Mini-Tutorial for AMBER software

11th May 2021

C. Quintans, with the help of many people, namely Sergei Gerassimov, Andrea Bressan, Vincent Andrieux, Carlos Azevedo, Yann Bedfer, Christian Dreisbach, Michela Chiosso, ...

These materials

If you have problems accessing gitlab (AMBER and/or COMPASS), you can also obtain the tar files from:

My cernbox public link
https://cernbox.cern.ch/index.php/s/5zwEWf6vuhtTyg1
(if it asks for password, it is “amber”)

Or a public directory of mine in lxplus:
> /afs/cern.ch/work/c/catarina/public/Tutorial/

Disclaimer

This is a sort of rehearsal, a preparation for a “real tutorial”, that we want to organize.

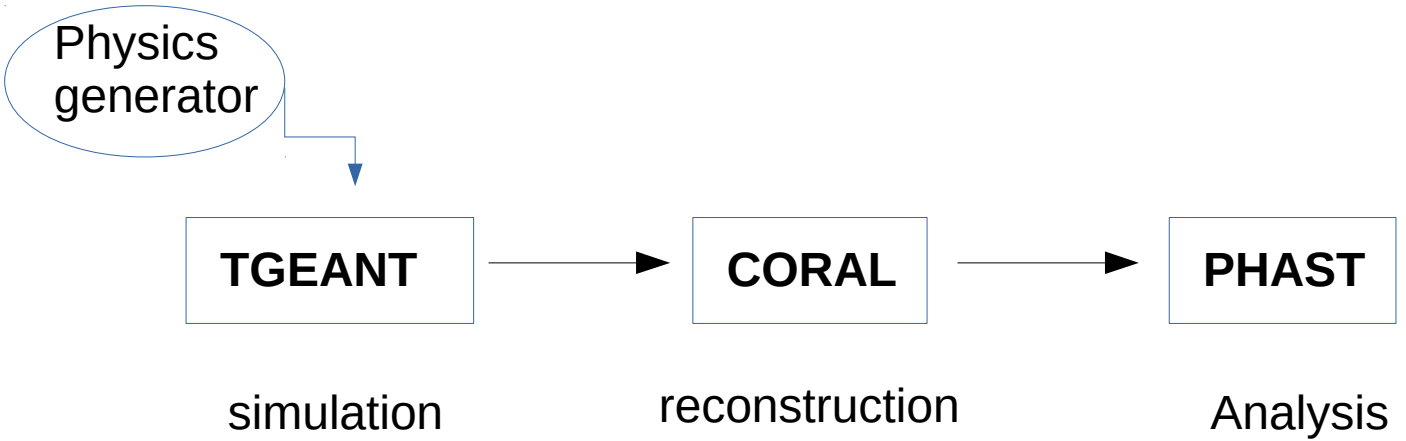
The goal is to have you started:

- Where to find the software;
- How to install it;
- How to test it;
- Where you can find more information;
- Whose experts can you contact for help;

Pre-requisites

- Know a bit about Monte-Carlo
- Being familiar with linux
- Have an account at lxplus

The MC chain



Physics generators: Pythia, LEPTO, HEPGen, or any other (but for latter, run in stand-alone)

TGEANT: GEANT4 based

Many dependencies on other packages (gcc, ROOT, Qt5, CLHEP, XercesC, cernlib, ...)

Documentation

TGEANT:
https://wwwcompass.cern.ch/compass/software/offline/TGeant/TGeantOldPage/

PHAST:
http://ges.home.cern.ch/ges/phast/index.html

Pythia 8:
http://home.thep.lu.se/~torbjorn/Pythia.html

GEANT4:
https://geant4.web.cern.ch/support/user_documentation

LHAPDF:
https://lhapdf.hepforge.org/

ROOT:
https://root.cern/manual/

How to start

Connect to lxplus.cern.ch using ssh and hability to export graphical windows:

> ssh -X -C [username]@lxplus.cern.ch

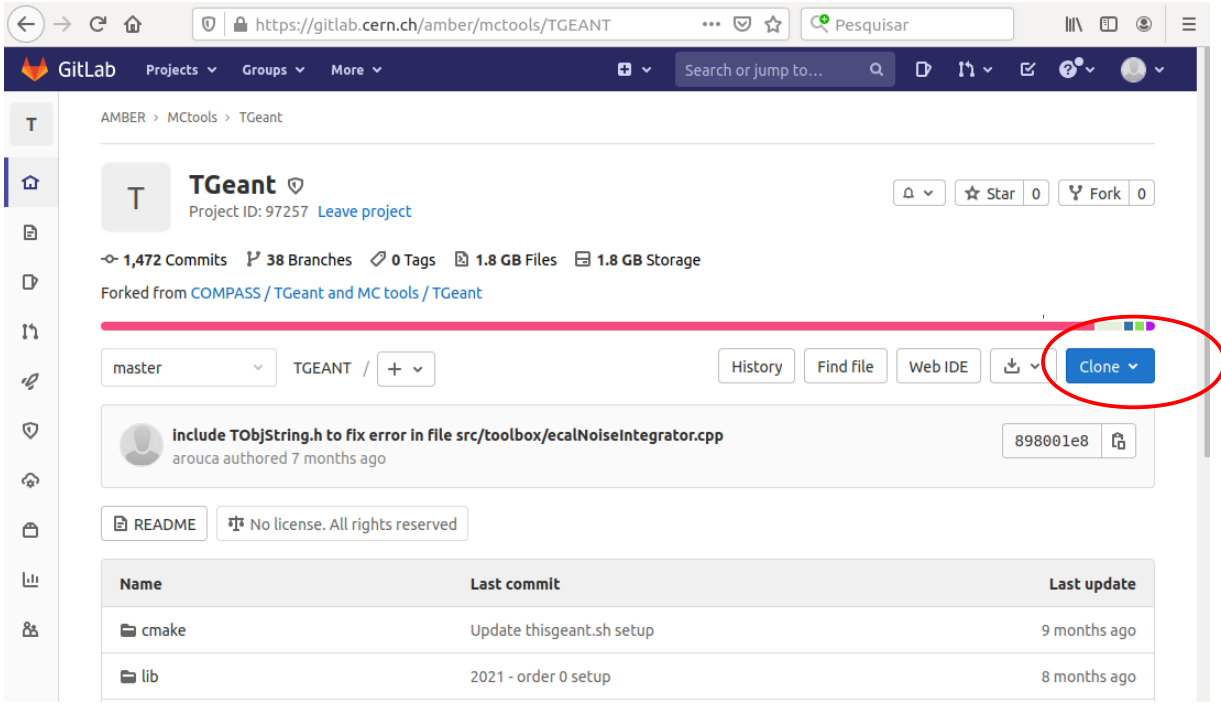
Make sure that you have enough space to install all the software.
For the simple exercises, count ~2G. You can use your work-space:

> /afs/cern.ch/work/[u]/[username]/
(if you do not have one already, create it from https://account.cern.ch/account
Resources and Services → List Services → AFS services → Settings → Create workspace here)

The preferred shell for these exercises is bash. If by default you have csh or tcsh, change it to bash when starting the exercise:
> bash
(if you already use bash, first remove your .bashrc file)

TGEANT for AMBER

If you have access to the AMBER gitlab:
https://gitlab.cern.ch/amber/mctools/TGEANT



- Select:

Clone with HTTPS

and copy URL.

- Now, in your terminal, do:

> git clone https://gitlab.cern.ch/amber/mctools/TGEANT.git

TGEANT for AMBER

If you do not have access to the AMBER gitlab, get the pre-prepared tar file:

```
> cp /afs/cern.ch/work/c/catarina/public/Tutorial/TGEANT-master.tar.gz .
> gunzip TGEANT-master.tar.gz
> tar -xvf TGEANT-master.tar
```

At this point, you have a folder TGEANT-master.

TGEANT has a lot of dependencies on other packages. These are software pre-installed at CERN. If at lxplus, you just have to give the paths to it, by running a script:

```
> cp /afs/cern.ch/work/c/catarina/public/Tutorial/myenv.sh .
> source myenv.sh
```

Compiling TGEANT

You are now ready to compile TGEANT, using an already prepared script and cmake:

```
>  cd TGEANT-master
> cp cmake/bootstrap.sh .
```

Edit this bootstrap.sh file with a text editor of your choice, and change
cmake → cmake3
such that the line reads:
cmake3 $REALPATH -DCMAKE_BUILD_TYPE=$BUILD_TYPE -DCMAKE_INSTALL_PREFIX=$REALPATH/install
and save it.

Now start the installation:

```
> ./bootstrap.sh 1
```

“building”, “generating”, “linking” and, once at 100%, “installing → ~5 min

Your TGEANT

Make an environment variable $TGEANT to point to your installation:

```
> export TGEANT=$PWD/install
> echo $TGEANT
```

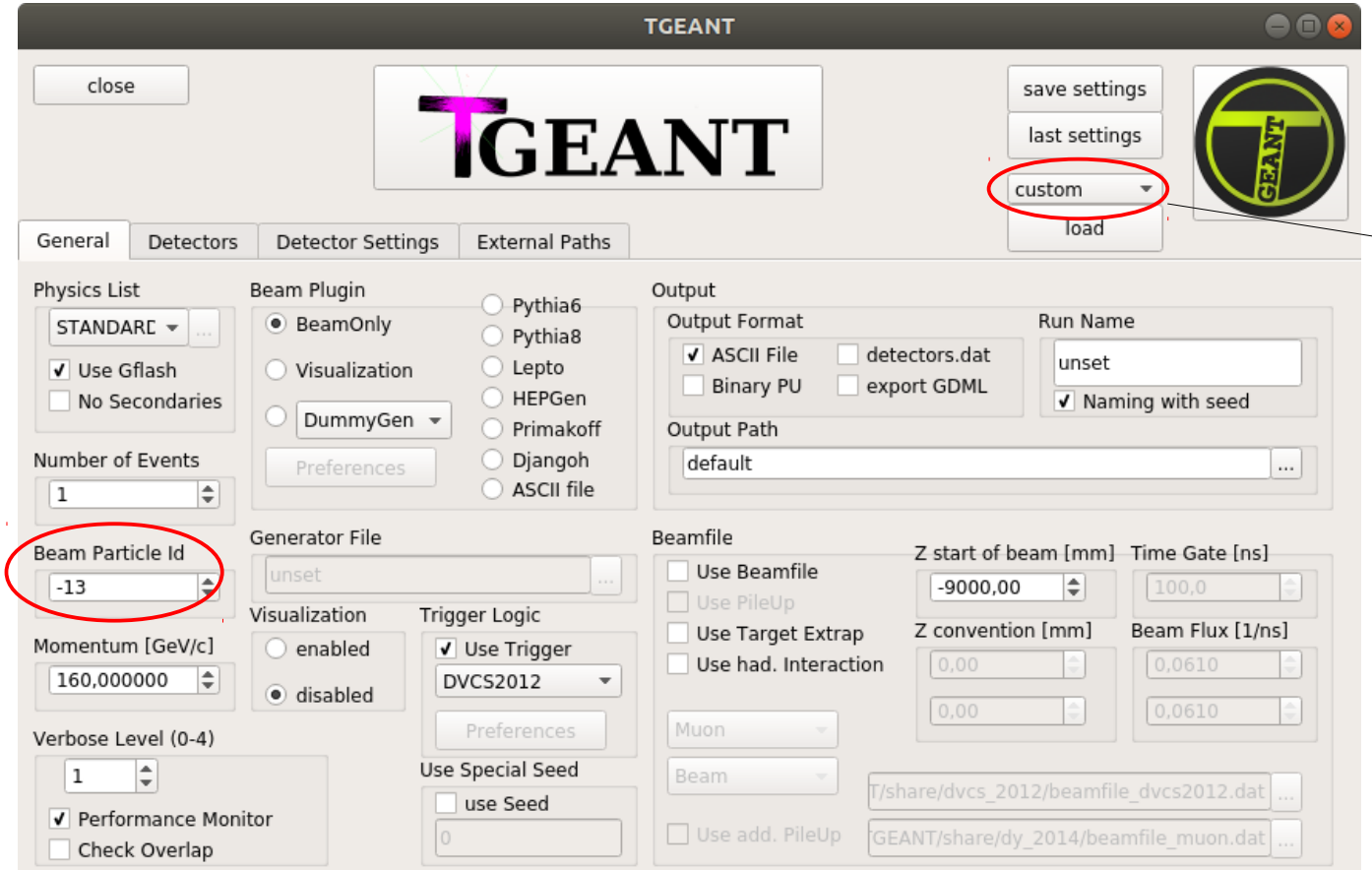
Make this permanent by putting this in your myenv.sh, one folder up (if doing that, just export TGEANT=[$PWD]/install).

Create a folder for your TGEANT exercises:

```
> mkdir my_tgeant
> cd my_tgeant
```

Testing TGEANT interactively

> $TGEANT/bin/Interface



Choose the setup you Want. For example: “2018-DY”
Then load (below)

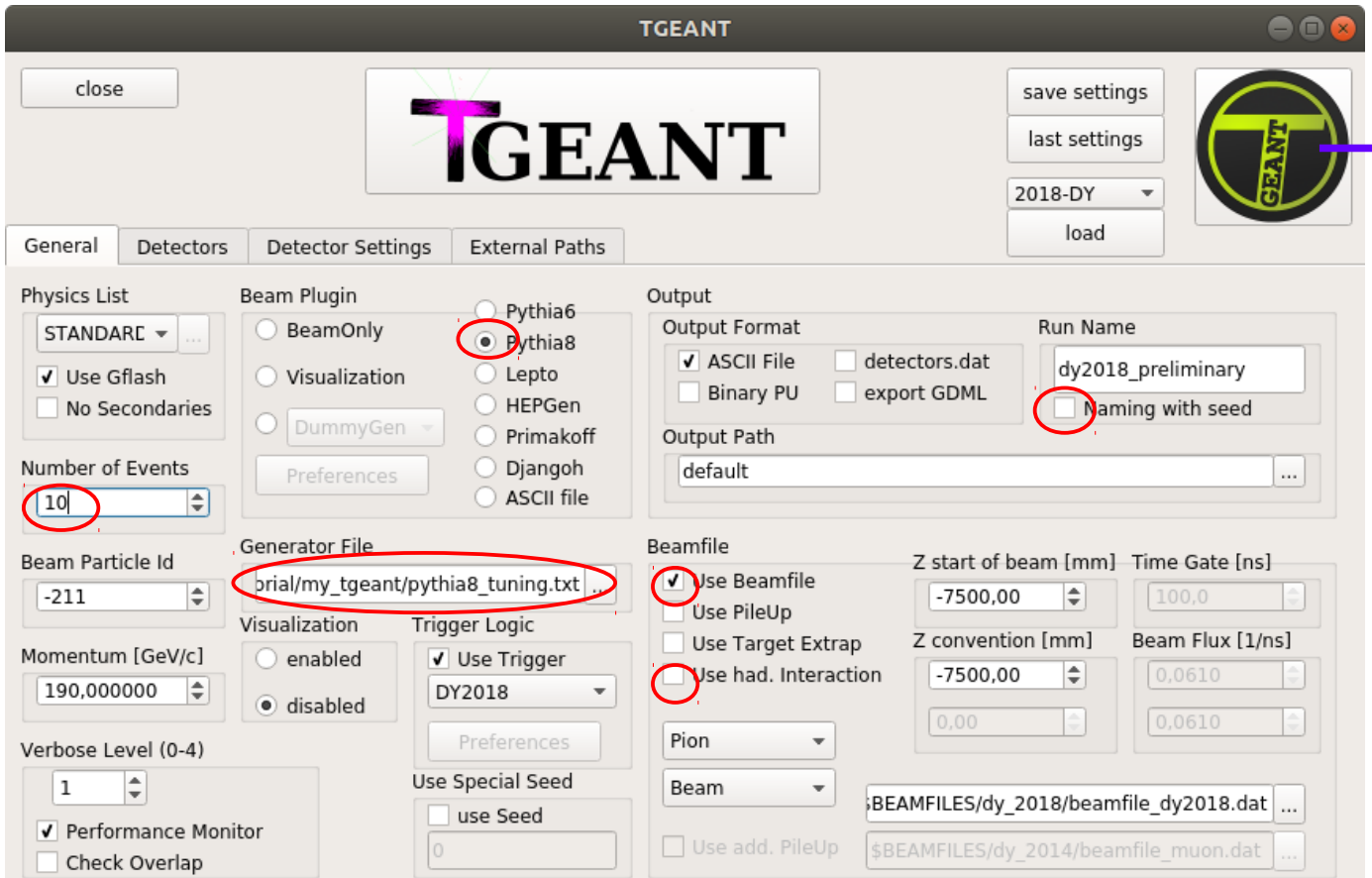
Beam particle ID refers to the PDG particle codes:

-211 = pi^- -13 = mu^+

211 = pi^+ 13 = mu^-

Testing TGEANT interactively

> $TGEANT/bin/Interface



Start button

In this example, I chose 2018-DY setup, and will use the pythia8 generator.
It requires a file with settings for Pythia:

> cp /afs/cern.ch/work/c/catarina/public/Tutorial/pythia8_tuning.txt .

Testing TGEANT interactively

Now you want to save these settings for later.

On top right, click “Save settings” and choose a file name indicative of the physics it is appropriate for. For example: mysettings_tgeant_DY2018.xml

This file will be saved in your my_tgeant folder, as a .xml file that you can edit and modify by hand.

Click the “Start button” to generate the events.

TGEANT and the physics generators

If you do not want to use the proposed physics generators but your own, you can use it stand-alone, write generated particles to an ascii file with format (per event):

[# particles]
[x_vertex] [y_vertex] [z_vertex]
[ID] [px] [py] [pz] → Position in cm. X-axis is along beamline
[ID] [px] [py] [pz] → ID is the particle GEANT code. Momentum in GeV/c
...

And as “Beam plugin” (generator input) you choose “Ascii file”.

TGEANT is very verbose, but not necessarily on the details you would like to have on what was generated. It is a good idea to have the generator prepared for stand-alone mode, and test its parameters in that way.

From TGEANT to ROOT output

After running interactively, you should have a file .tgeant in your my_tgeant folder. Here are your generated events. There is a TOOLBOX that allows to create histograms etc of what was simulated.

More practical: convert the output to a .root format, that you can inspect directly.

```
> $TGEANT/bin/TGEANT2ROOT  dy2018_preliminary_run001.tgeant  dy2018_preliminary_run001.root
> root dy2018_preliminary_run001.root
[root] new TBrowser
[root] .q
```

Some other inputs obtained from TGEANT

Now go back to the interactive session, and create 2 inputs for coral that will be needed later:

- .gdml file
- detector.dat file

For that, do all the same selections, but in the central right part, at “Output”,
un-tick "ASCII file" (that creates the normal .tgeant output)
and select both "detectors.dat" and "export GDML".

Run and keep the created files for later.

Detectors.dat can be edited normally and should be studied. Gdml also, but too long.

TGEANT in batch mode

> $TGEANT/bin/TGEANT mysettings_tgeant_DY2018.xml > my_tgeant_test.log &

This will process in background the number of events you asked for, writing to a log file the output that would otherwise appear in your terminal.

Done in this way, you are still using the local login machine → not adequate for longer jobs!!! (specially if you are at lxplus)

instead, use HTCondor (at CERN), Frontera (Texas, USA), or the farm in your home institute.

CORAL

If you have access to the COMPASS gitlab:

https://gitlab.cern.ch/compass/coral

As you did for TGEANT, copy the https URL for cloning, and in your terminal, do:

```
> git clone https://gitlab.cern.ch/compass/coral.git
```

If you do not have access to the COMPASS gitlab, get the pre-prepared tar file:

```
> cp /afs/cern.ch/work/c/catarina/public/Tutorial/coral-master.tar.gz .
> gunzip coral-master.tar.gz
> tar -xvf coral-master.tar
```

Compiling CORAL

```
> cd coral
> cp cmake/bootstrap.sh .
```

Edit the bootstrap.sh file with any editor of your choice, change the call to cmake:
cmake -> cmake3
and add an option -DALL=ON

such that the line will read like this:

```
cmake3 $REALPATH -DALL=ON -DCMAKE_BUILD_TYPE=$BUILD_TYPE -DCMAKE_INSTALL_PREFIX=$REALPATH/install
```

and save it. Now start the installation:

```
> ./bootstrap.sh 1
```

This procedure takes 5-10 minutes. If there was no fatal error, you should have it installed. But we do not run it from here (not practical).

Some other input obtained from CORAL

```
> cd install
> source setup.sh
```

This sets you an environment variable $CORAL that points to your coral installation. You can make this permanent by including it in your myenv.sh file.

Now we create a dictionary file, like a roadmap for typical tracks, that helps in the reconstruction: the **dico** file. We need a specific executable for that

```
> cd $CORAL/../src/track/trafdic/makeDico
```

There are here 2 option files “muon” and “hadron”, depending on type of setup (refers to beam). For DY, better copy the “hadron” one and modify:

```
> cp makeDico.hadron.opt makeDico.DY.opt
```

Dico’ s option file

Edit this file, and change "detector table" to point to the detectors.dat you created in your my_tgeant folder.

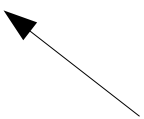
Check that "TraF iCut [15]" is the correct beam charge for your case.

Comment line SOL_field, if AMBER DY or Hadron (here we do not do it because tgeant used 2018-DY setup, that has target dipole)

TraF ReMode [20] 2 // 1: Use material map; 2: Material map + dE/dX ---> DY
define zone -700 3500 before M1 //DY, or change these values (Z after target (in mm) up to SM1)

Plus add these lines if DY, before end, and replacing first to point to the .gdml file in my_tgeant :

// - ROOTGeometry files, or in this case .gdml file we created in tgeant:
CsGDMLGeometry file /afs/cern.ch/work/c/catarina/public/Tutorial/my_tgeant/dy2018_preliminary_run001.gdml
CsROOTGeometry massDefault .105658367
CsROOTGeometry simpleELoss 0
CsROOTGeometry ELossStraggling 0



Replace by yours

Creating the dico file

> $CORAL/../build/bin/makeDico makeDico.DY.opt

It takes one minute, and after 233200 events it creates you a large dicofit.out file.

You keep it for later.

PHAST for CORAL

Now you need to get the latest version of phast. From COMPASS gitlab:

https://gitlab.cern.ch/compass/phast

As you did for TGEANT and CORAL, copy the https URL for cloning, and in your terminal, do:

```
> git clone https://gitlab.cern.ch/compass/phast.git
```

If you do not have access to COMPASS gitlab, get the pre-prepared tar file:

```
> cp /afs/cern.ch/work/c/catarina/public/Tutorial/Phast.tar.gz.8.019 .
> tar xzvf Phast.tar.gz.8.019
```

Either way, you should get a folder “phast”, and inside it a sub-folder “coral”. We compile phast first. We will activate the flag to have also GUI accessible.

Compile phast and make coral.exe

```
> cd phast
> ln -s Makefile.lxplus_centos7 Makefile
> unset LD_PRELOAD
> make -j8 WITH_GRAPH=1
```

This step takes ~5 minutes. If it worked, you should end-up with a phast executable.

Now we build coral-inside-phast:

```
> export PHAST=$PWD
> cd coral
> unset LD_PRELOAD
> make -j8 WITH_GRAPH=1
```

This takes ~5 minutes. After that, if it did not fail with fatal error, you should have a file coral.exe

Testing CORAL interactively

Now, side by side with your my_tgeant, create a new folder my_coral:

```
> cd ../..
> mkdir my_coral
> cd my_coral
> ls $CORAL/../src/user/
```

This last command shows you the default option files for running coral, depending on the physics you are interested in:

- Some are adequate for real data processing (trafdic.[year].[physics_case].opt)
- Some are good for MC (trafdic.[mc OR tg].[year].[physics_case].opt)

For the DY-2018 setup:

```
> cp $CORAL/../src/user/trafdic.tg.2018.opt .
```

CORAL options file

Edit your new trafdic.tg.2018.opt, and change it:

CsTGEANTFile file [absolute path to your .tgeant file]
detector table [absolute path to your detectors.dat file]
TraF Dicofit [absolute path to your dico file]
CsGDMLGeometry file [absolute path to your gdml file]

TraF Graph [0] 1 --> uncomment this line and put to 1, to have GUI

Running CORAL interactively

At this point, if you built from bootstrap.sh and you already have the CORAL executable, a little trick to run coral:

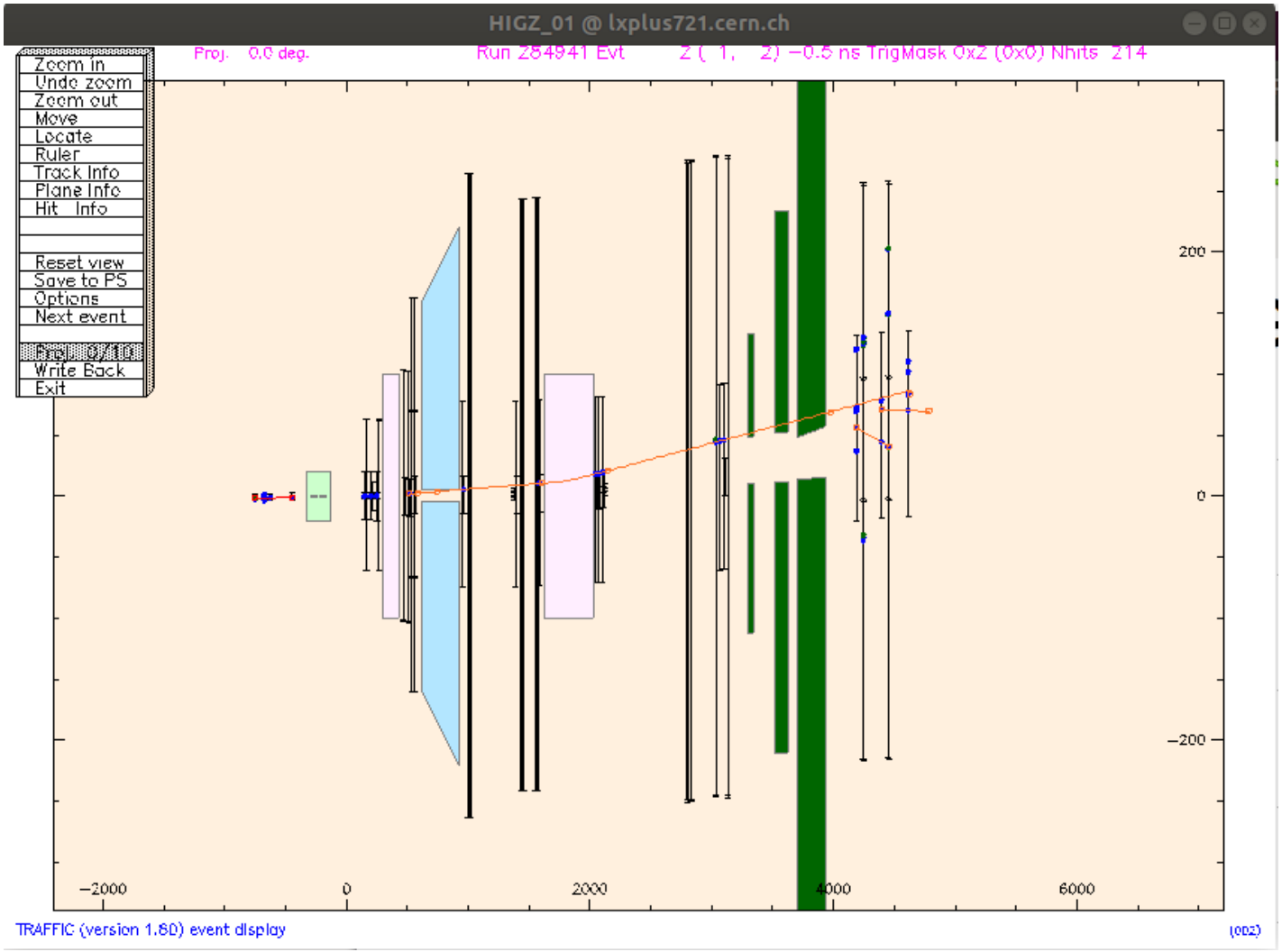
```
> export CORAL=$CORAL/..
> export LD_LIBRARY_PATH=$PHAST/lib:$PHAST/user:$LD_LIBRARY_PATH
> export PATH=$PHAST/coral:$PATH
```

This can be put to your myenv.sh

```
> $PHAST/coral/coral.exe trafdic.tg.2018.dy.opt
```

It takes time. It is very verbose. You will see errors passing, but if they are not fatal, do not worry about them at this point. It will ask you for the size of GUI you want. Choose “3” for small, “2” for larger.

CORAL GUI



It shows you reconstructed tracks and hits in the event.

From the “options”, can have also vertexing, passive materials drawn, or mag. fields Drawn.

By default “top view”. Use “proj” To see from other angles.

→”Next event”

CORAL logs

When CORAL finishes, it sent to the terminal window a sort of log, a summary of the reconstruction:

```
Time in Track Fit     =  0.0250  sec/ev
Time in End-of-Event  =  0.0180  sec/ev
Total time in TRAFFIC =  0.1130  sec/ev
Total number of TRAFFIC beam  tracks / ev          = 0.5000
Total number of TRAFFIC event tracks / ev          = 2.0000
Number of TRAFFIC beam  tracks with momentum / ev  = 0.5000
Number of TRAFFIC event tracks with momentum / ev  = 1.3000
Overal track finding efficiency in 2-50 GeV range = 75.0000 %
```

```
/------------/ Vertex Kalman Filter statistics \--------------\
| The number of events (def)                          10    100% |
| Was called                                           1   10.0% |
| Primary was found                                    1   10.0% |
| Primary with mu'                                     0    0.0% |
| Primary with BMS                                     0    0.0% |
| Primary with mu' and BMS                             0    0.0% |
| Primary with mu' and BMS OK                          0    0.0% |
|-----------------------------------------------------------------|
| Less than 2 tracks left in vertex                    0    0.0% |
| Rejection by Chi2/ndf cut                            0    0.0% |
|-----------------------------------------------------------------|
| # of events with #prim vert > 1                      0    0.0% |
| Events which profit from rescu procedure             0    0.0% |
|-----------------------------------------------------------------|
| Average # of tracks in primary vertex              2.0          |
| Number of secondary vertices                         0    0.0% |
|-----------------------------------------------------------------|
| Total time spent per event                       0.011    100% |
\-----------\_____________________________________/-----------/
```

CORAL in batch mode

Edit your trafdic.tg.2018.opt, in order not to call graph mode:

TraF Graph [0] 0 → graphics mode switched off

If you are going to run CORAL in background, but still in the login lxplus machine, change the number of events to process to some small number:

events to read 500

Now you can run it:

> $PHAST/coral/coral.exe trafdic.tg.2018.opt > meu_coral.log &

Besides the log, you will obtain 2 outputs:

- trafdic.mc.root → Control histograms
- mDST.root → Your miniDST, contains the reconstructed events (tracks, vertices)

PHAST

PHAST and the analysis of the MC events (from mDST) is your next step.

> cd ../user

In this directory you will find practical examples on how to do that. Create a UserEvent with not-yet-used number. Use the methods described here:

http://ges.web.cern.ch/ges/phast/doxygen-html/annotated.html

And learn with Sergei all the secrets from PHAST

How to get help

- Sergei Gerassimov (PHAST)
- Yann Bedfer (CORAL)
- Vincent Andrieux (CORAL)
- Catarina Quintans (CORAL)
- Christian Dreisbach (TGEANT)
- Andrea Bressan (TGEANT)
- Carlos Azevedo (GEANT4)
- … and many more

Or use the new mailing list:

amber-software@cern.ch

For experts and experts-to-be
Known issues and to-do list

- Add to AMBER gitlab CORAL and PHAST packages
- For AMBER TGEANT master, merge the branch used for DY (AmberCarbonTarget)
- Apply fix to TGEANT master affecting DY (“use hadron interactions” bug)
- Install LHAPDF 6.2.3 (or higher) as common software package
- Install Pythia 8.2.40 (or higher) as common software package
- Contact Pythia8 authors about needed fix for pion PDFs (JAM/xFitter issue)

- Create an AMBER group computing account in lxplus
- ...