Studies with the ZEUS Common Ntuple, working with dCache

Summer student program
Hamburg, DESY
25.07.07-18.09.07

Oleksandr Volynets
Kiev, KNU

1. Working with ZEUS common ntuples

1. My program for structuring common ntuples (ntuples_to_minintuples.cxx).

1. Working with ZEUS common ntuples

1. My program for structuring common ntuples (ntuples_to_minintuples.cxx).

Idea of common ntuples:
Keep all data from ther ZEUS detector at HERA in root and paw files in *"common ntuples"*. Every group can work with the same data. More info:
http://www-zeus.desy.de/ZEUS_ONLY/analysis/comntp

What is Root:
It is analysis framework for different types of analysis. More info: *http://root.cern.ch*

What are ntuples (in root they are also called "tree"):
It is set of parameters for further analysis written in root and paw files format.

Problem with root Common ntuples:
Many *parameters* in one *tree* (600 parameters ➔ usually needed only ~20-50) ➔ rather time consuming.

Idea of solving the problem:
Write each ntuple block to it's own *tree* according to orange table.

Why is it useful???:
Not so many *parameters* in one *tree* ➛ not so time consuming at last (5 times!!!! faster for my sample).

First reviews:

1. Achim Geiser: Let users choose if they want to separate blocks or keep them together.
2. Maks Borodin: You can use dCache if you use root 5.14.00 just writing in path zeus://acs/.....

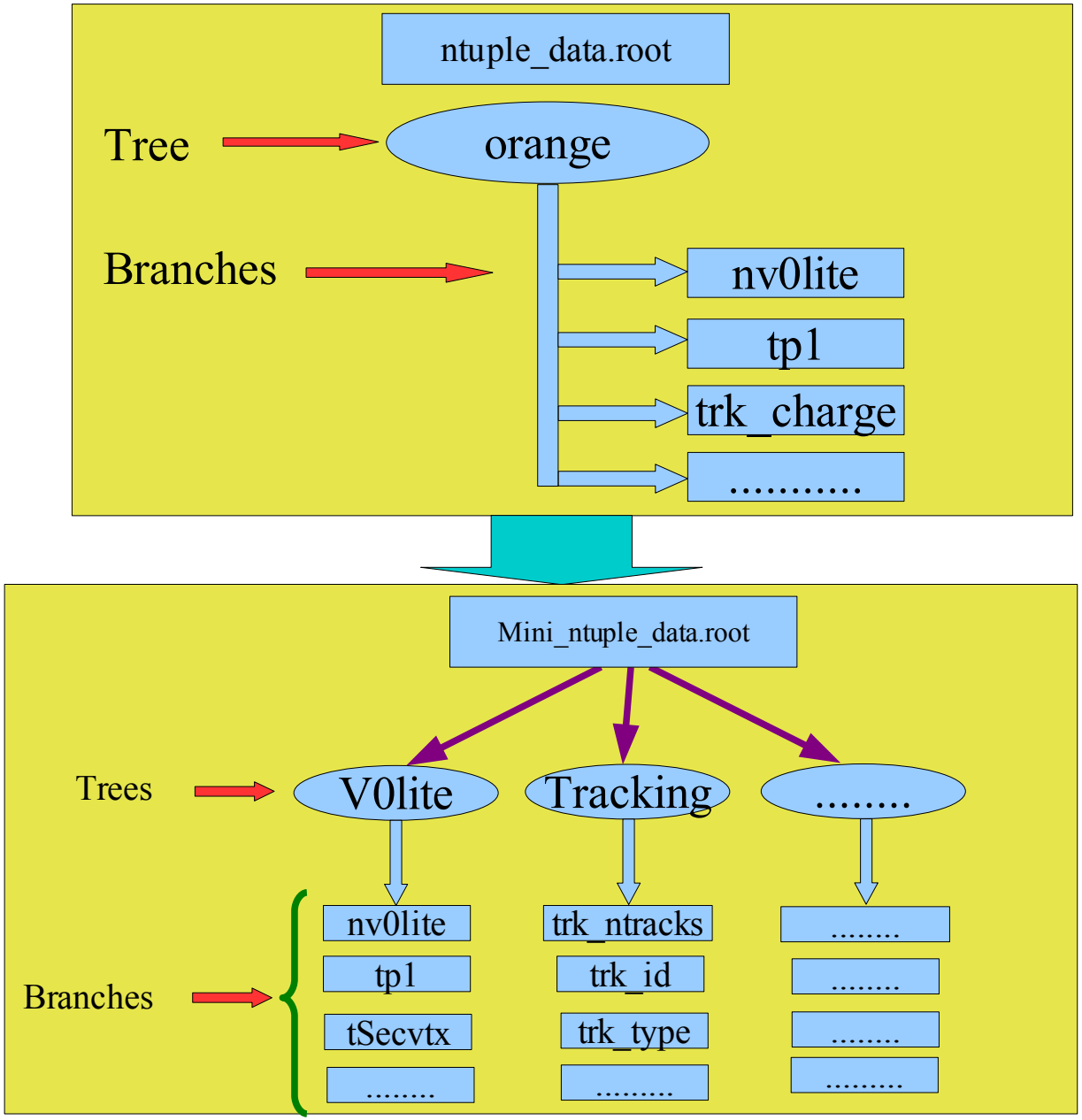


Figure 1.
Idea of structuring data in ZEUS Common ntuples

Computing realization:

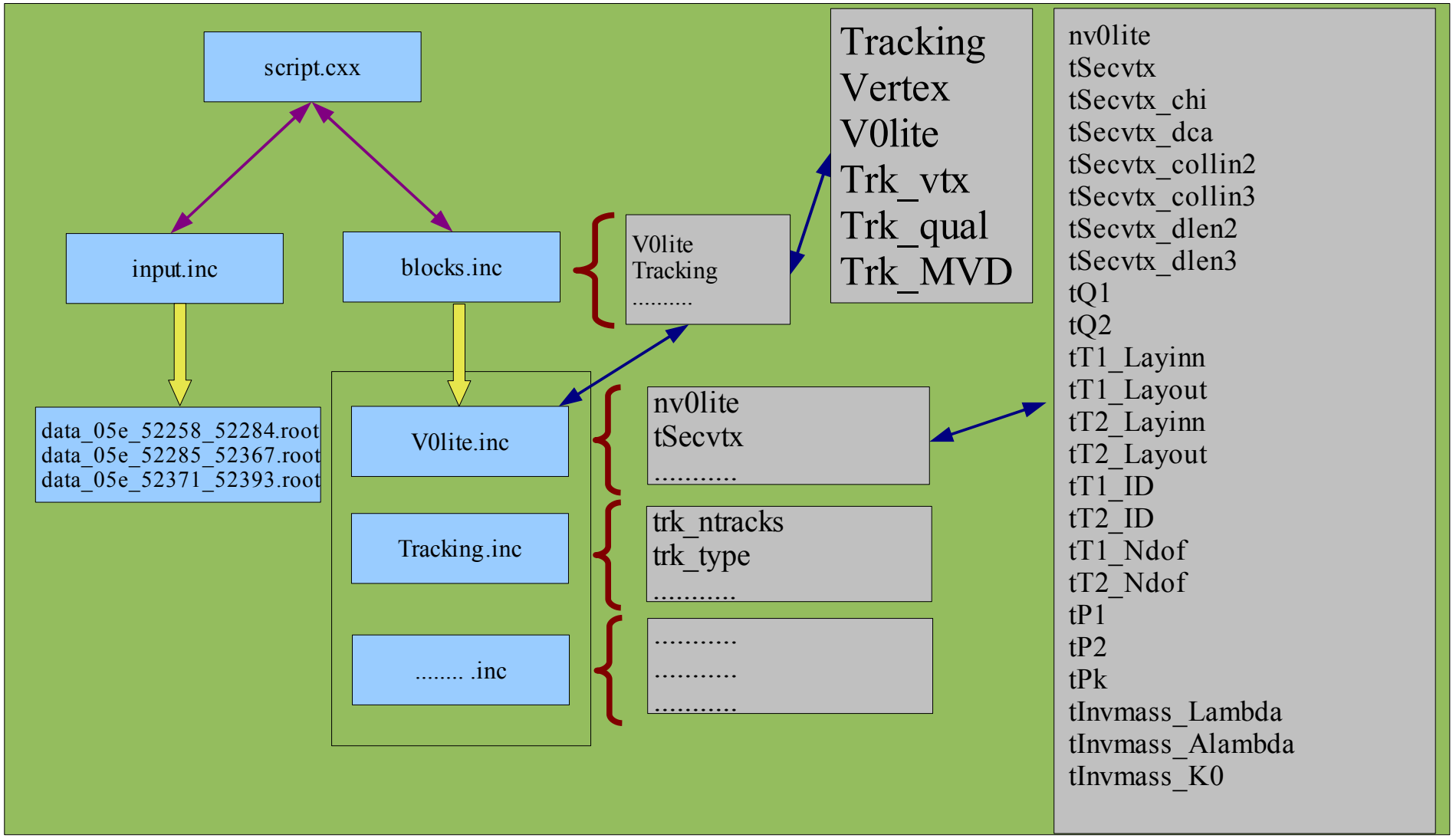


Figure 2. Structure of script

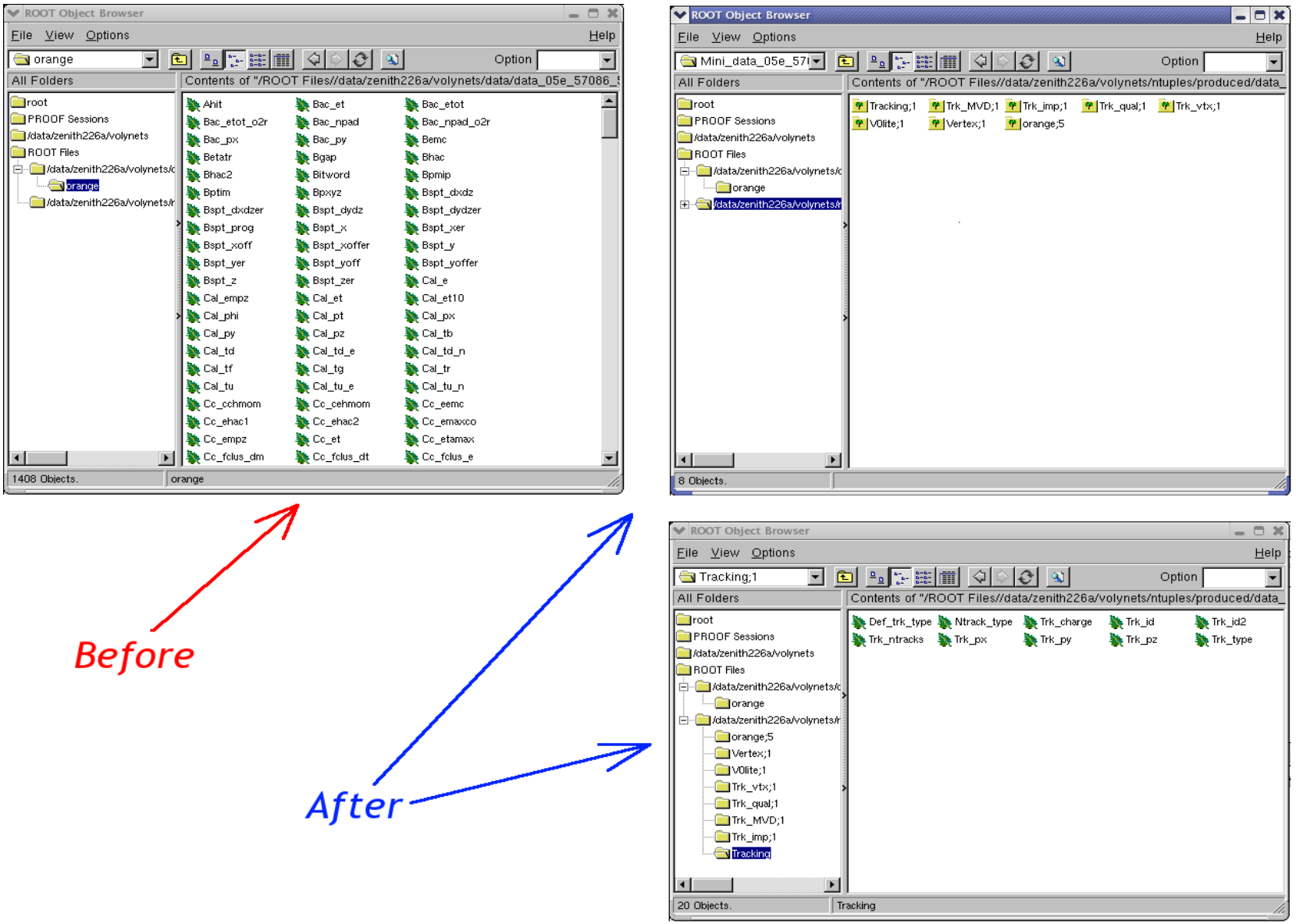


Figure 3. Illustraton of converting common ntuple files to mini_ntuples.

User interface:

1)Choose blocks (file blocks.inc) that you need (or don't need).

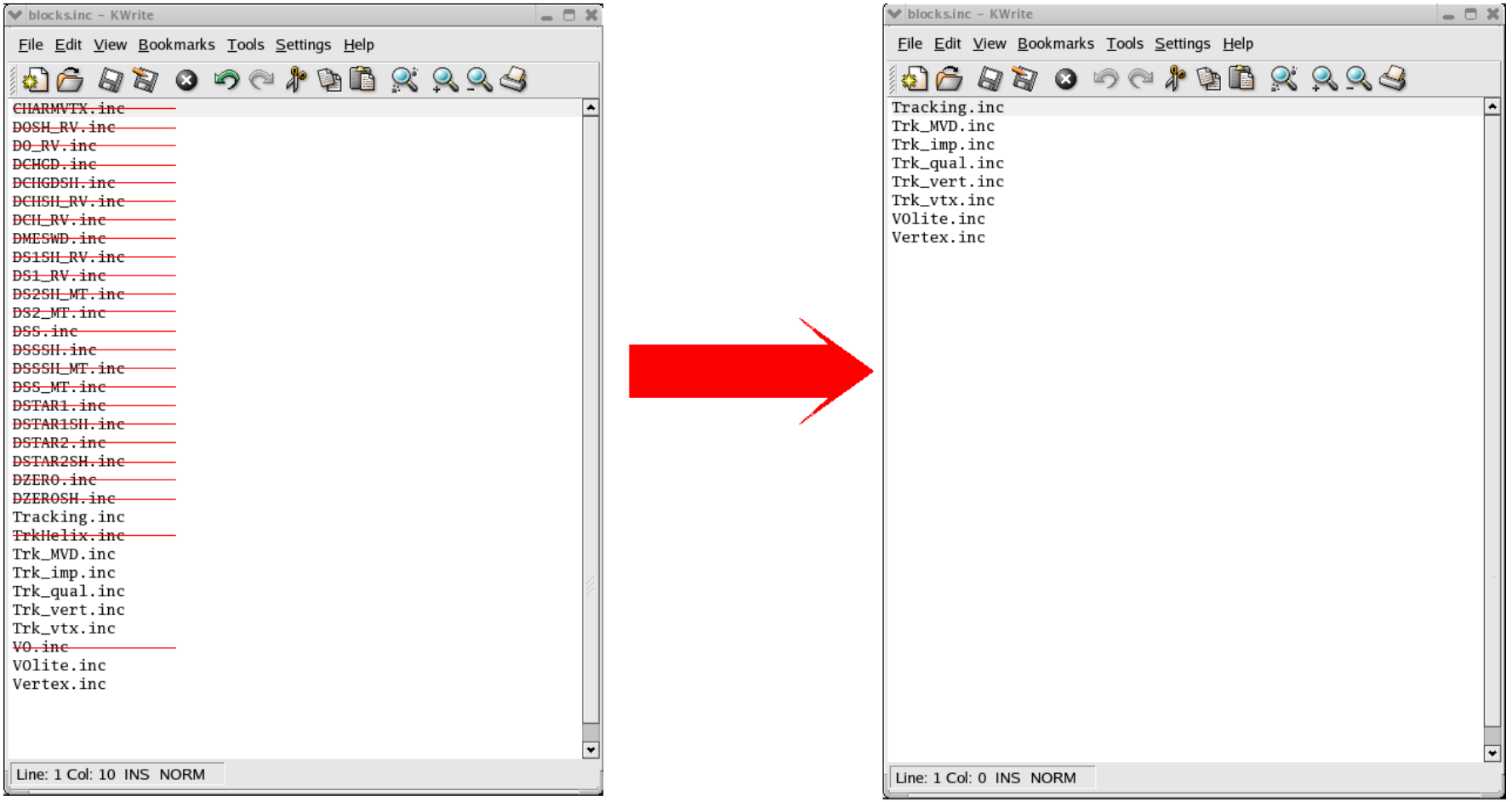


Figure 4. Setecting needed blocks.

2)Change [optional] branches you need and don't
(file Tracking.inc for example).

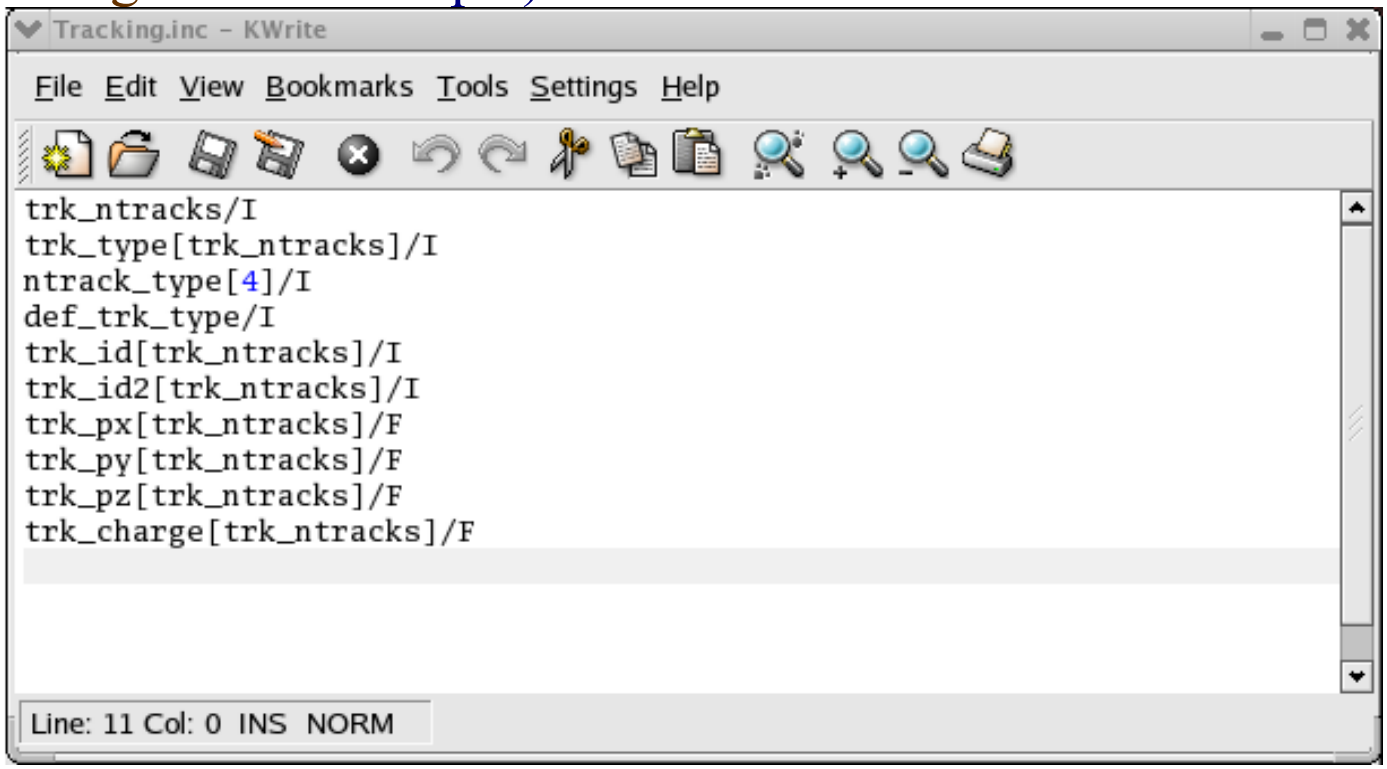


Figure 5. Selecting needed parameters in one of the blocks.

3)Choose input files (file input.inc).

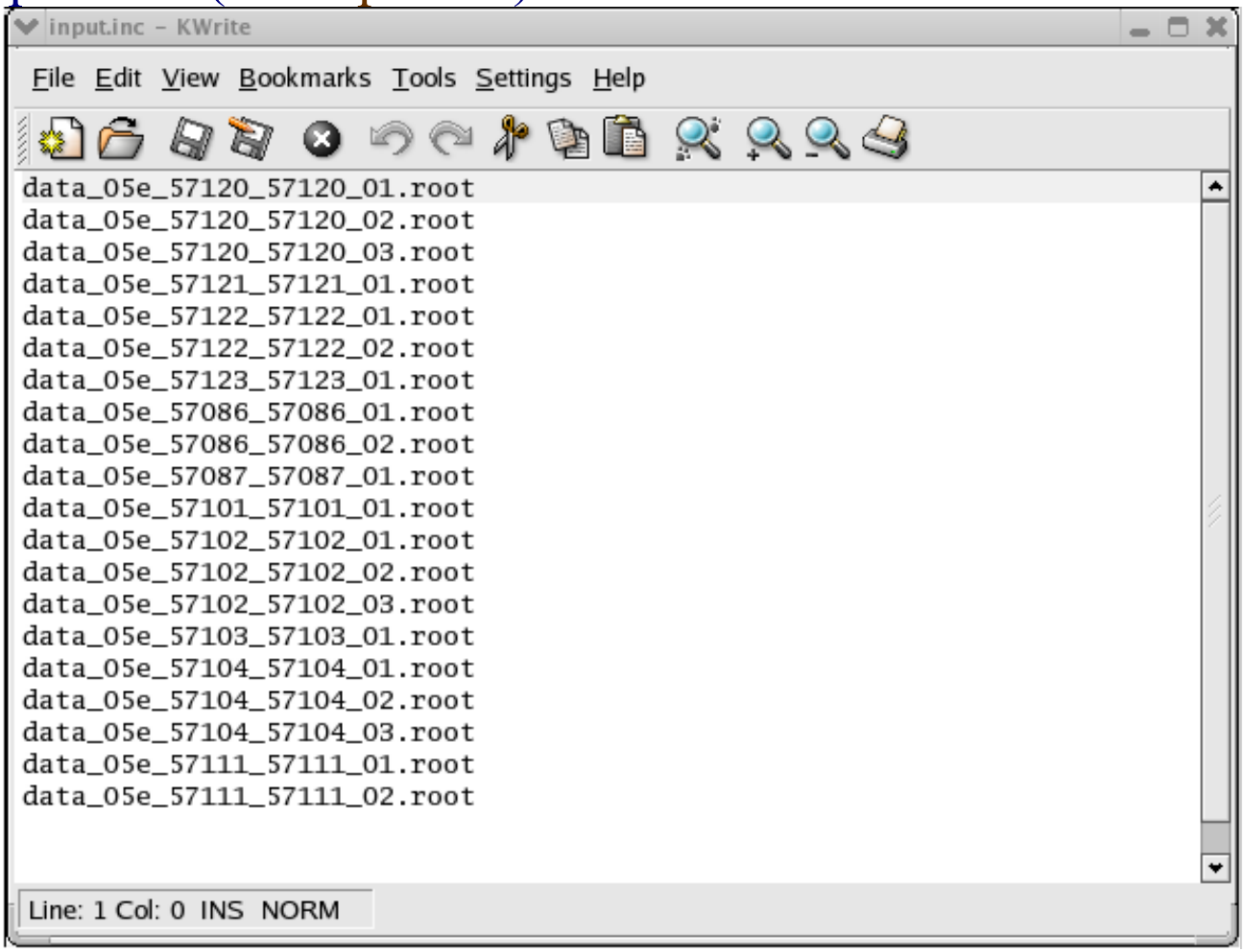


Figure 6. Selecting input files.

You can use here also "data_05e*.root"

4)[Optional] change pathes to input and output files and mode of generating.

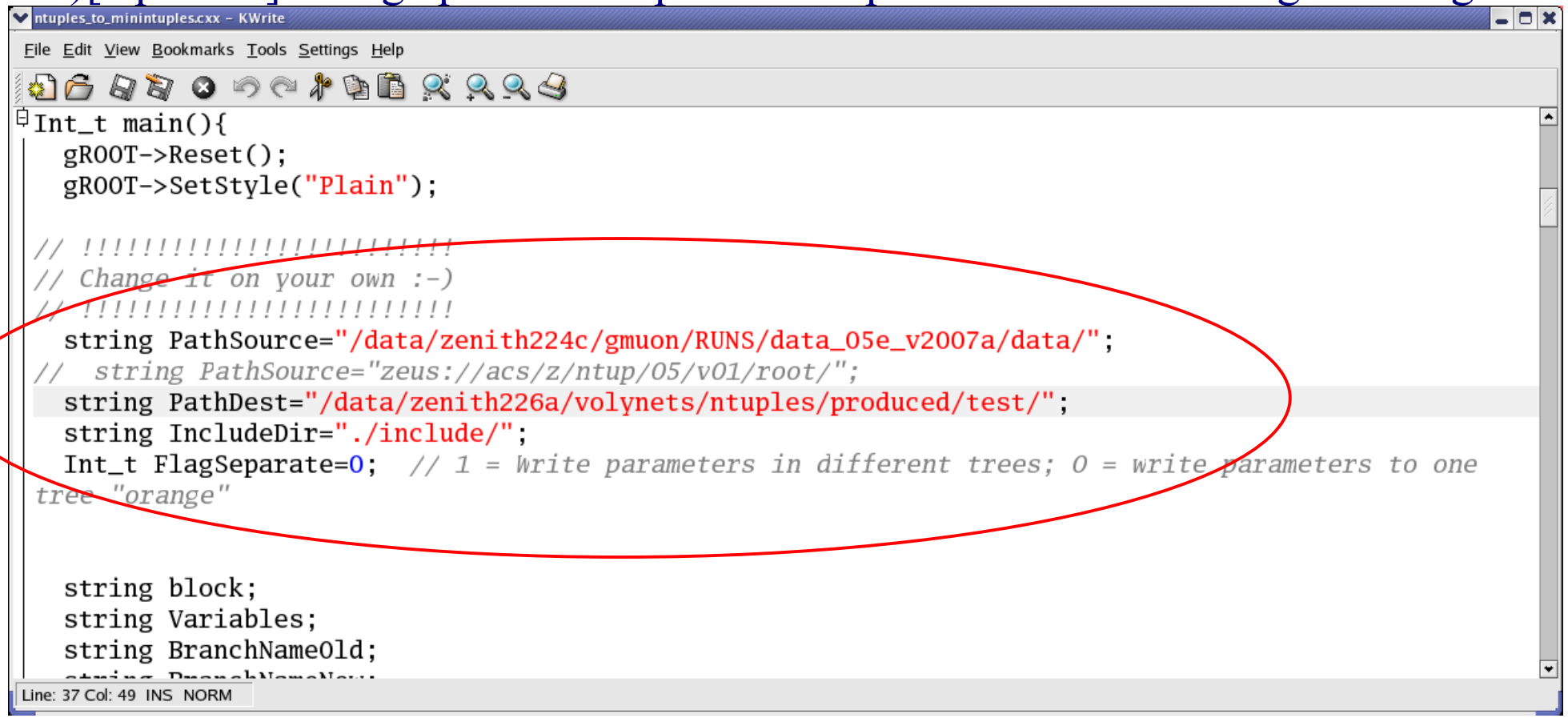


Figure 7. Selecting pathes and generating mode.

5)Run root

6)Execute script:

```
root [0]  .L ntuples_to_minintuples.cxx
root [1]  main()
```

7)That's all :-)

What is the difference of other projects (advantages):

- It doesn't need to be compiled etc. This is .cxx (root) script.
- Users can change what parameters of each block they need and don't need.

Plans of further developing:

- Fill other include files for another ORANGE blocks (gmuon, BAC, CAL,...), because now there are only some include files for demonstration.

You can download script and user's manual at

http://www-zeus.desy.de/~volynets/Mini_ntuples/

Examples of histograms using Mini_ntuple files:

From ntuple data:

From mini_ntuple data

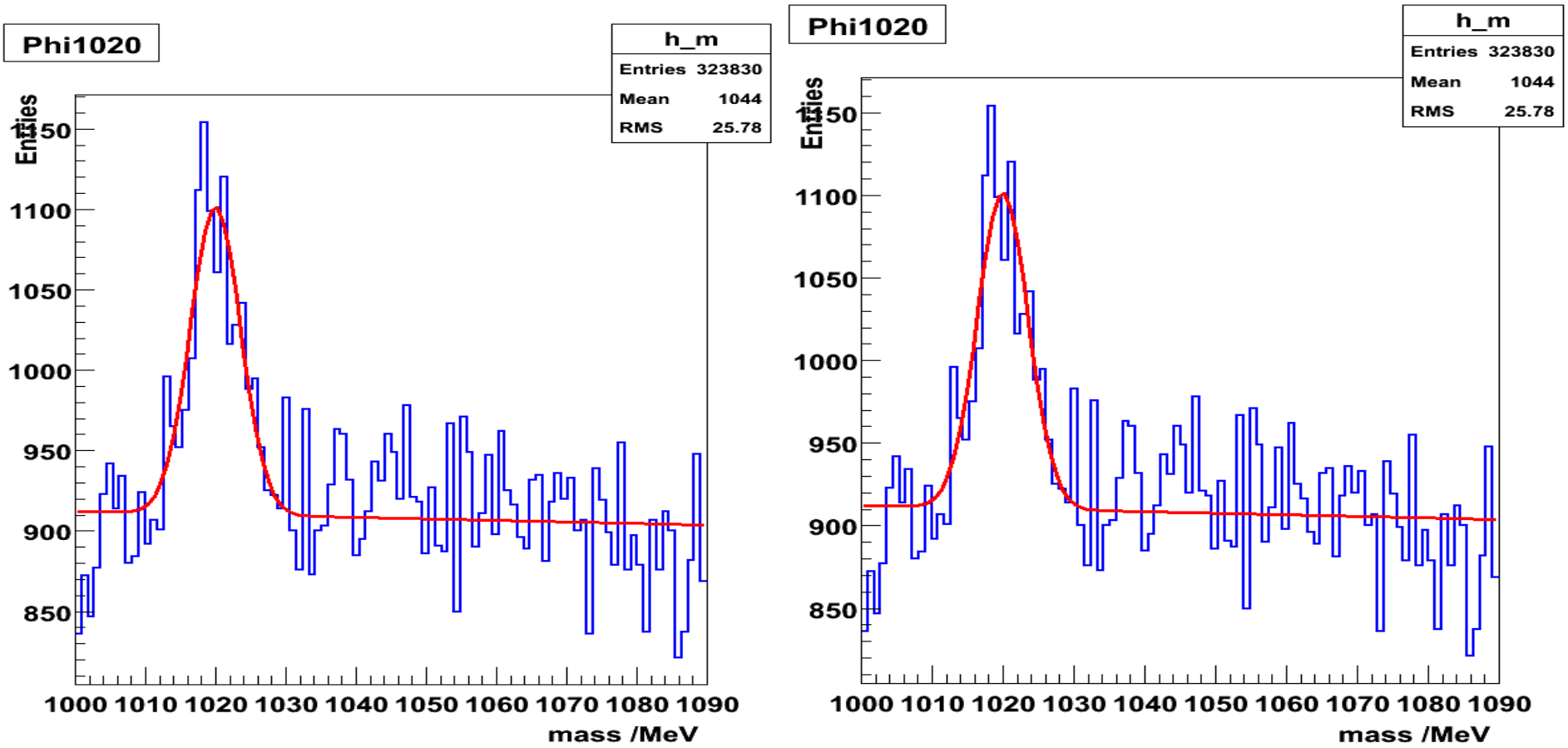


Figure 8. $\Phi(1020) \rightarrow K^{+}+K^{-}$.

Blocks used: Tracking, Trk_vtx, Trk_MVD, Trk_imp, Trk_qual, Vertex, Trk_vert.

From ntuple data:

From mini_ntuple data

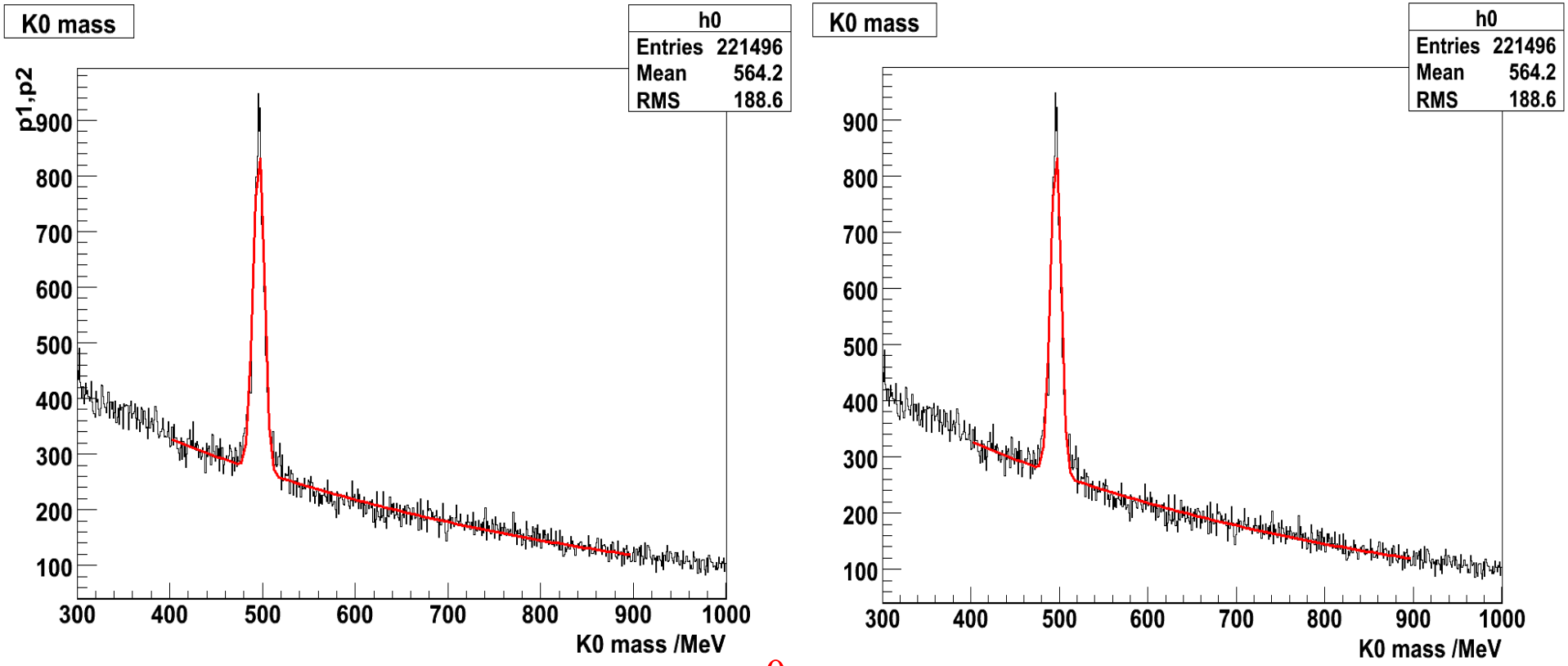
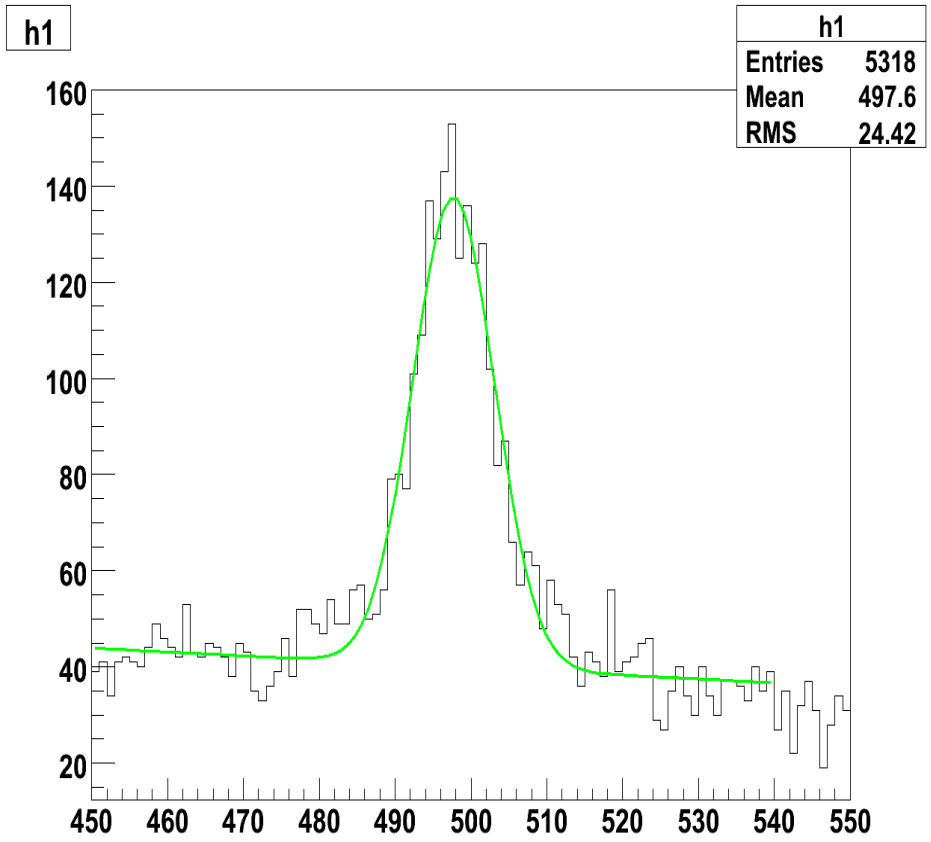


Figure 9. $K^0 \rightarrow \pi^+ + \pi^-$.

Blocks used: V0lite

From ntuple data:



From mini_ntuple data

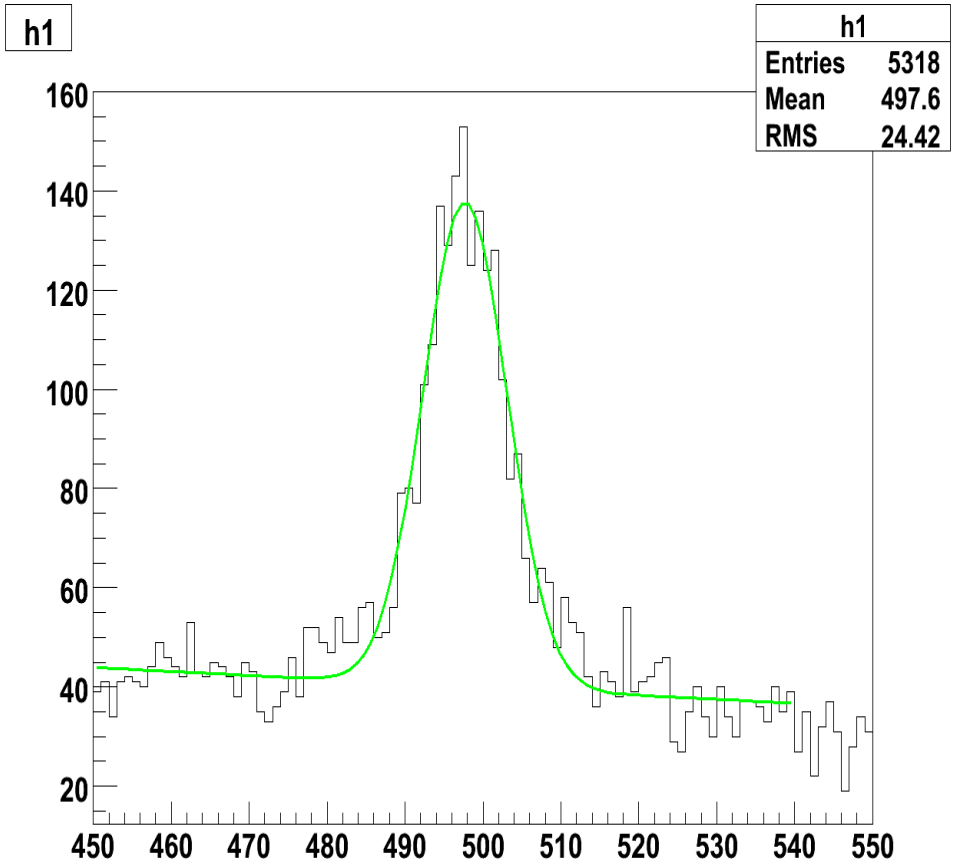


Figure 10. $K^0 \rightarrow \pi^+ + \pi^-$

Blocks used: Tracking, Trk_vtx, Trk_MVD, Trk_imp, Trk_qual, Vertex, Trk_vert, V0lite.

Conclusions: all blocks were written in a right way!!!!!!!!!!!!

2.Working with Gutcode – analysis framework.

2. Working with Gutcode – analysis framework.

What is Gutcode:

It is a analysis framework for different types of analysis (first of all muons) used by HFL group.

Task:

To make it working with dCache (reading data not only from NFS (zeniths) to save place on local disks, but if only the speed factor is not terribly big).

Problems:

It was written under root 5.08.00 and it didn't work under 5.14.00 (which is needed by ZACON – tools for reading files from dCache).

Possible ways of solution:

1) Include all analysis features to ZACON.

   Problems: It will need much time to rewrite ZACON and to do in such way means that every analysis tool should be included to ZACON.

2) Include ZACON tools to Gutcode.

   Problems: Almost the same: ZACON code should be included to every analysis tool.

3) Use "zeus://" protocol, provided by ZACON for common use.

YES!!!: It can be used in every analysis tool. You should add initialization of this protocol:

gSystem->Load("libdcap.so");

gSystem->Load("libzio.so");

before reading files and then you can use dCache as you use NFS (zenith):

fChain->Add("zeus://acs/z/ntup/05/v01/root/*.root");

For more information, ask the OFFLINE group :-)

So, for reading data from dCache with Gutcode, one should add files:

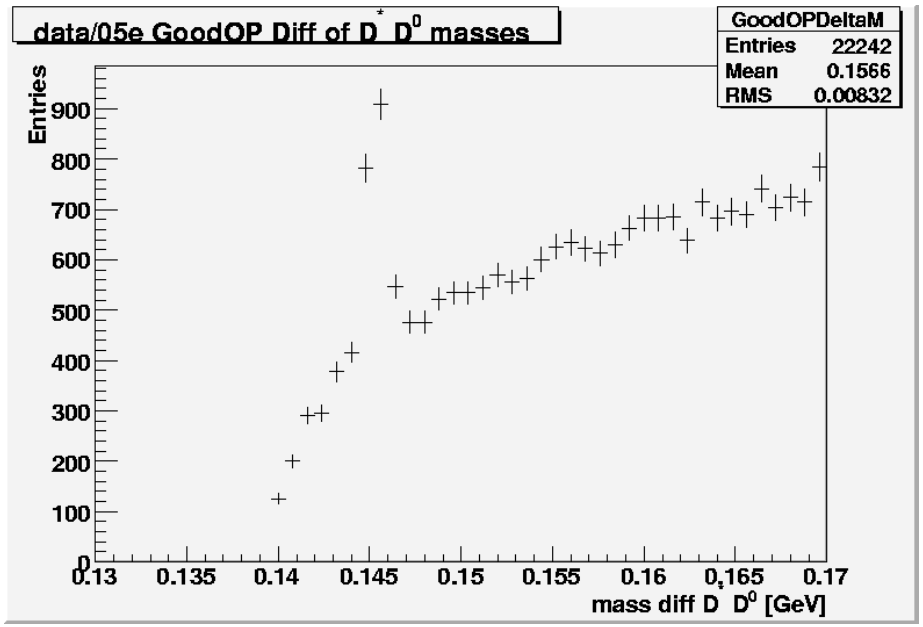
$GUTCODEDIR/analysis/dcache/ABTdata.txt,

where AB is the year of data, T is type of data (e/p - electron, positron). These files should contain full path to dCache data:

05edata.txt: zeus://acs/z/ntup/05/v01/root/data_05e_57014_57014_02.root
zeus://acs/z/ntup/05/v01/root/data_05e_57050_57050_01.root
zeus://acs/z/ntup/05/v01/root/data_05e_57051_57051_01.root

Examples of histograms produced with Gutcode using zenith and dCache are presented below:

From zenith:



From dCache

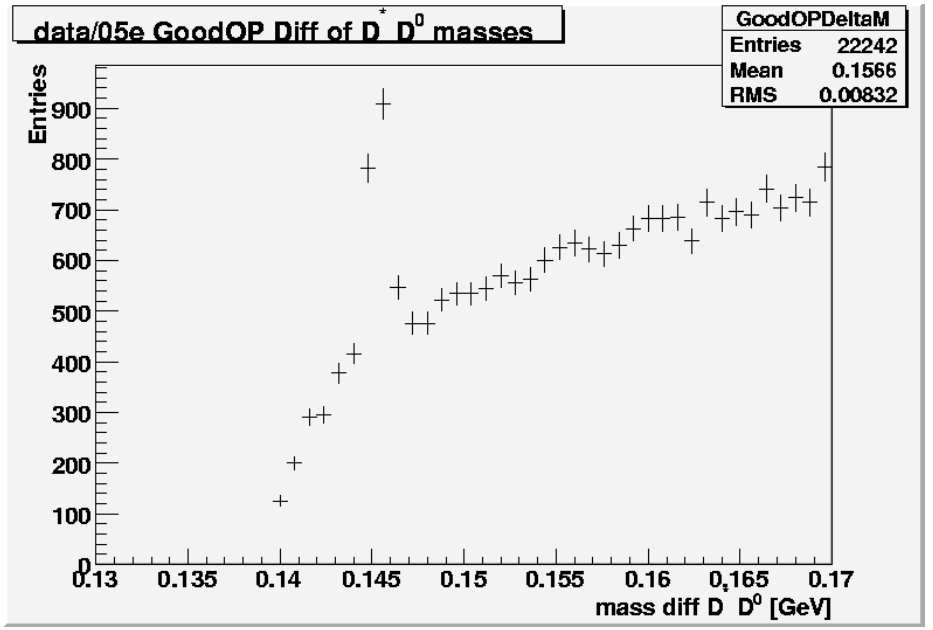
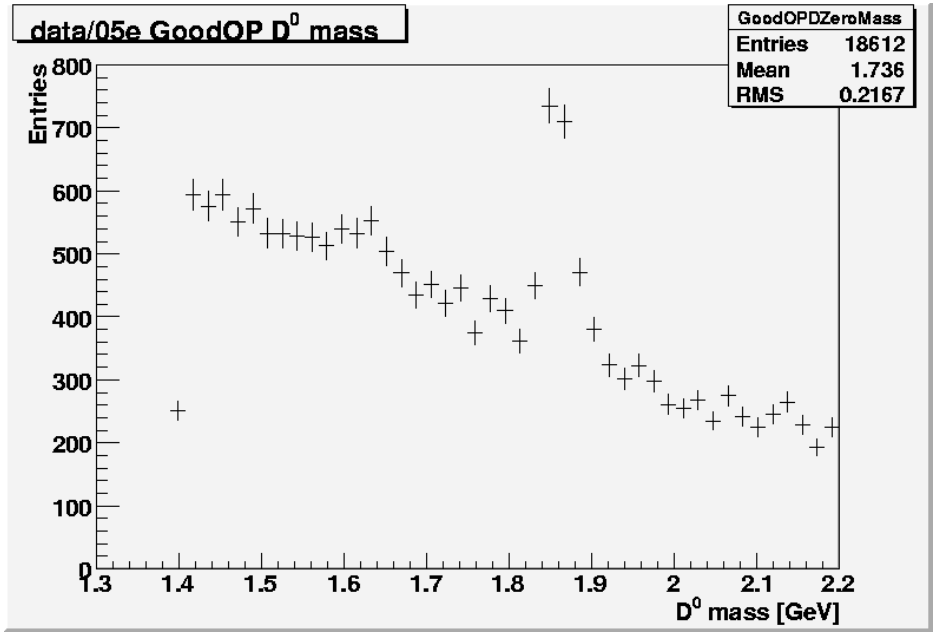


Figure 11. D* and D^0 mass difference.

Examples of histograms produced with Gutcode using zenith and dCache are presented below:

From zenith:



From dCache

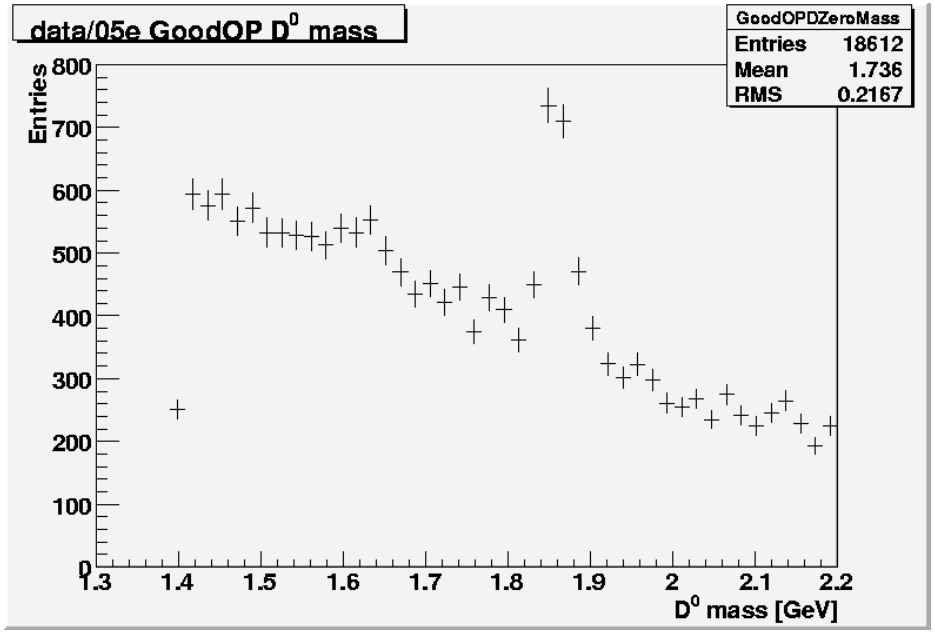


Figure 12. D^0 mass distribution.

Conclusions

1. There was a tool developed that can produce small mini_ntuples from Common ntuples for better perfomance. It allows to operate 20-50 parameters instead of 600. It was tested on several particles ($\Phi(1020)$, K^0) and different blocks (Tracking, Trk_vert, V0lite etc.). It worked 5 time faster compared with using Common ntuples. It is also compatible with dCache (using "zeus://" protocol).
2. Modifications for Gutcode analysis tool was applied in order to work with dCache. It was checked on several particles (D^*, D^0) and for different years (96 - 2000, 2005).

I am going to support my program according to the future modifications of data format etc. and Gutcode with further developing of Root, ZACON etc.

Plans

- To make my diploma here with your help;
- Later make my PhD here also with your help.

Aknowledgements:

I would like to thank:

- My supervisor Achim Geiser for coordinating me in different problems;
- Dmitriy Gladkov for his Gutcode setup;
- Igor Rubinskiy, Philipp Roloff, Benjamin Kahle, Ingrid-Maria Gregor for discussion different problems;
- Vladimir Aushev for many advices during my being here;
- Tobias Haas, Elisabetta Gallo and Joachim Meyer and all DESY and ZEUS collaboration for the opportunity to participate Summer Student program.

Plans

- To make my diploma here with your help;
- Later make my PhD here also with your help.

Aknowledgements:

I would like to thank:

- My supervisor Achim Geiser for coordinating me in different problems;
- Dmitriy Gladkov for his Gutcode setup;
- Igor Rubinskiy, Philipp Roloff, Benjamin Kahle, Ingrid-Maria Gregor for discussion different problems;
- Vladimir Aushev for many advices during my being here;
- Tobias Haas, Elisabetta Gallo and Joachim Meyer and all DESY and ZEUS collaboration for the opportunity to participate Summer Student program.

Thank you for your attention!!