.F) and further processed in the user_ana package (see the routine user_erun.F). The collected information is the fed to the database using the structure decsribe by the ddl file calb.ddl and csql.ddl in the bankdefs package. The c_sql package was modified to allow the storage of larger quantity of information.

3.3 Requirements for hit base tracking

As default, the hit base tracking software requires 4 superlayers in DC Region 1 in order to define a track. This requirement has been lowered to 2 for g11.

3.4 Photon flux normalization

During the $g11$ run period, the photon flux was monitored using the Primex pair spectromter and photon normalization runs were routinely performed. Such runs were analyzed to determine and pair spetcrometer and tagging efficiency. The photon flux for production run was then determined running gflux during the data processing. The photon flux information, provided by gflux as text files (see Section 5), was stored in the monitoring histograms and ntuples, and saved in the Mysql database (see Appendix A).

4 Environment

The software and the scripts utilize a number of environment variables to select run index table, input and output data locations, include files, libraries, etc. The most important are:

```
CLAS_CALDB_RUNINDEX   calib_user.RunIndexg11a              run index table

FARM_OUT              /work/clas/farm_output/g11a/pass1     farm output path - failed output
FARM_OUT2             /work/clas/disk8/g11a/pass1           farm output path - reduced ntuple
FARM_OUT3             /work/clas/disk9/g11a/pass1           farm output path - failed skims
PROD_OUT              /work/clas/production2/g11a/pass1     monitoring histos
SILO_OUT              /mss/clas/g11a/production/pass1/v1    silo path of output

SILO_IN               /mss/clas/g11a/data                   silo path of input

TOP_DIR               /home/clasg11/top_dir                 TOP_DIR
OSNAME                LinuxRHEL3

CLAS_PARMS            /home/clasg11/CLAS_PARMS              not the usual parms area
CSQL_USER             offline_g11                           database user name
CSQL_DB               g11_offline                           database name
CSQL_TABLE            pass1                                 final result tables name

CLAS_BUILD            $TOP_DIR                              The software is based on PROD-4-9
CLAS_LIB              $CLAS_BUILD/lib/$OS_CLAS              but a modified copy is in $TOP_DIR

HBOOK_SIZE1           11308                                 Sizes of monitoring histos output.
HBOOK_SIZE2           11300                                 If the hbook file size differs from these,
                                                            something went wrong.

DATA_MIN_SIZE         2500000                               Minimum size of data, ntuple and reduced ntuple.
NTP_MIN_SIZE          700000                                If the files sizes are smaller, something
NTPR_MIN_SIZE         100000                                went wrong or the file is the last of the run.
```

The first five variables define the output locations. The data is spread among few CLAS work disks for space convenience.

The code was developed during the data acquisition, and it was necessary to keep a local version for debugging purpouses and to avoid conflicts due to the new start counter configuration. For this reason, the `TOP_DIR` points to `/home/clasg11/top_dir`, as is the software directory tree `CLAS_BUILD`.

The variable referring to size `HBOOK_SIZE1, DATA_MIN_SIZE, NTP_MIN_SIZE` etc are explained in 5.1.

See App. C for the complete list of variables used and their description.

5 Skims and Output Files

The input file name has the usual format `clas_######.A##` where the first number represents the run number and the second number its extention. For g11, a typical run is made of 28 files. The output was organized as follows:

- The bos output with the reconstructed events was typically 3 GB in size, thus it was splitted in two files.
- The program `pdu` was used on every file to monitor the drift chambers occupancy, producing a hbook output.
- A log file.

- A histogram file monitoring physics quantities of interest.
- One skim (bos output) containing one positive and one negative track.
- One skim (bos output) containing one Kaon (a combined OR of the particle ID schemes: PID, SEB, GPID) and one charged track.
- Three PAW ntuples were produced: one "full" ntuple, one standard `ntuple21`, one reduced ntuple. The latter was also saved in ROOT format by use of the utility `h2root`.
- Five files were produced with a set of programs/scripts storing information about beam trips and photon flux.

The various output nomenclature and description is summarized in Table 1. In the same table is shown the directory tree of the output. The root tree is one of the five environmental variables `FARM_OUT, FARM_OUT2, FARM_OUT3, PROD_OUT, SILO_OUT` described in Sec. 4, so that the full path is *root tree + directory tree*.

Description	Nomenclature	Size	Directory
Input File	`clas_043616.A21`	2 GB	silo (`SILO_OUT`)
Bos Output	`run_43616_pass1.a21.1`, `run_43616_pass1.a21.2`	2 GB ~1 GB	data

log file
pdu
monitoring histogram
physics histograms
	`log_43616_pass1.a21.txt`
	`pdu_43616_pass1.a21.hbook`
	`hst_43616_pass1.a21.hbook`
	`vpk_43616_pass1.a21.hbook`
			log
			pdu
			hist
			skims/vpk

Bos Skims:			
2 positive, 1 negative 1 Kaon, 1 track	`2pos1neg_43616_pass1.a21.bos` `1ckaon_1track_43616_pass1.a21.bos`		skims/2pos1neg skims/1ckaon_1track
PAW Ntuple, Root file			

full ntuple
ntuple 21
reduced ntuple
root reduced ntuple
	`ntp_43616_pass1.a21.hbook`
	`ntp21_43616_pass1.a21.hbook`
	`ntpd_43616_pass1.a21.hbook`
	`rootr_43616_pass1.a21.hroot`
			ntuple
			skims/ntp21
			skims/ntpreduced
			skims/rootreduced

Photon Flux and Trip Files	`gflux43616_tc.a21.dat`
	`gflux43616_eb.a21.dat`
	`gflux43616_erg.a21.dat`
	`gflux43616_erg2.a21.dat`
	`clas_043616.A21.trip`
			gflux
			gflux
			gflux
			gflux
			trip

Table 1: Output files name, size and directory tree starting from the root tree. As an example the failed reduced ntuple are written in: /work/clas/farm_output/g11a/pass1/**skims/ntpreduced**. See Sec. 4.

5.1 Criteria for success/failed reconstruction

The reconstruction could fail for various reasons: `user_ana` could crash, the farm node could die, the trip programs could fail, etc. In case of any of these failures, nothing should be written onto the tapes.

A size check for four output files is performed with the script `verify_conditions` (see Sec. 8). Only if all four conditions are met the files are written on tape. In any other case, the files are written on the CLAS work disks for further investigation. In more details:

- In case of success, the monitoring histogram size can assume only the values `HBOOK_SIZE1, HBOOK_SIZE2`, (see Sec. 4 for the actual values). Any other size is indicative of a crash of `user_ana`.
- The typical output data size is 3 GB. A minimum cut of 2.5 GB is applied (variable `DATA_MIN_SIZE`).
- The typical ntuple size is 1.1 GB. A minimum cut of 0.7 GB is applied (variable `NTP_MIN_SIZE`).
- The production of the reduced ntuple is the last process in the scripts. If this is successful, chances are that everything else was successful. The typical ntuple size is 130 MB. A minimum cut of 100 MB is applied (variable `NTPR_MIN_SIZE`).

The following files are written on the CLAS work disks independently of the above check.

- log file.
- monitoring histograms.
- ntuple21 and reduced ntuple.
- physics monitoring histograms.
- trip files.

6 user_ana configuration

The tcl file is located in `/home/clasg11/cooking/recsis_g11.tcl`. The following tcl variables, with obvious meaning, reflect the g11 run conditions:

```
setc prlink_file_name  "prlink_g11_1920.bos";
setc bfield_file_name  "bgrid_T67to33.fpk";
set torus_current      1920;
set mini_torus_current 0;
set poltarget_current  0;
set TargetPos(3)       -10.;
```

Furthermore, modifications to the tracking code to improve the reconstruction efficiency required the following settings:

```
set trk_maxiter       8;
set trk_minhits(1)    2;
set trk_lrambfit_chi2 50.;
set trk_tbtfit_chi2   70.;
set trk_prfit_chi2    70.;
set trk_statistics    3 ;
set st_tagger_match   15.;
```

The mysql database was activated with the lines:

```
set lmysql      -1;
set nmysql      -1;
```

To store correctly the run extension in the database, the following commands were included in the script `go_COOK` (see Sec. 8)

```
printf "set runfile %d;\n" $RUNEXT >> recsis_g11.tcl
printf "go 100000000;\n"           >> recsis_g11.tcl
printf "exit_pend;\n\n"            >> recsis_g11.tcl
```

to modify the local farm copy of the tcl file accordingly.
The final list of bank kept in the datastream was:

HEAD, TAGR, CL01, HEVT, EVNT, DCPB, ECPB, ECHB, SCPB, CCPB, STPB, TGPB, TBER, TBTR, SCRC, STR, MVRT, PART, TBID, GPID, TDPL, EPIC

It was decided not to keep the raw banks (DC0, etc) for it would be easier to recook a 2 *GB* input file than to retrieve and analyze a much larger output file.

7 The passes

For calibration purposes seven **pass 0** were performed. The first file of each of the 421 runs was processed. Only the bos output and the full ntuple was kept on disk. A typical pass took about two days to run at the JLAB farm computers. The first pass was completed before the end of the data acquisition.

Pass 1 started on *11/05/2004*. By *12/24/2004* 95% was completed, with the remaining 5% failed due to errors / crashes. After several attempts to recover the remaining files, by *1/31/2005* 99% of g11 was completed and the data processing stopped, with around 90 out of 10,500 files left out due to unsolved problems.

The start and end time of each pass is shown in Table 2 .

pass	**start time**	**end time**
pass01	7/01/2004	7/15/2004
pass02	8/05/2004	8/07/2004
pass03	8/15/2004	8/17/2005
pass04	9/02/2004	9/05/2004
pass05	9/20/2004	9/22/2004
pass06	10/05/2004	10/07/2004
pass07	10/18/2004	10/20/2004
pass1	11/05/2004	12/24/2004 (95%) 1/31/2005 (99%)

Table 2: The various passes of g11 data processing. Pass01 took longer because it kept running while the data was being taken.

The complete set of mysql variables was available from *pass02*. A web page was set up to show these variables for each pass, and can be found at:

`http://www.jlab.org/Hall-B/secure/g11/cooking/monitoring/monitor.html`

where the usual username and password are required. A screenshot of the page is shown in Figure 3. The user can select the pass with the buttons at the top of the page, and the variables to display with the mouse.

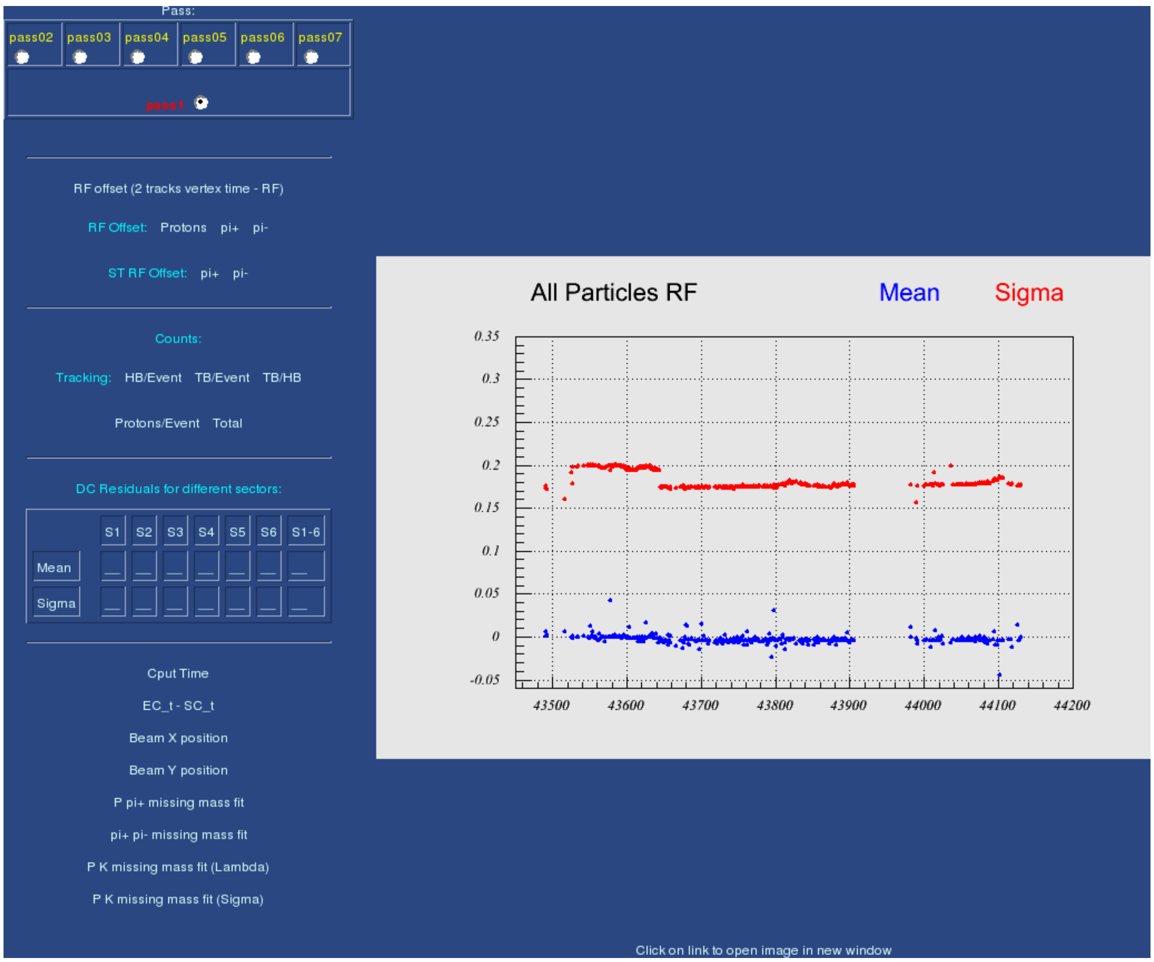

Figure 3: The monitoring page for the g11 online mysql variables. The page can be found at *http://www.jlab.org/Hall-B/secure/g11/cooking/monitoring/monitor.html* Top left: pass selection buttons. Left: variables available.

8 Overview of the scripts used

The script used for g11 data processing are `csh` scripts. They can be grouped as

- I/O to the JLAB farm computers
- data processing script
- integrity check

8.1 Submitting jobs to the farm

The file `/home/clasg11/cooking/lists/list_g11.txt` contains the list of runs to process. The scripts

```
/home/clasg11/cooking/cook_runs
/home/clasg11/cooking/go_run
```

prepare a *jsub* file based on the above list and on the template:

```
JOBNAME: g11a cooking
PROJECT: clas
MAIL:    clasg11@jlab.org
OS:      Linux
QUEUE:   production


OTHER_FILES: /u/home/clasg11/cooking/recsis_g11.tcl
OTHER_FILES: /u/home/clasg11/cooking/user_ana
OTHER_FILES: /u/home/clasg11/cooking/go_COOK
OTHER_FILES: /u/home/clasg11/cooking/verify_conditions
OTHER_FILES: /u/home/clasg11/top_dir/utilities/pdu/pdu
OTHER_FILES: /u/home/clasg11/cooking/norm/norm.pl
OTHER_FILES: /u/home/clasg11/cooking/norm/check_trip_Linux
OTHER_FILES: /u/home/clasg11/cooking/norm/check_trip.sh
OTHER_FILES: /u/home/clasg11/cooking/norm/sync
OTHER_FILES: /u/home/clasg11/cooking/norm/gflux
OTHER_FILES: /u/home/clasg11/cooking/skims/valeri/g11.exe
OTHER_FILES: /u/home/clasg11/cooking/skims/valeri/launch
OTHER_FILES: /u/home/clasg11/cooking/skims/ntpreduced/ntp_filter_Linux
OTHER_FILES: /u/home/clasg11/cooking/skims/ntpreduced/inpfile


COMMAND: go_COOK
```

where all the programs necessary for reconstruction and skims to be copied in the local node are listed. The above scripts are easily customizable in order to process the whole runs or select only particular files (for example the *A00* files).

8.2 The main scripts

The `csh` script `/home/clasg11/cooking/go_COOK` perform the following actions:

- Loads the proper environment variables (see Sec. 4).
- Takes care of the nomenclature of the various outputs (see Sec. 5).
- Executes the *pearl* script `norm.pl` which produces trip and photon flux files.
- Launches the program `pdu` on the data file.
- Launches the program `user_ana` .
- Executes the program `ntp_filter_Linux` which produces the reduced ntuple file.
- Transform the reduced ntuple file the equivalent ROOT version.
- Executes the script `verify_conditions` which determine wether or not the files have the correct size (see Sec. 5.1).
- Copy the outputs to the tape silo or the work disks according to the result of the above check.

8.3 Monitoring the progress.

A script `/home/clasg11/cooking/go_finished` will check the silo for the presence of the various outputs. The output is as follow:

file	number of entries	description
mon_ins	10410	monitoring histos
log_ins	10410	log files
ntp10_ins	10410	ntuple 10
ntp21_ins	10410	full ntuple
ntpr_ins	10848	reduce ntuple
root_ins	10410	root file from reduced ntuple
vpk_ins	10410	physics monitoring histo
out1_ins	10410	bos output
out2_ins	10410	first bos skim
out3_ins	10409	second bos skim
pdu_ins	10375	pdu file
gflux1_ins	10368	photon flux vs. T counter
gflux2_ins	10368	photon flux vs. E counter
gflux3_ins	10368	photon flux vs. Energy (20 MeV bins)
gflux4_ins	10368	photon flux vs. Energy (50 MeV bins)
trip_ins	10410	trip file

The files common to all outputs are then grouped in the text file
`/home/clasg11/cooking/list/finished`.

8.4 Is "failed" really failed? Cleanup.

The last files of a run could fail the criteria for a successful result mentioned in Sec. 5.1. due to the fact that they lack the proper statistic. When the size check failed, all the output are written on the CLAS work disk. A script `Master_check` then perform the following actions:

- Collects information about the files present in the work disks used.
- Discriminates between last files of a run and normal files.
- Performs the normal check for the normal files. If all the 16 outputs are present and the criterias are met, they will be put onto the tape silo. Otherwise they will be deleted from the disks for space convenience.
- Perform the monitoring histogram size check on the last files. If the condition is met, and the 16 outputs are present on the disks, they will be put onto the tape silo. Otherwise they will be deleted from the disks for space convenience.
- Deletes all the files in the work disks that are also listed in
  `/home/clasg11/cooking/list/finished`.

9 The mysql monitoring scripts

Every time a new pass is done, the enviroment variable `CSQL_TABLE` should be changed accordingly. For g11, it assumed the values:

`pass0, pass02, pass03, pass04, pass05, pass06, pass07, pass1`.

These names refers to the mysql tables created under the database `g11_offline`. In the directory `/home/clasg11/cooking/monitor` the script `make_mysql_mons` downloads the variables from the mysql database and produces data files. The script will look in the subdirectory `mysql` for the customizable functions. As of the writing of this document, the table from which the script reads the variable is hardcoded in the functions, so the user has to change them every time there is a new pass. For example the function in the file `EC.sql` that reads the EC timing info from *pass02*:

```
use g11_offline ;
select  time, runno, meanECm, sigmaECt from pass02 order by - time;
```

has to be changed like follows to read from *pass03*:

```
use g11_offline ;
select  time, runno, meanECm, sigmaECt from pass03 order by - time;
```

After the data files have been produced, the ROOT macro `monitor.C` will produce *GIF* files to be displayed on web. The user can move the files in the subdirectory `img` and modify accordingly the template `monitor.html` to obtain the page shown in Figure 3.

A Mysql variables

The following is the list of the variables used in the mysql database. An X indicates "sectors" or "superlayer". For example $ResSLXSX\_ave$ are in fact 36 variables (6 sectors, 6 superlayers).

Field	Type	Description
id	int(11)	
time	timestamp(14)	When the file has been processed
user	char(12)	Username
jobname	char(32)	Jobname
node	char(32)	farm node
calibdb	char(64)	database name
runindex	char(32)	Run index
runno	int(11)	Run number
runext	int(11)	Run extension
NPROC	int(11)	Number of processed events
CPU	float	CPU time used
FC	float	Faraday Cup (for electron runs)
FCG	float	Live Gated Faraday Cup
TG	float	Gated Time
IBEAM	float	Beam Current
NpipSX	int(11)	Number of pi+ per sector
Nhb	int(11)	Number of hit based tracks
Ntb	int(11)	Number of time based tracks
Nprot	int(11)	Number of protons
Npip	int(11)	Number of pi+
Ndeut	int(11)	Number of deuterons
Nphot	int(11)	Number of photons
Nepiphp	int(11)	Number of pi+ for heliciy + events
Npiphn	int(11)	Number of pi+ for heliciy + events
Ngamma	int(11)	Number of photons
Ng_norm	int(11)	Number of incident photons

Field	Type	Description
meanRFgoodtag	float	Start time - RF time : Mean
sigmaRFgoodtag	float	Start time - RF time Sigma
meanRFalltag	float	Start time - RF time : Mean, all photons
sigmaRFalltag	float	Start time - RF time Sigma, all photons
meanRFprot	float	Proton Vertex Time (TOF) - RF Time: Mean
sigmaRFprot	float	Proton Vertex Time (TOF) - RF Time: Sigma
meanRFpip	float	pi+ Vertex Time (TOF) - RF Time: Mean
sigmaRFpip	float	pi+ Vertex Time (TOF) - RF Time: Sigma
meanRFpim	float	pi- Vertex Time (TOF) - RF Time: Mean
sigmaRFpim	float	pi- Vertex Time (TOF) - RF Time: Sigma
meanSTpip	float	pi+ Vertex Time (ST) - RF Time: Mean
sigmaSTpip	float	pi+ Vertex Time (ST) - RF Time: Sigma
meanSTpim	float	pi- Vertex Time (ST) - RF Time: Mean
sigmaSTpim	float	pi- Vertex Time (ST) - RF Time: Sigma
meanECt	float	EC Time - Vertex Time: Mean
sigmaECt	float	EC Time - Vertex Time: Sigma
meanECb	float	EC Beta for neutrals: Means
sigmaECb	float	EC Beta for neutrals: Sigma
meanECm	float	pi0 mass: Mean
sigmaECm	float	pi0 mass: Sigma
xbeam	float	x mean of vertex (from MVRT)
ybeam	float	y mean of vertex (from MVRT)
sig_xbeam	float	x sigma of vertex (from MVRT)
sig_ybeam	float	y sigma of vertex (from MVRT)
mm_p_pip	float	P pi+ missing mass: Mean
smm_p_pip	float	P pi+ missing mass: Sigma
mm_pip_pim	float	pi+ pi- missing mass: Mean
smm_pip_pim	float	pi+ pi- missing mass: Sigma
mm_kp_lambda	float	K+ missing mass (Lambda0): Mass
smm_kp_lambda	float	K+ missing mass (Lambda0): Sigma
mm_kp_sigma	float	K+ missing mass (Sigma0): Mass
smm_kp_sigma	float	K+ missing mass (Sigma0): Sigma
ResSLX_ave	float	Residual Mean per sector
ResSLX_sig	float	Residual Sigma per sector
ResSLXSX_ave	float	Residual Mean per sector, superlayer
ResSLXSX_sig	float	Residual Sigma per sector, superlayer

B tcl file

This file is located at: `/home/clasg11/cooking/recsis_g11.tcl`

```
source /u/group/clas/builds/release-4-9/packages/tcl/recsis_proc.tcl;
#
# define packages
turnoff ALL;
global_section off;
turnon seb trk tof egn user pid;
#
#
inputfile           InputFile;
setc chist_filename histfile;
setc log_file_name  logfile;
#
#
setc outbanknames(1) "HEADTAGRCL01HEVTEVNTDCPBECPBECHBSCPBCCPBSTPBTGPBTBERTBTRSCRCSTR MVRTPARTTBIDGPIDTDPLEPIC";
outputfile outfile1 PROC1 2047;
setc outbanknames(2) "HEADTAGRCL01HEVTEVNTDCPBECPBECHBSCPBCCPBSTPBTGPBTBERTBTRSCRCSTR MVRTPARTTBIDGPIDTDPLEPIC";
outputfile outfile2 PROC2 2047;
setc outbanknames(3) "HEADTAGRCL01HEVTEVNTDCPBECPBECHBSCPBCCPBSTPBTGPBTBERTBTRSCRCSTR MVRTPARTTBIDGPIDTDPLEPIC";
outputfile outfile3 PROC3 2047;
#
#
setc prlink_file_name  "prlink_g11_1920.bos";
setc bfield_file_name  "bgrid_T67to33.fpk";
#
#
set torus_current      1920;
set mini_torus_current 0;
set poltarget_current  0;
set TargetPos(3)       -10.;
#
#
# Franz's tcl variables
set trk_maxiter 8;
set trk_minhits(1) 2;
set trk_lrambfit_chi2 50.;
set trk_tbtfit_chi2 70.;
set trk_prfit_chi2 70.;
set trk_statistics 3 ;
#
#
set st_tagger_match 15.;
#
#
set lst_do          -1;
set ltime_do        -1;
set ltagger_do      -1;
set lseb_nt_do      -1;
set lseb_ntn_do     -1;
set lall_nt_do      -1;
set lscr_nt_do      -1;
set lseb_hist       -1;
set lseb_h_do       -1;
set lmon_hist       -1;
set ltrk_h_do       -1;
set legn_h_do       -1;
set ltof_h_do       -1;
set lfec_hist       -1;
set lfec_h_do       -1;
set ltagger_h_do    -1;
set lpart_nt_do     -1;
set lst_nt_do       -1;
set ltbt_nt_do      -1;
set lmvrt_nt_do     -1;
set lpid_make_trks   0;
set ltbid_nost_do   -1;
set lgpid_do        -1;
set lmysql      -1;
set nmysql      -1;
#
# tell FPACK not to stop if it thinks you are running out of time
fpack "timestop -9999999999"
#
#
# do not send events to event display
set lscat $false;
set ldisplay_all $false;
#set nevt_to_skip  44000;
#
#
setc rec_prompt "CLASCHEF_recsis> ";
```

C Environmental Variable file *set_env*

This file is located at: `/home/clasg11/set_env`

```
#!/bin/csh -f
# enviroment variables for G11

limit coredumpsize 209700

setenv CLAS_CALDB_RUNINDEX calib_user.RunIndexg11a          # run index

setenv WORK         /home/clasg11/cooking                   # all scripts locations
setenv FARM_OUT     /work/clas/farm_output/g11a/pass1       # farm output path
setenv FARM_OUT2    /work/clas/disk8/g11a/pass1             # farm output path - reduced ntuple
setenv FARM_OUT3    /work/clas/disk9/g11a/pass1             # farm output path - failed skims
setenv PROD_OUT     /work/clas/production2/g11a/pass1       # monitoring histos

setenv SILO_OUT     /mss/clas/g11a/production/pass1/v1      # silo path of output
setenv SILO_IN      /mss/clas/g11a/data                     # silo path of input
setenv CHEF_G11     clasg11                                 # user name of chef

setenv ROOTSYS        /u/apps/root/4.00-03-gcc3.2.3/root
setenv ROOTLIB        $ROOTSYS/lib
# setenv CERN         set with "setup" above
setenv CERN_LEVEL     2003
setenv CERN_LIB       $CERN/$CERN_LEVEL/lib
setenv CERN_BIN       $CERN/$CERN_LEVEL/bin
setenv CERN_ROOT      $CERN/$CERN_LEVEL

setenv TOP_DIR        /home/clasg11/top_dir                # TOP_DIR
setenv OSNAME         "LinuxRHEL3"
setenv OS_NAME        $OSNAME
setenv OSCLAS         $OSNAME
setenv OS_CLAS        $OSNAME
setenv CVSROOT        /group/clas/clas_cvs

setenv MYSQL              /apps/mysql
setenv MYSQBIN            $MYSQL/bin
setenv MYSQLIB            $MYSQL/lib/mysql
setenv MYSQINC            $MYSQL/include/mysql
setenv MYSQL_INCLUDE_PATH /apps/mysql/include/mysql
setenv MYSQL_LIB_PATH     $MYSQL/lib/mysql

setenv CLAS_PARMS     /home/clasg11/CLAS_PARMS              # usual parms area. Note: calibration databa
se is used
setenv CSQL_DBHOST  clasdb                                  # database host name
setenv CSQL_USER    offline_g11                             # database user name
setenv CSQL_DB      g11_offline                             # database name
setenv CSQL_TABLE   pass1                                   # final result tables name
setenv CSQL_DDL     $TOP_DIR/bankdefs/csql.ddl              # ddl file name
setenv CSQL_CALIB   $TOP_DIR/bankdefs/calb.ddl              # calibration constants DDL
setenv CSQL_COMM    "g11 test"                              # comment that will appear in the final table.

# for constants,
setenv GROUP          clas
setenv CLAS_LOCATION  /group
setenv CLAS_LEVEL     release-4-9                           # frozen at 4-9
setenv CLAS_ROOT      $CLAS_LOCATION/clas
setenv CLAS_TOOLS     $CLAS_ROOT/tools
setenv BUILDS         $CLAS_ROOT/builds
setenv CLAS_PROD      $CLAS_ROOT/builds/PRODUCTION
setenv CLAS_DEVEL     $CLAS_ROOT/builds/DEVELOPMENT
setenv CLAS_BUILD     $TOP_DIR
setenv CLAS_LIB       $CLAS_BUILD/lib/$OS_CLAS
setenv CLAS_SLIB      $CLAS_BUILD/slib/$OS_CLAS
setenv CLAS_BIN       $CLAS_BUILD/bin/$OS_CLAS
setenv CLAS_PACK      $CLAS_BUILD
setenv CLAS_CMS       $CLAS_PACK/cms
setenv CLAS_SCRIPTS   $CLAS_PACK/scripts
setenv RECSIS         $CLAS_PACK
setenv RECSIS_RUNTIME /group/clas/clsrc/recsis/runtime

setenv HV_LOCATION    $CLAS_PACK/Hv

setenv HBOOK_SIZE1     11308
setenv HBOOK_SIZE2     11300
setenv DATA_MIN_SIZE  2500000
setenv NTP_MIN_SIZE   700000
setenv NTPR_MIN_SIZE  100000


set path = ($ROOTSYS/bin $path  $CERN_BIN $TOP_DIR/bin/$OS_NAME $CLAS_BIN $MYSQBIN  $CLAS_PACK/scripts
           $CLAS_TOOLS/caldb /apps/bin)
setenv LD_LIBRARY_PATH $ROOTSYS/lib:${CLAS_LIB}:/usr/lib:${CERN_LIB}:${MYSQLIB}:/usr/local/lib:/usr/lib:
                       ${TOP_DIR}/lib/${OS_NAME}:${TOP_DIR}/slib/${OS_NAME}
setenv LOCAL_LIB $TOP_DIR/lib/$OSNAME
```

D Monitoring histograms

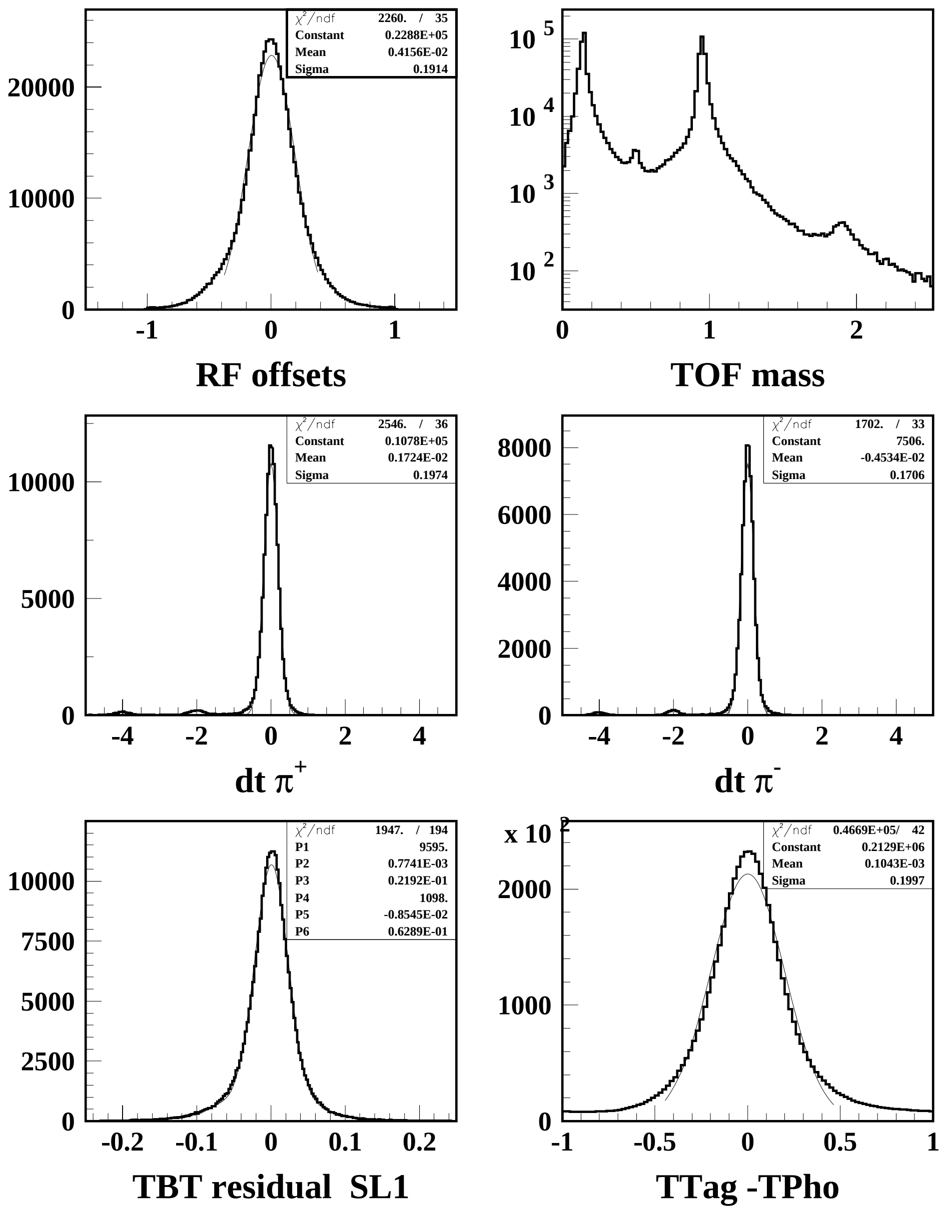

Figure 4: Sample of monitoring hisograms

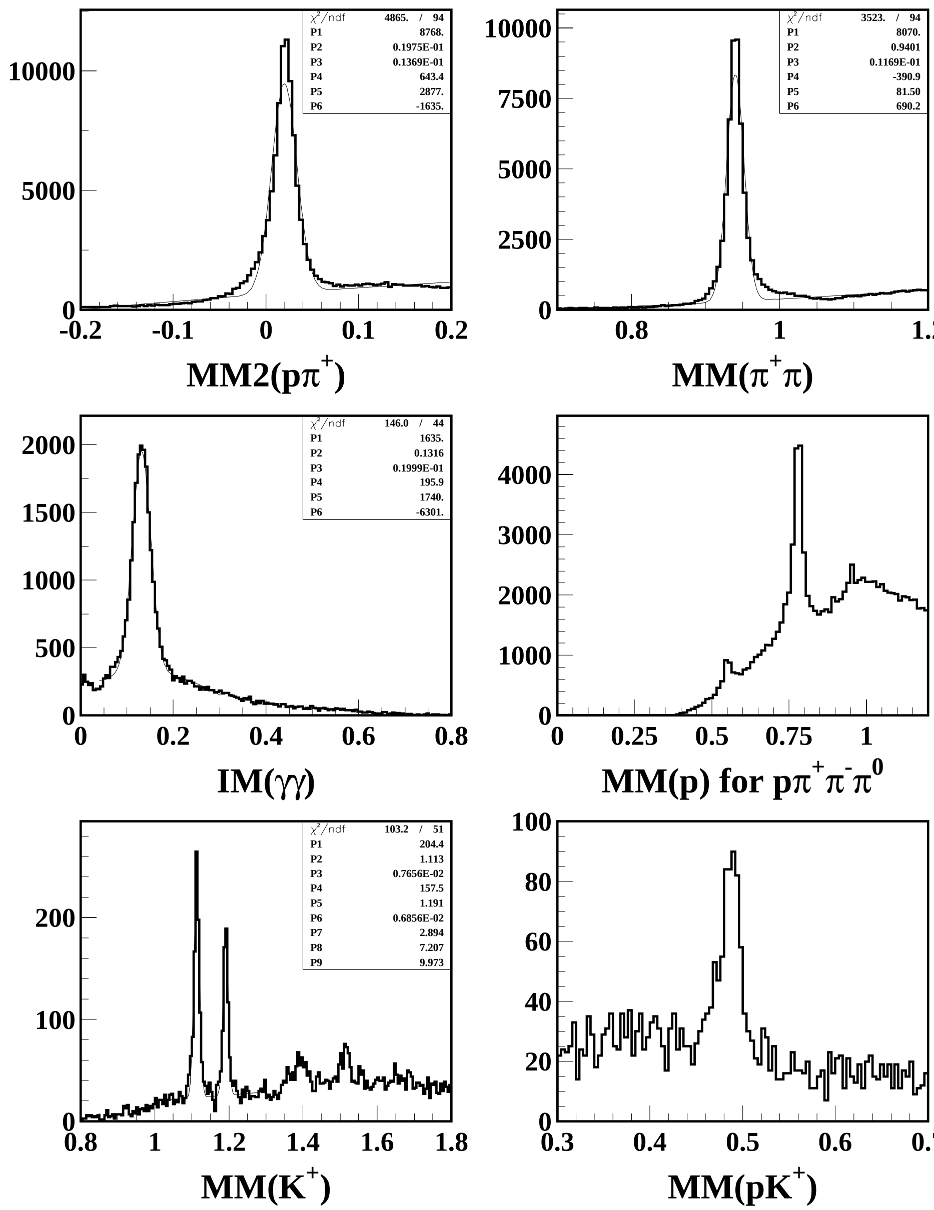

Figure 5: Sample of monitoring hisograms

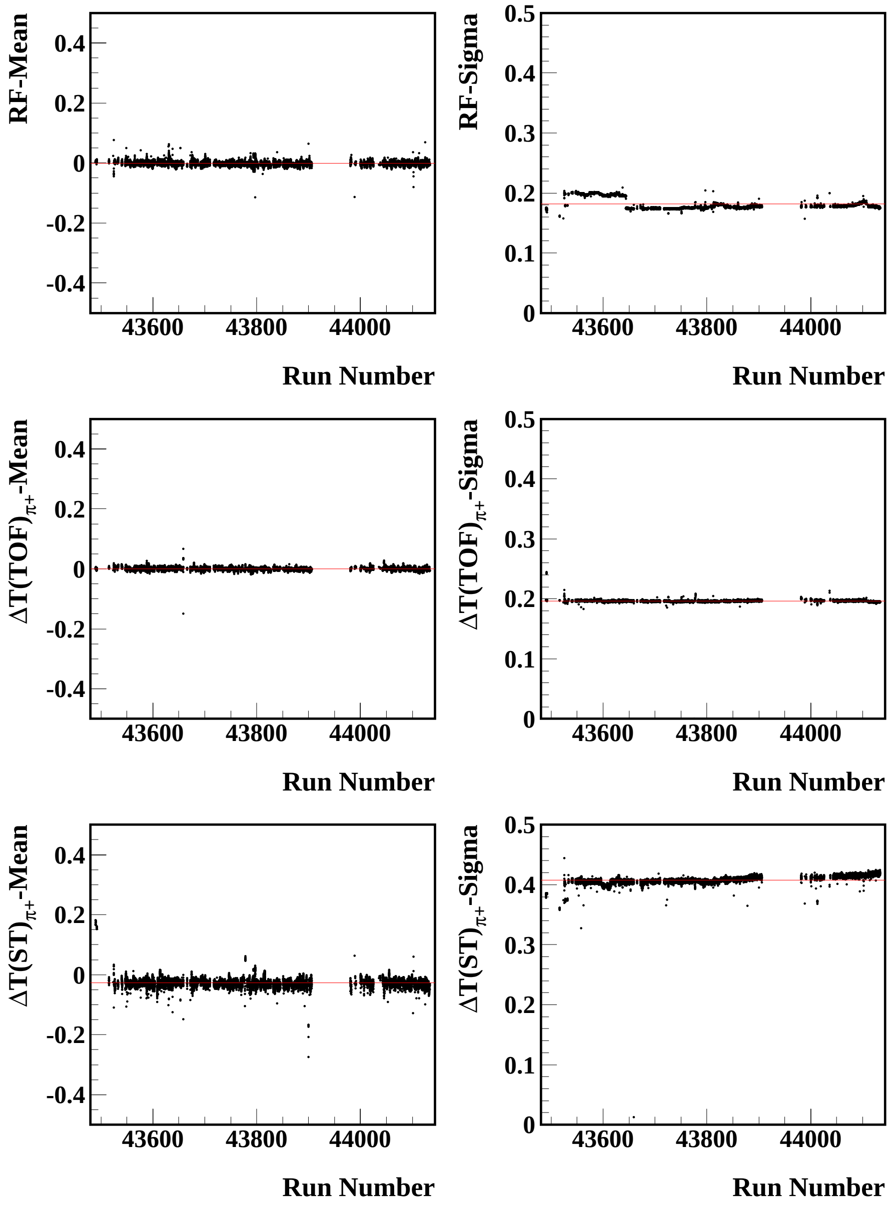

Figure 6: Number of positive pions normalized to the number pf processed events as a function of the run number.

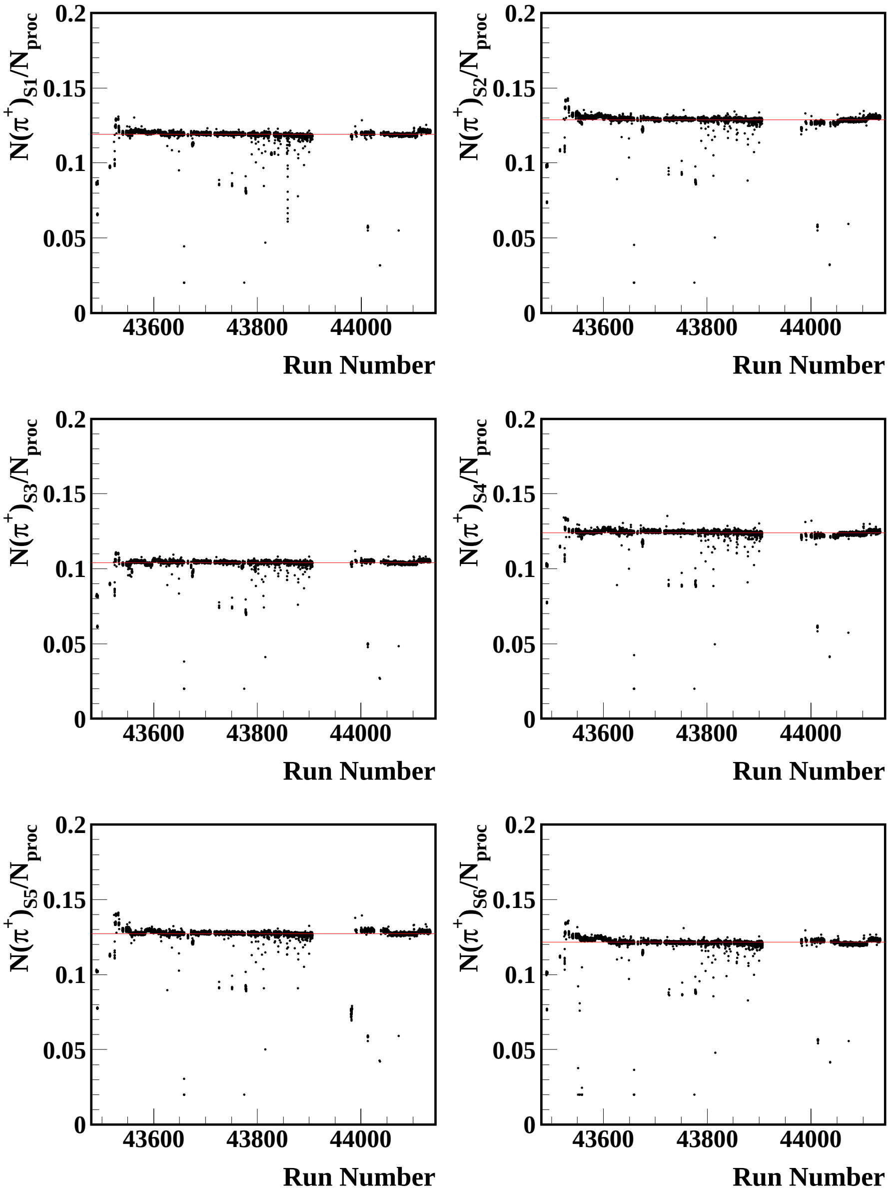

Figure 7: Mean and Sigma of the RF correction, the π^+ TOF and ST vertex time.

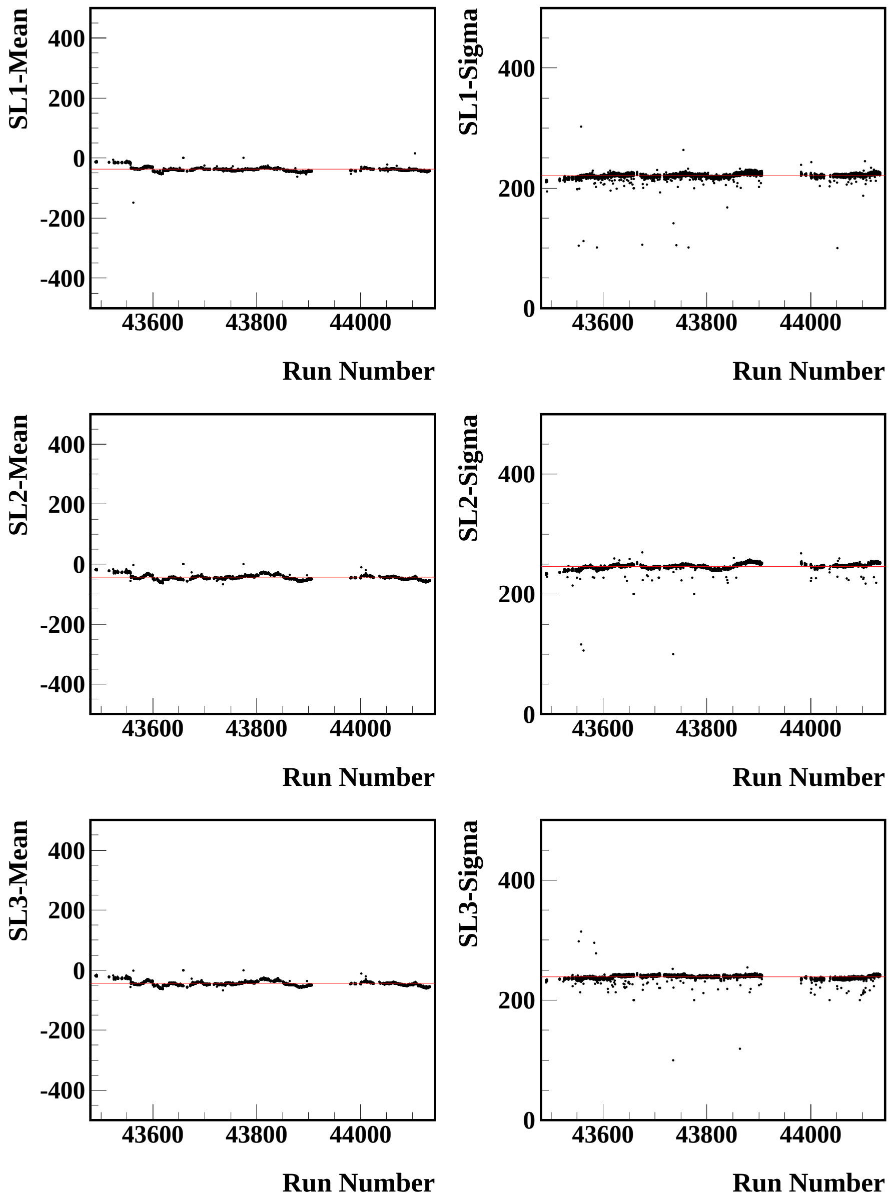

Figure 8: Means and Sigma of the Time Base tracking residuals for superlayer 1, 2 and 3..

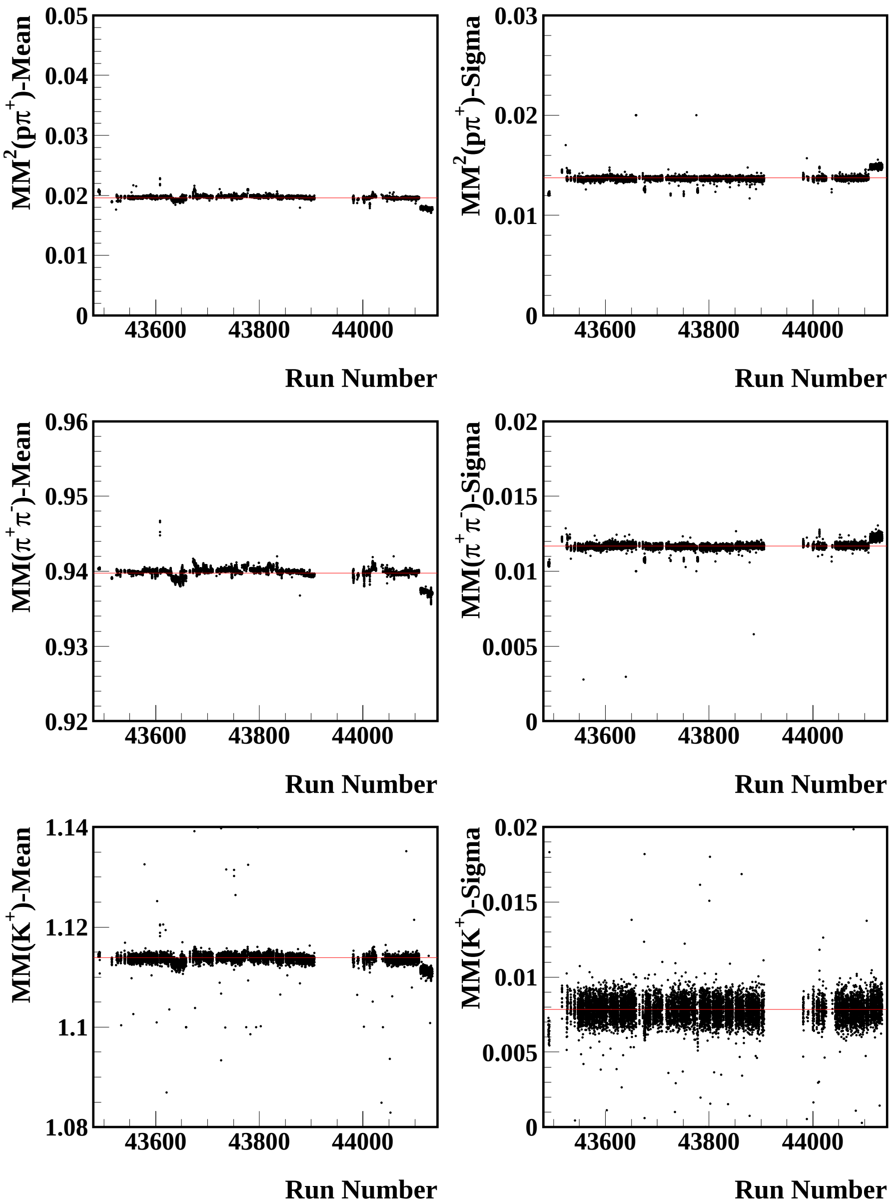

Figure 9: Means and Sigma of the missing mass of $p\pi^+$, $\pi^+\pi^-$, and K^+ as a function of the run number. Variation in the mass position allow to check the beam energy stability.