

Documentation of the Analysis Software of the Conversion Group

K. Aamodt ,F. Bock, P. Braun-Munzinger, P. Gonzalez, M. Ivanov, K. Koch,
S. Lindal, E. López Torres, P. Ladrón de Guevara, A. Marín, K. Reygers,
D. Röhrich, E. Serradilla, J. Stachel, T.S. Tvetter, J.P. Wessels, M. Wilde
for the ALICE Collaboration

December 13, 2010

Contents

1. Introduction	4
2. Afterburning-Software	5
2.1. The Material Macros	5
2.1.1. Plot_Mapping_Histos_Events.C	5
2.1.2. Photon_Characteristics_Events.C	5
2.2. The Meson Macros	6
2.2.1. ExtractSignal.C	7
2.2.2. CorrectSignal.C	11
2.2.3. CutStudiesOverview.C	11
2.2.4. ProduceFinalResults.C	12
2.3. The Gamma Macros	13
2.4. The Afterburning Header Files	13
2.4.1. PlottingMeson.h	13
2.4.2. ExtractSignal.h	14
2.4.3. PlottingGammaConversionHistos.h	14
2.4.4. PlottingGammaConversionAdditional.h	15
2.5. The Scripts	17
2.5.1. start_Material_Comp.sh	18
2.5.2. start_FullMesonAnalyse.sh	18
A. Appendix	21
A.1. Plots of Plot_Mapping_Signal_Events.C	21
A.2. Plots of Photon_Characteristics_Events.C	21
A.3. Plots of ExtractSignal.C	22
A.4. Plots of CorrectSignal.C	23
A.5. Plots of CutStudiesOverview.C	23
A.6. Plots of ProduceFinalResults.C	24

1. Introduction

This document shall present the analysis software of the conversion group, it will describe the Aliroot-software as well as the afterburning macros. In this analysis converted photons are reconstructed from electrons and positrons measured in the central tracking system. Afterwards the photons can be combined to mesons. Furthermore it is possible to reconstruct Dalitz decays with this software as well as to combine 1 photon reconstructed with this method and another one reconstructed in the calorimeters. The analysis is done in two basic steps, one analysing the data directly after the reconstruction, selecting electrons and positrons from the reconstruction and then combining them first to photons and afterwards to mesons. The second step is operating on the level of this output, and will only process data which are available on that level.

2. Afterburning-Software

This section will describe the afterburning macros. It is divided into the explanation of the scripts and the description of the macros themselves which can be divided into 3 main topics, the material macros, the meson macros and the gamma macros.

2.1. The Material Macros

For successfully running the material macros it is necessary to create a new subfolder called “\$Cutnumber” and in this a subfolder \$Suffix“ with respect to the working directory, otherwise the output can not be written.

2.1.1. Plot_Mapping_Histos_Events.C

This macro was designed to produce the plots for the material checks, additionally it provides the calculation of the deviation between data and monte-carlo for one cut.

It can be started by

```
root -l -x -q -b Plot_Mapping_Histos_Events.C\(\"DataFile.root\"\\,\"MCfile.
root\"\\,\"Cutnumber\"\\,\"NameOutputfile\"\\,\"Suffix\"\\,\"optConf\"\\,\"optEn
\\\")
```

The values for accepted for *optConf* are “conf” or ”” and for *optEn* “7TeV”, “900GeV” and “HI”. This macro will produce a summary ps-file with the name “NameOutputfile.ps”, as well as a data-file, containing the systematic error calculation “Export_Mapping.dat”. Moreover single plots will be produced with the file ending \$Suffix, specified in the calling. In the subfolder called “Plot_Mapping_Histos_Events” plots for each R , Z and ϕ bin will appear in separate plots. Furthermore the bidimensional plots will be produced, and the R and Z -distributions in different versions.

The optEn-flag will take care of the correct labelling of the plots and if you run with the *optConf*=“conf”, some plots will be tagged with “conference” in the end, and some improvements will be made to their styling. Furthermore the individual plots in the Z , R and ϕ bins will only be produced in the combined output-file.

2.1.2. Photon_Characteristics_Events.C

The Photon_Characteristics-macro was invented to check the photon properties. Here plots like the dE/dx -distribution, p_t of e^\pm and γ , as well as basic reconstruction properties (DCA, χ^2). It can be started by

```
root -l -x -q -b Photon_Characteristics_Events.C\(\"Datafile.root\"\\,\"MCfile.
root\"\\,\"Cutnumber\"\\,\"NameOutputfile\"\\,\"Suffix\"\\,\"optConf\"\\,\"optEn
\\\")
```

The options for energy and conference have the same meaning as for the Plot_Mapping_Histos_Events.C. Similarly a summarizing output ps-file will be produced and the single plots will be stored in the suffix directory in the subfolder “Photon_Characteristics_Events”.

2.2. The Meson Macros

For the meson macros there are several options which are defined in a common way, they are

- **Meson**
 - * Pi0 - The input will be analysed with respect to π^0 properties
 - * Eta - The input will be analysed with respect to η properties
 - * Pi0EtaBinning - The input will be analysed with respect to π^0 properties, but in the same p_t binning as processed for the η analysis
- **Suffix** - a graphical output format can be chosen according to your needs (like “eps”, “gif” ...).
- **Cutnumber** - this has to be handed over in order to extract which cut of a given output file of the AnalysisTask has to be analysed
- **optEn** - according to this choice, ranges, labels and calculations will be adjusted taking into account the energy dependence
 - * 7TeV
 - * 900GeV
 - * HI
- **optMult**
 - * “Mult” - If this option is chosen, the spectra will be normalised to the selected events after multiplicity choices\ V0AND choices\ centrality choices including vertex requirements. Furthermore the CutStudiesOverview.C will run in a special mode producing comparison plots to show for instance R_{mult} and R_{CP} .
 - * ”” - All spectra will be normalized to the physics selection.
- **optConf**
 - * “conf” - Some plots will be tagged with “conference” in the end, as well as a specific p_t binning might be chosen depending on the energy.
 - * ”” - The usual plots will be produced.
- **optMC**
 - * “kFALSE” - input treated as data, for output files “data” will appear in the name
 - * “kTRUE” - input will be treated as MC-simulation, for output files “MC” will appear in the name

In the next subsections it will be explained what the different macros are doing, what is contained in their output files and where the plots coming from the macros are stored. As already necessary for the material macros we need a file structure, originating in your working directory build up as follow, a directory for each CutNumber containing a directory with the name of your preferred outputfile ending (like “eps”, “gif” ...). If this structure is not provided the output can not be saved.

2.2.1. ExtractSignal.C

The output of the ExtractSignal.C is the basis for all further macros, as it basically provides all histograms which are needed for the further calculation like, efficiencies, acceptance, RAW yields, fitted masses, First of all the structure of the macro will be described and then the functions contained, in the end the outputs will be described.

The extract Signal can be called by:

```
root -x -q -l -b ExtractSignal.C\+\+(\\"$Meson\"\\,\"$RootFile\"\\,\"$Cutnumber
\\,\"$Suffix\"\\,\"$optMC\"\\,\"$optEn\"\\,\"$optFit\"\\,\"$optConf\"\\,\"$
$optMult\"\\,$BinsPtMeson\\)
```

The Input is basically only the \$RootFile, which has to be an output file of the GammaConversion AnalysisTask. The other components of this call are just options, where most of them are common and will have the same effects as for the other macros. However it has to be emphasised, that in the case of Conference running a different binning might be chosen, having less bins in p_t and being especially adapted to be shown in public. The usual runnign mode is just for internal discussions. Depending on this you will have to adjust the \$BinsPtMeson variable, which gives the maximum number of bins taken for this analysis. However if you have to much bins compared to the implemented number it will be set to the maximum for the chosen running mode. Depending wether optMC is set to “kFALSE” or “kTRUE” the correction file will be produced as well and some additional checks on the basis of MC will be performed.

Structural Conception:

The macro ExtractSignal.C first of all reads the 2D invariant mass histograms, where the invariant mass is plotted against p_t of the pair. Afterwards it creates from this an array of 1D histograms summed over a certain p_t range, which depends on the energy option as well as the Meson which was specified. For the background there is the option to created a weighted background in Z-vertex classes as well as multiplicity bins. Having done this one normalizes the background to the signal in a given region next to the signal peak on the right to match the statistics in the signal-histograms. Then the background is substracted from the signal and the signal histogram is fitted with a Gaussian convoluted with an exponential tail and a linear background-function or with a CrystallBall-function. From this fit the mass-position is extracted as well as the FWHM, S/B and Significance. Afterwards the integration region of to get the RAW-yield is shifted according to the deviation of the mass from the nominal PDG-value and the RAW-Yield is calculated substracting the Integral under the linear BG-fit as well in the same region. To evaluate the systematic error of the yield extraction the integration range is varied from very close to the peak, to an intermediate range to a very wide range around the peak. Additionally this procedure is repeated taking a normalisation region for the background on the left hand side of the peak.

After having extracted all this value they are saved in the corresponding histograms and written into a file. Furthermore the plots of invariant mass in p_t bins are created, ones with the signal and the background and ones with the substracted signal and the fit.

For Montecarlo files the true invariant mass spectra are treated in the same way, and the acceptance and efficiency are calculated for each case of the signal extraction and written into a correction-file.

Description of the functions:

- void Calculate GammaCorrection()

This function calculates all γ -spectra in the same binning as for the meson as well as the decay- γ -spectra, purity, reconstruction-efficiency and the conversion probability.

- void ProcesseEM(TH1D* fGammaGamma, TH1D* fBck, Double_t * fBGFitRange)

This function produces a scaled background adjusted in the fBGFitRange to the fGammaGamma-histogram of the fBck-histogram.

- `void ProcessRatioSignalBackground(TH1D* fGammaGamma, TH1D* fBck)`

This function creates the ration of fGammaGamma to fBck.

- `void FillMassHistosArray(TH2F* fGammaGammaInvMassVSPt, TH2F *fAlphaPeakPos)`

This function produces the array of p_t binned invariant mass-histograms of the signal as well as the signal in $||\alpha|| < 0.1$, additionally the signal histogram in the mid and full p_t range is calculated.

- `void FillMassMCTrueMesonHistosArray(TH2F* fHistoTrueMesonInvMassVSPt)`

This function will create the in p_t -binned invariant mass spectra for the true MC spectrum. It can only be called if you are running MC, otherwise the input histogram does not exist.

- `void CreatePtHistos()`

In this function all histograms containing informations which are plotted against p_t are created like: Mass, FWHM, TrueMass,

- `void FillPtHistos()`

In this function all histograms containing informations which are plotted against p_t are filled like: Mass, FWHM, TrueMass,

- `void PlotInvMassSinglePtBin(TH1D* fHistoMappingGGInvMassPtBin, TH1D* fHistoMappingBackNormInvMassPtBin, TString namePlot, TString nameCanvas)`
- `void PlotInvMassInPtBins(TH1D** fHistoMappingGGInvMassPtBin, TH1D** fHistoMappingBackNormInvMassPtBin, TString namePlot, TString nameCanvas, TString namePad)`
- `void PlotInvMassRatioInPtBins(TH1D** fHistoMappingGGInvMassPtBin, TString namePlot, TString nameCanvas, TString namePad)`
- `void PlotWithFitSubtractedInvMassSinglePtBin(TH1D * fHistoMappingSignalInvMassPtBin, TF1 * fFitSignalInvMassPtBin, TString namePlot, TString nameCanvas)`
- `void PlotWithFitSubtractedInvMassSinglePtBin2(TH1D * fHistoMappingSignalInvMassPtBin, TF1 * fFitSignalInvMassPtBin, TString namePlot, TString nameCanvas)`
- `void PlotWithFitSubtractedInvMassInPtBins(TH1D ** fHistoMappingSignalInvMassPtBin, TF1 ** fFitSignalInvMassPtBin, TString namePlot, TString nameCanvas, TString namePad)`
- `void PlotWithFitPeakPosInvMassInPtBins(TH1D ** fHistoMappingSignalInvMassPtBin, TF1 ** fFitSignalInvMassPtBin, TString namePlot, TString nameCanvas, TString namePad)`

These functions will plot the histograms put in the the file specified by “namePlot”. For the PlotWithFitSubtracted*-functions additionally the fitted function will be plotted. For the functions not containing “Single” in the upper right corner the energy, number of events, the process and the ALICE-Logo will appear.

- `void FitSubtractedInvMassInPtBins(TH1D* fHistoMappingSignalInvMassPtBinSingle, Double_t* fMesonIntRange, Int_t ptBin, Bool_t vary)`
- `void FitPeakPosInvMassInPtBins(TH1D* fHistoMappingSignalInvMassPtBinSingle, Double_t * fMesonIntRange, Int_t ptBin, Bool_t vary)`


```
void FitCBSubtractedInvMassInPtBins(TH1D*
    fHistoMappingSignalInvMassPtBinSingle, Double_t * fMesonIntRange, Int_t
    ptBin, Bool_t vary, TString functionname)
```

This functions fit the invariant mass spectra, wher FitSubstractedInvMassInPtBins() will fit with a Gaussian convoluted with an exponential tail, which accounts as well for FitPeakPosInvMassInPtBins(). Only FitCBSubtractedInvMassInPtBins() will fit the spectrum with a CrystallBall-function. This function is not yet final and should not be used.

```
• void ProduceBckWithoutWeighting(TH2F *fBckInvMassVSPt)\
void ProduceBckProperWeighting(TList* fESDContainer, TList*
    fBackgroundContainer)
```

This functions will take care of the proper binning of the background invariant mass spectra in p_t either with or without weighting the background in Z-vertex and multiplicity bins.

```
• void IntegrateHistoInvMass(TH1D * fHistoMappingSignalInvMassPtBinSingle,
    Double_t * fMesonIntRange)
void IntegrateHistoInvMassStream(TH1D *
    fHistoMappingSignalInvMassPtBinSingle, Double_t * fMesonIntRange)
void IntegrateFitFunc(TF1 * fFunc, TH1D *
    fHistoMappingSignalInvMassPtBinSingle, Double_t * fMesonIntRange)
```

This functions take care of the correct integration of the invariant mass spectra in the defined fMesonIntRange.

```
• void CalculateFWHM(TF1 *)
```

Calculates the full width half maximum from the fit on the signal in the invariant mass spectrum.

```
• void FillHistosArrayMC(TH2F* fHistoMCMesonPtEtaWithinAcceptance, TH1D *
    fHistoMCMesonPt, TH1D * fDeltaPt)
```

Creates one dimensional MC-histograms in the correct p_t binning.

```
• void CalculateMesonAcceptance();
void CalculateMesonEfficiency(TH1D*, TString);
```

Calculate the acceptance and efficiencies from the montecarlo data.

```
• void SaveHistos(Int_t, TString, TString);
```

In this function all histograms not needed for the correction of the spectra are saved, consistently for MC and data, like RAW-Yield, mass, FWHM, ..., however in the case of running MC some additional histograms are saved in order to compare them to the data.

```
• void SaveCorrectionHistos(TString, TString);
```

Here all files definitely need as well in the case of not running MC in one of the following macros are saved in order to have some of the MC information still available at the later level.

```
• void Initialize(TString setPi0, Int_t numberOfBins);
```

This function initializes all array which are needed in the further computation. Furthermore it set the on the meson dependend variables to their specific value by checking the content of *setPi0*. The number of bins is given by *numberOfBins*, through this value it can be controlled how far the spectrum should reach.

- `Double_t CrystalBallBck(Double_t *,Double_t *);`
`Double_t CrystalBall(Double_t *,Double_t *);`

This are the functional definitions of the CrystallBall-function with and without linear background.

- `void Delete();`

Here all memory allocated the Initialize() will be set free again.

The Outputs:

The ExtractSignal.C has several dat files as output, further more it will produce at least 1 root file, if running on MC it will even produce two of them.

To keep track of the errors during running the macro an error log file is written. It will contain the information whether the fitting crashed or not. It is name as following:

`$Meson_$optMC_FileErrLog_$Cutnumber.dat`

In addition a data file named

`$Meson_$optMC_EffiCheck_RAWDATA_$Cutnumber.dat`

is produced, containing for each p_t bin the entries in each invariant mass bin, as well as the extracted yields, background and fitted values for the 6 different extraction methods. Furthermore in this file all scaling factors for the weigthing of the background in the different multiplicity and z-bins are displayed. Additionally the fit parameters for the fitting in $|\alpha| < 0.1$ will be written in case of the π^0 at 7 TeV.

The root file named

`$Meson_$optMC_AnalysisResultsWithoutCorrection_$Cutnumber.root`

will be written as standard output file. The first table of histograms are contained no matter, whether you run MC or data, the ones mentioned below just appear if you analyse montecarlo-data.

deltaPt	histoYieldMeson	histoYieldMesonWide
histoYieldMesonPerEvent	histoSignMeson	histoSBMeson
histoMassPosition	histoMassMeson	histoWidthMeson
histoFWHMMeson	histoYieldMesonNarrow	histoYieldMesonPerEventNarrow
histoSBMesonNarrow	histoSignMesonLeftWide	histoSBMesonLeftWide
histoYieldMesonPerEventWide	histoSignMesonWide	histoSBMesonWide
histoYieldMesonLeft	histoYieldMesonLeftPerEvent	histoSignMesonLeft
histoSBMesonLeft	histoMassMesonLeft	histoWidthMesonLeft
histoFWHMMesonLeft	histoYieldMesonLeftNarrow	histoYieldMesonLeftPerEventNarrow
histoSignMesonLeftNarrow	histoSBMesonLeftNarrow	histoYieldMesonLeftWide
histoYieldMesonLeftPerEventWide	histoSignMesonNarrow	
Mapping_BackNorm_InvMass_FullPt	Mapping_GG_InvMass_FullPt	Mapping_Back_InvMass_FullPt
ESD_NumberOfGoodESDTracksVtx	ESD_EventQuality	fESD_ConvGammaPt
Signal_InvMassFit_in_Pt_Bin\$iPt	Mapping_BckNorm_InvMass_in_Pt_Bin\$iPt	fHistoMappingSignalInvMass_in_Pt_Bin\$iPt
Mapping_TrueMeson_InvMass_in_Pt_Bin\$iPt		
fMC_allGammaPt	fHistoGammaMCConvPt1	fHistoGammaESDTrueConvPt1
histoYieldTrueMesonWide	histoTrueSignMeson	MC_Meson_genPt
histoTrueSBMeson	histoYieldTrueMesonNarrow	fHistoMCMesonPtWithinAcceptance
histoYieldTrueMeson	fMCMesonAccepPt	fHistoMCMesonPtEtaWithinAcceptance

The CorrectionFile output is named

`$Meson_MC_AnalysisResultsCorrectionHistos_$Cutnumber.root`

it contains

fMC_allGammaPt	fHistoGammaMCConvPt1	fHistoGammaESDTrueConvPt1
MC_DecayPi0GammaPt	fMC_DecayEtaGammaPt	fMC_DecayEtapGammaPt
fMC_DecayOmegaGammaPt	fMC_DecayRho0GammaPt	fMC_DecayK0sGammaPt
fMCGammaPurity	fMCGammaConvProb	fMCGammaRecoEff
fMCMesonAccepPt	histoTrueMassMeson	histoTrueFWHMMeson
MC_Meson_genPt	ESD_EventQuality	MesonEffiPt
MesonNarrowEffiPt	MesonWideEffiPt	MesonLeftEffiPt
MesonLeftNarrowEffiPt	MesonLeftWideEffiPt	TrueMesonEffiPt
TrueMesonNarrowEffiPt	TrueMesonWideEffiPt	ESD_TruePi0_InvMass_vs_Pt

2.2.2. CorrectSignal.C

The CorrectSignal.C, corrects the meson-spectra, as well as it produces for the basic plots putting the results for all integration ranges and normalization ranges together. It can be called by typing:

```
root -x -l -b -q CorrectSignal.C\("FileWithoutCorr.root"\, "CorrectionFile.
root"\, "$Cutnumber"\, "$Suffix"\, "$Meson"\, "$optMC"\, "$optEn"\, "$
$optMult"\, "$optConf"\)
```

“FileWithoutCorr.root” is the uncorrected output file of the ExtractSignal.C, you will have this for each combination of \$Meson as well as \$optMC, the “CorrectionFile.root” will always be called \$Meson_MC_AnalysisResultsCorrectionHistos.\$Cutnumber.root. You can only put together the files for the same meson, however you can correct MC as well, but only 1 meson can be processed at the same time. If you run on simulated data, you will produce the acceptance and efficiency-plots as well. Furthermore in then the generated MC spectrum is compared to the spectrum resulting from the corrections and efficiencies are compared and plotted.

For the processed cutnumber the systematic error due to the signal extraction is calculated in addition and written to

“\$Meson_\$optMC_SystematicErrorYieldExtraction_\$CutNumber.dat”.

The outputfile it named

“\$Meson_\$optMC_AnalysisResultsCorrection_\$CutNumber.root”

and contains:

CorrectedYieldNormEff	CorrectedYieldTrueEff	CorrectedYieldTrueEffWide
CorrectedYieldTrueEffNarrow	CorrectedYieldTrueEffLeft	CorrectedYieldTrueEffLeftWide
CorrectedYieldTrueEffLeftNarrow	CorrectedYieldTrueEff_Mt	CorrectedYieldTrueEffWide_Mt
CorrectedYieldTrueEffNarrow_Mt	CorrectedYieldTrueEffLeft_Mt	CorrectedYieldTrueEffLeftWide_Mt
CorrectedYieldTrueEffLeftNarrow_Mt	CorrectedYieldTrueEff_Xt	CorrectedYieldTrueEffWide_Xt
CorrectedYieldTrueEffNarrow_Xt	CorrectedYieldTrueEffLeft_Xt	CorrectedYieldTrueEffLeftWide_Xt
CorrectedYieldTrueEffLeftNarrow_Xt	histoYieldMeson	histoFWHMMeson
histoMassMeson	histoSBMeson	histoSignMeson
histoYieldMesonPerEvent	fMCMesonAccepPt	ESD_EventQuality
TrueMesonEffiPt	histoTrueMassMeson	histoTrueFWHMMeson
ESD_NumberOfGoodESDTracksVtx	MC_Meson_genPt	MCYield_Meson
histoRatioComparisonPi0MCtoRec		

2.2.3. CutStudiesOverview.C

The CutStudiesOverview.C was designed to compare the results for different Cuts and extract the systematic error due to this. Now it can produce some more plots like the R_{mult} or R_{CP} plot and the plots for the γ -spectrum comparison as well. What will be produced depends highly on the energy and on the multiplicity-flag. The macro can be started as follows:

```
root -x -q -l -b CutStudiesOverview.C\("CutSelection.log"\, "$Suffix"\, "$
$Meson"\, "$optMC"\, "$optMult"\, "$optEn"\, $NumberOfCuts\)
```

The CutSelection.log specifies which cuts you want to compare, but be aware, that the first cut in the list will be your reference cut and the ratio will always be build to that cut. The variable \$NumberOfCuts specifies how many cuts should be compared. Please make sure that the CutSelection.log has at least so many entries.

Usually the RAW-yields and the corrected yields will be compared and the systematic error calculation will be performed by taking the difference of the corrected yields. Additionally the number of tracks with one vertex will be plotted for each cut, compared in one plot.

If the optMult = "Mult" is chosen and not HI running the R_{mult} will be produced. If you choose multiplicity running and HI is chosen, the different cuts will be normalized to the number of collisions according to the centrality bin chosen in the cut number. Furthermore the corrected plots will not be produced and the systematic errors will not be calculated. Additionally the the R_{CP} plot will be produced. For the systematic error studies a data file called:

\$Meson.\$optMC_SystematicErrorCutStudies.dat

2.2.4. ProduceFinalResults.C

The ProduceFinalResults.C will put all your results together and plot it in a presentable form. Furthermore it will automatically just evaluate the parts which are possible to do depending on the energy you have chosen. This macro is not yet designed to run over HI-data, please do not use it in that case! You can start it by typing.

```
root -x -l -b -q ProduceFinalResults.C\("\$Pi0CorrectedFile.root"\,\"$EtaCorrectedFile.root"\,\"$SystemErrorPi0.dat"\,\"$standardCut"\,\"$CutNumber"\,\"$optMC"\,\"Hagedorn"\,\"$optSameBinningEta"\,\"$optEn\
\"$optConf"\,\"$optMult"\)
```

The options are the same as for all the other macros, except the \$optSameBinningEta which should be empty if you have the usual files for π^0 and η , but if you want to produce the plot η/π^0 you should use "same" for this variable and accordingly the output file off the CorrectSignal.C which has "Pi0EtaBinning" in it. Otherwise the Pi0CorrectedFile.root and EtaCorrectedFile.root are the usual output file for π^0 and η either data or MC. The option "Hagedorn" will make the bin shift of your spectrum according to a Hagedorn-function this can be changed to "Levy" as well. This macro is a little bit different from the others, as it includes additional dependencies compared to the other macros. One of the first things in the code is the definition of the additional includes.

```
if(option.CompareTo("7TeV") == 0){
    collisionSystem = "pp @ 7TeV";
    const char* fileNameNLOPi0 = "ALICENLOcalcPi0Vogelsang7TeV.dat";
    const char* fileNameNLOEta = "ALICENLOcalcEtaVogelsang7TeV.dat";
    const char* fileNameCaloPhos = "PHOS_pi0_7TeV_2010-11-22.root";
    const char* fileNameChargedSpectra = "ChargedSpectra7TeV_20100915.dat";
    const char* fileNameChargedExpectation = "ChargedPythia22092010.dat";
    cout << "You have choosen 7TeV" << endl;
} else if( option.CompareTo("900GeV") == 0) {
    collisionSystem = "pp @ 900GeV";
    const char* fileNameNLOPi0 = "ALICENLOcalcPi0Vogelsang900GeV.dat";
    const char* fileNameCaloPhos = "PHOS_pi0_900GeV_2010_11-29.root";
} else if( option.CompareTo("HI") == 0) {
    collisionSystem = "PbPb @ 2.76TeV";
} else {
    cout << "No correct collision system specification, has been given" << endl;
    return;
}
```

Please make sure that these files exist or rename them, or comment the corresponding parts out. This is not done automatically! Furthermore you have to make sure that the rootfile contain the correct histograms or graph with corresponding names, otherwise you have to adjust these in the

code. The charged spectra have to be given in the format: p_t & value & sys Err & stat Err. Otherwise some thing will be mixed up. For the NLO files it is: p_t & value at $\mu/2$ & value at μ & value at $\mu \cdot 2$.

During running this macro first of all the FWHM, mass, SB, significance and RAW-yield will be produced together for π^0 and η if it is 7TeV running and only π^0 if it is running at 900GeV. Afterwards the efficiency and acceptance will be produced in the same style if you run \$optMC=kTRUE. Afterwards a comparison of the corrected π^0 and η -spectra is performed (including the ratio compared to world data, if you run in the mode where they have the same binning in p_t). Therefore the p_t , m_t and x_t spectra are compared.

Additionally the spectra are fitted with the standard functions in there validity regions: Hagedorn, Levy, Powerlaw, Exponential, Boltzmann. The results are written to the file:

`$optMC.FinalExtraction.$Cutnumber.$optSameBinningEta.Hagedorn.dat,`

as well as the results from the binshift correction with the final factors, which will be the next plot produced. After having done this the spectra are compared to NLO, other π^0 measurements in ALICE and the charged spectra. The same procedure is repeated for η if you run at 7TeV, however there are no official η measurements yet and they are not compared to charged spectra. If you run at 7TeV one additional step will follow the inclusion of systematic error on the π^0 spectrum. As soon as the calculation for η and the 900GeV spectrum are available they will be included as well. The spectrum is then plotted with systematic errors and the spectrum is fitted again.

The last step is writing the following histograms and graphs to a file named:

`$optMC.GammaConversionResultsFullCorrection.$optSameBinninEta.root`

```
RAWYieldPerEventsPi0
CorrectedYieldPi0      Pi0_SystErrorCompl
CorrectedYieldPi0BinShifted  Pi0_SystErrorCompl_BinShifted
RAWYieldPerEventsEta
CorrectedYieldEta
```

2.3. The Gamma Macros

2.4. The Afterburning Header Files

2.4.1. PlottingMeson.h

This header includes the basics for plotting it is compilable and will just be used in the ExtractSignal.C. It has the following functions.

- `void StyleSettings()`

Which sets some basic properties of the canvas, pads and the color scheme.

- `void DrawGammaHisto(TH1* histo1, TString CutSelection, TString Title,
TString XTitle, TString YTitle, Float_t xMin, Float_t xMax, Int_t bck)
{`

This function will plot histo 1 with title as specified in the call, with the x-axis reaching from xMin to xMax as black points and errors if bck=0, and blue histogramm if bck=1.

2.4.2. ExtractSignal.h

In this header all variable for the ExtractSignal.C are predefined as well as the functions needed. In this file you have to adjust the different binning you want to have in p_t

2.4.3. PlottingGammaConversionHistos.h

This header was designed to plot the histograms of the gamma conversion group with different setting the functions are:

- `void StyleSettings()`

Sets some basic settings for canvases and pads, as well as the color palette.

- `void StyleSettingsThesis()`

Sets some more advanced settings for canvases and pads, as well as a multicolor palette.

- `void SetPlotStyle()`

Fancy color palette is specified.

- `void GammaScalingHistogramm(TH1 *histo, Float_t Factor)`
`void GammaScalingHistogramm(TH2 *histo, Float_t Factor)`

Histograms are scaled by Factor.

- `void StylingSliceHistos(TH1 *histo, Float_t markersize)`

Set the marker style to 22 and marker size of the TH1's.

- `void DrawAutoGammaHistos(TH1* histo1,
TH1*histo2,
const char *Title, const char *XTitle, const char *YTitle,
Bool_t YRangeMax, Float_t YMaxFactor, Float_t YMinimum,
Bool_t YRange, Float_t YMin ,Float_t YMax,
Bool_t XRange, Float_t XMin, Float_t XMax)`

`void DrawAutoGammaHisto(TH1* histo1,
const char *Title, const char *XTitle, const char *YTitle,
Bool_t YRangeMax, Float_t YMaxFactor, Float_t YMinimum,
Bool_t YRange, Float_t YMin ,Float_t YMax,
Bool_t XRange, Float_t XMin, Float_t XMax)`

- `void DrawAutoGammaHisto2D(TH2 *histo,
const char *Title, const char *XTitle, const char *YTitle, const
char *Input,
Bool_t YRange, Float_t YMin ,Float_t YMax,
Bool_t XRange, Float_t XMin, Float_t XMax)`

`void DrawRatioGammaHisto(TH1* histo1,
const char *Title, const char *XTitle, const char *YTitle,
Bool_t YRangeMax, Float_t YMaxFactor, Float_t YMinimum,
Bool_t YRange, Float_t YMin ,Float_t YMax,
Bool_t XRange, Float_t XMin, Float_t XMax)`

- `void DrawCutGammaHistos(TH1* histo1, TH1* histo2,
TH1* histo3, TH1*histo4,
const char *Title, const char *XTitle, const char *YTitle, const
char *Legend1, const char *Legend2,
Bool_t YRangeMax, Float_t YMaxFactor, Float_t YMinimum,
Bool_t YRange, Float_t YMin ,Float_t YMax,
Bool_t XRange, Float_t XMin, Float_t XMax)`

- `void DrawCutGammaHisto(TH1* histo1, TH1* histo2,
const char *Title, const char *XTitle, const char *YTitle, const
char *Legend,
Bool_t YRangeMax, Float_t YMaxFactor, Float_t YMinimum,
Bool_t YRange, Float_t YMin ,Float_t YMax,
Bool_t XRange, Float_t XMin, Float_t XMax)`

- `void DrawResolutionGammaHisto(TH1* histo1,
const char *Title, const char *XTitle, const char *YTitle,
Bool_t YRangeMax, Float_t YMaxFactor, Float_t YMinimum,
Bool_t YRange, Float_t YMin ,Float_t YMax,
Bool_t XRange, Float_t XMin, Float_t XMax)`

- `void DrawAutoGammaHisto2DRes(TH2 *histo,
const char *Title, const char *XTitle, const char *YTitle,
const char *Input,
Bool_t YRange, Float_t YMin ,Float_t YMax,
Bool_t XRange, Float_t XMin, Float_t XMax)`

- `void DrawAutoGammaMesonHistos(TH1* histo1,
const char *Title, const char *XTitle, const char *YTitle,
Bool_t YRangeMax, Float_t YMaxFactor, Float_t YMinimum,
Bool_t YRange, Float_t YMin ,Float_t YMax,
Bool_t XRange, Float_t XMin, Float_t XMax)`

- `void DrawGammaSetMarker(TH1* histo1,
Style_t markerStyle, Size_t markerSize, Color_t markerColor,
Color_t lineColor)`
- `void DrawGammaSetMarkerTGraph(TGraph* graph,
Style_t markerStyle, Size_t markerSize, Color_t markerColor,
Color_t lineColor)`
- `void DrawGammaSetMarkerTGraphAsym(TGraphAsymmErrors* graph,
Style_t markerStyle, Size_t markerSize, Color_t markerColor,
Color_t lineColor)`
- `void DrawGammaSetMarkerTF1(TF1* fit1,
Style_t lineStyle, Size_t lineWidth, Color_t lineColor)`

- `void DrawGammaCanvasSettings(TCanvas* c1, Double_t leftMargin, Double_t
rightMargin, Double_t topMargin, Double_t bottomMargin)`
- `void DrawGammaPadSettings(TPad* pad1, Double_t leftMargin, Double_t
rightMargin, Double_t topMargin, Double_t bottomMargin)`

2.4.4. PlottingGammaConversionAdditional.h

This header is used to make plotting of logos, texts and lines easier for the conversion plots. The functions are:

- `void DrawAliceLogoPiOPerformance(Float_t startTextX, Float_t startTextY, Float_t startPiOTextX, Float_t differenceText, Float_t startLogoX, Float_t startLogoY, Float_t widthLogo, Float_t textSize, Float_t nEvents, TString collisionSystem, Bool_t mcFile , Bool_t rawData, Bool_t piOLabel)`

```
void DrawAliceLogoPiOMC(Float_t startTextX, Float_t startTextY, Float_t startPiOTextX, Float_t differenceText, Float_t startLogoX, Float_t startLogoY, Float_t widthLogo, Float_t textSize, Float_t nEvents, TString collisionSystem, Bool_t mcFile , Bool_t rawData, Bool_t piOLabel)
```

- `void DrawAliceLogoPiOPreliminary(Float_t startTextX, Float_t startTextY, Float_t startPiOTextX, Float_t differenceText, Float_t startLogoX, Float_t startLogoY, Float_t widthLogo, Float_t textSize, Float_t nEvents, TString collisionSystem, Bool_t mcFile , Bool_t rawData, Bool_t piOLabel)`

- `void DrawAliceLogoCombined(Float_t startTextX, Float_t startTextY, Float_t startPiOTextX, Float_t differenceText, Float_t startLogoX, Float_t startLogoY, Float_t widthLogo, Float_t textSize, Float_t nEvents, TString collisionSystem, Bool_t mcFile , Bool_t rawData, Bool_t piOLabel)`

```
void DrawAliceLogoCombinedMC( Float_t startTextX, Float_t startTextY, Float_t startPiOTextX, Float_t differenceText, Float_t startLogoX, Float_t startLogoY, Float_t widthLogo, Float_t textSize, Float_t nEvents, TString collisionSystem, Bool_t mcFile , Bool_t rawData, Bool_t piOLabel)
```

- `void DrawAliceLogoCombinedPreliminary(Float_t startTextX, Float_t startTextY, Float_t startPiOTextX, Float_t differenceText, Float_t startLogoX, Float_t startLogoY, Float_t widthLogo, Float_t textSize, Float_t nEvents, TString collisionSystem, Bool_t mcFile , Bool_t rawData, Bool_t piOLabel)`

```
void DrawAliceLogo(Float_t startX, Float_t startY, Float_t widthLogo, Float_t textHeight)
```

- `void DrawAliceLogo1D(Float_t startX, Float_t startY, Float_t widthLogo, Float_t textHeight)`

```
void DrawAliceLogoPerformance(Float_t startX, Float_t startY, Float_t widthLogo, Float_t textHeight, Float_t decrease, char *date, TString collisionSystem)
```

- `void DrawAliceLogoMC(Float_t startX, Float_t startY, Float_t widthLogo, Float_t textHeight, Float_t decrease, char *date, TString collisionSystem)`

```
void DrawAliceLogoPerformance2D(Float_t startX, Float_t startY, Float_t widthLogo, Float_t textHeight, Float_t decrease, char *date, TString collisionSystem)
```


- `void DrawAliceText(Float_t startX, Float_t startY, Float_t textHeight)`
- `void DrawStructure()`
- `void DrawArmenteros()`
- `void DrawdEdxLabel()`
- `void DrawGammaLines(Float_t startX, Float_t endX,
Float_t startY, Float_t endY,
Float_t linewidth)`

2.5. The Scripts

To make the running of the macros easier we introduced several shell scripts controlling the different combinations of macros, but still not all macros are included in this scripts. The main deviation at the moment is the splitting in material analysis and the combined meson and gamma analysis. Both scripts can be started in general by typing

```
sh $scriptname.sh [option] Datafile.root MCfile.root Outputformat
```

Here `scriptname.sh` represents the script which you want to run, `Datafile.root` and `MCfile.root` represent the locations of corresponding root-files. The variable `Outputformat` can be any picture ending which you would like to be produced. For this ending in each cut-directory a folder will be created named by the output-format. We can not assure that for all output-formats the pictures are exactly the same (for example in pdf output the ALICE-logo is missing), but in general it should work with all of them. Please be aware, that this files should include the same cutnumbers, otherwise the analysis will break at some point. Common options are:

- `-h`
- `-help`

Both of them will display some informations about how to run the macro. Furthermore a very short helper will be displayed if you start the scripts without any arguments. For all scripts in the beginning will be checked whether all necessary files for running do exist in the directory where you execute your script, if something is missing the script will be aborted by saying what is missing. During running this scripts you will be asked some questions, this which are in common will be discussed now.

- "Do you want to take an already existing CutSelection.log-file. Yes/No"
 - * **Yes** - the file "CutSelection.log", which should exist in the directory, where you are running, will be processed
 - * **No** - a new "CutSelection.log" file will be created from the `Datafile.root` and which will contain all cutnumbers included in this file.
- "Which collision system do you want to process? 7TeV (pp@7TeV), 900GeV (pp@900GeV), HI (PbPb@2.76GeV)"
 - * **7TeV** - All labels and binnings will be adjusted according to 7 TeV, furthermore the calculation of x_t will be adjusted. In case of the `start_FullMesonAnalyse.sh` π^0 and η mesons will be analysed.

- * **900GeV** - All labels and binnings will be adjusted according to 900 GeV, furthermore the calculation of x_t will be adjusted. In case of the start_FullMesonAnalyse.sh only π^0 's will be analysed.
- * **HI** - All labels and binning will be adjusted according to PbPb running at 2.76 TeV, as well as in the start_FullMesonAnalyse.sh only π^0 's will be analysed with a first attempt to correct the data. The final plots will be missing.
- "Do you want to produce conference plots? Yes/No?"
 - * **Yes** - Some of the plots will be produced with a different scaling and a tag "conference" in the end of the file name, which will be copied to a dedicated conference directory in each cutdirectory in case of material analysis and in the standard Cutdirectory in case of meson analysis. Furthermore in the meson analysis the chosen binning for conference running and usual running is different and the p_t bins will be chosen accordingly-
 - * **No** - For the meson analysis the usual binning will be chosen and no plots will be tagged with conference in their names.
- "Please check that you really want to process all cuts, otherwise change the CutSelection.log. Remember at first all gamma cutstudies will be carried out. Make sure that the standard cut is the first in the file. Continue? Yes/No?";
 - * **Yes** - The analysis will be continued with the cuts specified in the "CutSelection.log". If you want to modify the cutnumbers edit this file from a different shell and afterwards, confirm.
 - * **No** - The running will be aborted and the script will be finished.

2.5.1. start_Material_Comp.sh

This script is designed to produce basic files for the systematic error estimation of the material budget. You can start the script by typing During running you will be asked to specify the MC sample which you are running as it can not be extracted from the file name, there you have the options to either chose Pythia or Phojet, according to this choice a directory will be created in which all the outputs will be put. This is necessary to evalutated both MC-generators for the systematic error of the material separately and keep track of what you are doing. Afterwards the script will automatically run for each cut the two material macros:

- Photon_Characteristics_Events.C
- Plot_Mapping_Histos_Events.C

It is forseen to include in this the Photon_Resolution.C and the Cuts_Events.C as well, but there are still some modifications necessary for this. The output will appear in the MC-Generator-directory under the subfolder for the cut and the subfolder for the specified output format.

2.5.2. start_FullMesonAnalyse.sh

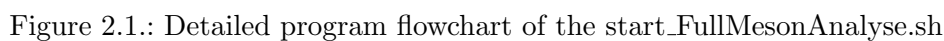
This script is a little bit more complex as it controlls the running of the full meson and gamma analysis, which has several components, being arranged differently depending on the energy you choose. First of all the script has some additional options:

- **-c** - The script will not process the material macros as well as the ExtractSignal.C, it will directly step in at the point of correcting the already existing file, therefore one full run has to have been done before.

- **-d** - Will step in even after the CorrectSignal.C and will only fulfil the CutStudiesOverview.C and the final macros for the meson and gamma analysis.
- **-r** - Only the final macros for the analysis will be carried out for the specified standard cut.

Running without any option will start really at the beginning of the combined analysis. A detailed programm flowchart of the script can be found in fig. 2.1. Either the outputfile is automatically forwarded to the next macro needing this input, or the information to build the name of the files. The detailed description of the macros has already been given in the previous subsections. During running the script you will be asked some additional questions, like how many bins you want to have for the meson analysis corresponding to the energy you are running (giving by this the maximum p_t for the analysis). Additionally you will be asked the following:

- "What is your standard cut? "
- * You should select 1 Cut from the CutSelection.log, which will be displayed above this question. Furthermore you should afterwards arrange the CutSelection.log like this, that the standard cut is the first in your file, as this will be taken as reference cut for the CutStudiesOverview.C. Only for the standard cut the final macros (ProduceFinalResults.C, CalculateGammaToPi0.C) will be carried out, so choose wise.
- "Which fit do you want to do? CrystalBall or gaussian convoluted with an exponential function? CrystalBall/Gaussian? "
- * **Gaussian** - A Gaussian function modified with an exponential tail and a linear background will be used to fit the invariant mass spectrum in the ExtractSignal.C, from this function the FWHM and mass position will be determined.
- * **CrystalBall** - The fitting of the initial invariant mass distribution is implemented with the CrystalBall-function as well, however this function is not behaving as expected. Therefore it is better to use this flag just for debugging.
- "Do you want to perform multiplicity studies? Yes/No? "
- * **Yes** - Each Cut will be normalised to the number of events after basic event selection, like vertex cuts, multiplicity or centrality cuts. This should only be used for multiplicity, centrality or V0OR/V0AND studies. Additionally in the CutStudiesOverview.C some more plots will be produced to show the multiplicity or centrality dependence of the spectra (like R_{mult} or R_{CP}).
- * **No** - Each Cut will be normalised to the number of events after AliPhysicsSelection, no further reduction of the events is taken into account for normalisation.



A. Appendix

A.1. Plots of Plot_Mapping_Signal_Events.C

The plots will appear in the \$Suffix\Plot_Mapping_Signal_Events directory.

- Diff_XY_distributions.\$Suffix
- R_distribution_lin.\$Suffix
- R_distribution_log.\$Suffix
- R_distributionSDDSSDENlarged_log.\$Suffix
- R_distributionSPDENlarged_log.\$Suffix
- R_distributionTPCENlarged_log.\$Suffix
- XY_distribution.\$Suffix
- XY_distribution_minimum.\$Suffix
- Z_distribution_lin.\$Suffix
- Z_distribution_log.\$Suffix
- ZR_distribution.\$Suffix

If you do not run in the conference mode there will be some other histograms be produced for the different Z -Bins, R -Bins and ϕ bins, as well as some specialized ones for the FMD.

A.2. Plots of Photon_Characteristics_Events.C

The plots will appear in the \$Suffix\Photon_Characteristics_Events directory.

- ArmenterosAftCuts_data.\$Suffix
- ArmenterosAftCuts_montecarlo.\$Suffix
- Armenterosfull_data.\$Suffix
- Armenterosfull_montecarlo.\$Suffix
- Asymetry_data.\$Suffix
- Asymetry_montecarlo.\$Suffix
- Clusters.\$Suffix
- dEdx_E_data.\$Suffix
- dEdx_E_montecarlo.\$Suffix

- dEdx_P_data.\$Suffix
- dEdx_P_montecarlo.\$Suffix
- Electron_Pt.\$Suffix
- Mother_InvMass.\$Suffix
- NumberOfGoodESDTracks.\$Suffix
- Photon_Chi2.\$Suffix
- Photon_ConvProb_conference.\$Suffix
- Photon_CosPointingAngle.\$Suffix
- Photon_DcaDaughters.\$Suffix
- Photon_Eta.\$Suffix
- Photon_Likelihood.\$Suffix
- Photon_Mass.\$Suffix
- Photon_NormDcaDaughters.\$Suffix
- Photon_Pt.\$Suffix
- Photon_Purity_conference.\$Suffix
- Photon_RecoEff_conference.\$Suffix
- Pi0_alpha.\$Suffix
- Positron_Pt.\$Suffix
- ProjectionAsymmetry.\$Suffix
- ProjectiondEdx.\$Suffix
- V0_Cut_InvMass.\$Suffix

A.3. Plots of ExtractSignal.C

This plots will appear in the \$Suffix subfolder under the ExtractSignal folder. For the all mesons which were selected and for MC and data.

- \$Meson_\$optMC_MesonSBRatio_\$Cutnumber.\$Suffix
- \$Meson_\$optMC_MesonFittedPeakPos_\$Cutnumber.\$Suffix
- \$Meson_\$optMC_MesonSubtracted_\$Cutnumber.\$Suffix
- \$Meson_\$optMC_MesonSubtractedFittingMidPt_\$Cutnumber.\$Suffix
- \$Meson_\$optMC_MesonWithBck_\$Cutnumber.\$Suffix
- \$Meson_\$optMC_MesonWithBckLeft_\$Cutnumber.\$Suffix
- \$Meson_MC_TrueMesonFitted_\$Cutnumber.\$Suffix

A.4. Plots of CorrectSignal.C

The plots will appear in the `$Suffix\CorrectSignal` directory. For all selected mesons as well as MC and data.

- `$Meson_Acceptance_$Cutnumber.$Suffix`
- `$Meson_$optMC_ComparisonPi0Yields_$Cutnumber.$Suffix`
- `$Meson_$optMC_ComparisonPi0Yields_true_$Cutnumber.$Suffix`
- `$Meson_$optMC_CorrectedYield_Mtspectra_TrueEff_$Cutnumber.$Suffix`
- `$Meson_$optMC_CorrectedYield_TrueEff_$Cutnumber.$Suffix`
- `$Meson_$optMC_CorrectedYield_Xtspectra_TrueEff_$Cutnumber.$Suffix`
- `$Meson_$optMC_FWHM_$Cutnumber.$Suffix`
- `$Meson_$optMC_InvMassFullPt_$Cutnumber.$Suffix`
- `$Meson_$optMC_InvMassFullPtWithoutBG_$Cutnumber.$Suffix`
- `$Meson_$optMC_Mass_$Cutnumber.$Suffix`
- `$Meson_$optMC_RAWYieldPt_$Cutnumber.$Suffix`
- `$Meson_$optMC_SB_$Cutnumber.$Suffix`
- `$Meson_$optMC_Sign_$Cutnumber.$Suffix`
- `$Meson_histoMCYieldMeson_$Cutnumber.$Suffix`
- `$Meson_MC_Efficiency_$Cutnumber.$Suffix`
- `$Meson_MC_EfficiencyCompared_$Cutnumber.$Suffix`
- `$Meson_MC_TrueEfficiency_$Cutnumber.$Suffix`
- `$Meson_TrueEffSimple_$Cutnumber.$Suffix`

A.5. Plots of CutStudiesOverview.C

Not all plots will appear in each running mode, but they will appear in the *CutStudies*-directory in your main running dir. They will be produced separately for MC and data, as well as for each meson.

- `$Meson_$optMC_CorrectedYield.$Suffix`
- `$Meson_$optMC_CORRGamma.$Suffix`
- `$Meson_$optMC_NchDistribution.$Suffix`
- `$Meson_$optMC_RAWGamma.$Suffix`
- `$Meson_$optMC_RAWYield.$Suffix`
- `$Meson_$optMC_RCP.$Suffix`
- `$Meson_$optMC_Rmult.$Suffix`

A.6. Plots of ProduceFinalResults.C

Not all plots will appear in each running mode, but they will appear in the *ProduceFinalResults*-directory in your cut directory in \$Suffix. They will be produced separately for MC and data, as well as for each meson.

- AcceptanceCombined_\$Cutnumber_.eps
- \$optMC_CombinedCorrectedYield_\$Cutnumber.eps
- \$optMC_CombinedCorrectedYieldMt_\$optSameBinningEta_\$Cutnumber.eps
- \$optMC_CombinedCorrectedYield_\$optSameBinningEta_\$Cutnumber.eps
- \$optMC_CombinedCorrectedYieldXt_\$optSameBinningEta_\$Cutnumber.eps
- \$optMC_CorrectedYieldOnlyPi0_\$Cutnumber.eps
- \$optMC_FWHMcombined_\$optSameBinningEta_\$Cutnumber.eps
- \$optMC_Mass_Eta_combined_\$optSameBinningEta_\$Cutnumber_conference.eps
- \$optMC_Mass_Pi0_combined_\$optSameBinningEta_\$Cutnumber_conference.eps
- \$optMC_Pi0EtaRatioPythia_\$optSameBinningEta_\$Cutnumber.eps
- \$optMC_Pi0EtaRatio_\$optSameBinningEta_\$Cutnumber.eps
- \$optMC_Pi0EtaRatioWorld_\$optSameBinningEta_\$Cutnumber.eps
- \$optMC_RAWYieldPtCombined_\$optSameBinningEta_\$Cutnumber.eps
- \$optMC_SBcombined_\$optSameBinningEta_\$Cutnumber.eps
- \$optMC_Signcombined_\$optSameBinningEta_\$Cutnumber.eps
- Eta_\$optMC_CorrectedYieldBinShifted_Hagedorn_\$Cutnumber.eps
- Eta_\$optMC_CorrectedYieldFitted_\$Cutnumber.eps
- Eta_\$optMC_CorrectedYieldwithNLO_\$Cutnumber_\$optSameBinningEta.eps
- Pi0_\$optMC_CorrectedYieldBinShifted_Hagedorn_\$Cutnumber_\$optSameBinningEta.eps
- Pi0_\$optMC_CorrectedYieldFitted_\$Cutnumber_\$optSameBinningEta.eps
- Pi0_\$optMC_CorrectedYieldwithNLO_\$Cutnumber_\$optSameBinningEta.eps
- Pi0_\$optMC_CorrectedYieldwithOtherPi0Meas_\$Cutnumber_\$optSameBinningEta.eps
- Pi0_\$optMC_CorrectedYieldwithOtherPi0MeasInclSyst_\$Cutnumber_.eps
- Pi0_\$optMC_CorrectedYieldwithOthers_\$Cutnumber_\$optSameBinningEta.eps
- Pi0_\$optMC_CorrectedYieldwithSysError_\$Cutnumber_.eps
- Pi0_\$optMC_CorrectedYieldwithSysErrorBinShifted_\$Cutnumber_.eps
- Pi0_\$optMC_CorrectedYieldwithSysErrorNLOBinShifted_simple_\$Cutnumber_.eps
- Pi0_\$optMC_CorrectedYieldwithSysErrorNLO_simple_\$Cutnumber_.eps
- TrueEffOnlyPi0_\$Cutnumber_\$optSameBinningEta.eps