D3PD Analysis Framework

0) REQUIREMENTS

* Installed ROOT analysis Framework.
* Installed g++ compiler.
* Installed GIT revision control system. (Optional)
* A working text editor (preferably one that does syntax highlighting).

1) QUICK INSTALLATION

on a linux or Unix box, prepare a directory structure

```
> cd $HOME
> mkdir -f project/dbx_work
> cd project/dbx_work
```

get it from GIT sw repository

```
>git clone ~/Dropbox/dbxa.git
>mv dbxa analyses
>cd analyses/toplu
>../getCorrectionFiles.sh
>../prep_after_getfiles.sh
```

IMPORTANT: Please make sure that you run the scripts: getCorrec. and prep_after. from TOPLU directory. Otherwise your links will be broken.

```
---ATTENTION----
before running remember to set your environment (this is automatic for grid runs)
> cd toplu
on linux:
> export LD_LIBRARY_PATH=$PWD:${LD_LIBRARY_PATH}
on mac:
> export DYLD_LIBRARY_PATH=$PWD:${DYLD_LIBRARY_PATH}
----------------
```

```
########################################################################
-----old and not used------
########################################################################
*********This is for rel 16*******
```

------to prepare to run in your laptop----- (this is automatic for grid

```
runs)
>tar xzf BAT-0.4.2.tar.gz
>cp analysis_core/prepbat.sh BAT/
>cd BAT
>./prepbat.sh
>cd ../KLFitter
>make
>cd ../../toplu
To build the default binary
>make
>ln -s KLFitter/libKLFitter.so .    (On macosx, .so is .dylib)
>cp BAT/installdir/lib/libBAT.so.3 .
>ln -s ../Uncertainties/JetUncertainties .

**********************************************************
Also compile the pcl code to use the fast power constrained limit
> cd ../pcl; make; cd -

>To build the binary for release 15:
> make rel15    [ not used anymore !!! ].
To build the binary for release 16:
> make rel16    [ this is the default ].
To build the binary for release 17:
> make rel17

##########################################################################
```

2) INTRODUCTION

This code provides a complete analysis over a set of D3DPs either in the data folder of your system or in the grid using dq2 tools. It is written in C++ with an emphasis on easy generalizability. It is also aimed to extend the analysis framework by adding useful classes.

2.1) HOW TO RUN THINGS

A) How to run it interactively

```
> cd toplu
> ./root_analysisd3pd  test.root -FF 3 -N4 2 ....
```

means
 FF analysis runs in 3 copies, copies are defined in FF_1-card.txt FF_2-card.txt   FF_3-card.txt .
 N4 analysis runs in 2 copies, copies are defined in N4_1-card.txt N4_2-card.txt .

A1) other command line parameters (1:on 0:off, default off)

```
> ./root_analysisd3pd  test.root -Q 1     : turn on QCD estimation while running on data (automatic on GRID)
> ./root_analysisd3pd  test.root -S 1     : turn on systematics estimation on MC
> ./root_analysisd3pd  test.root -HF 1    : turn on Heavy Flavour treatment on Wjets sample (automatic on GRID)
> ./root_analysisd3pd  test.root -D 1     : turn on dump analysis.
```

B) how to run on grid:

```
> cd toplu
> make grid
> scp 4grid.tgz lxplus:
```

(example in ~unel/public/4grid)

setup .bash_profile:
 you should add following 2 lines in .bash_profile file at your lxplus account (exmaple: /afs/cern.ch/user/u/unel/).

```
  export AtlasSetup=/afs/cern.ch/atlas/software/dist/AtlasSetup
  alias asetup='source $AtlasSetup/scripts/asetup.sh'
```

setup the grid (example in ~unel/public/readme)
edit the file ./rungrid.sh to select the user parameters, these are:

```
 ov=146        #your job version
 un="unel"     #your grid nickname
 aver=17.0.5   # athena version to be setup
 dosyst=0      # 0:dont do systematics, 1:run with the systematics
analizler="'-D 1'"   # "-FF 1" OR "-D 1"  # D is represent dump analysis and FF is represent heavy quark analysis. '-D 1':run with dump analysis. '-D 0': run without dump analysis.
```

(Remember you should submit each RunType with the same jobversion number)
edit the file submitall.sh to select data or MC files or electrons vs muons then

```
 ./submitall &
```

wait for jobs to finish. I get emails for each finished job.
Also it is possible to use rungrid.sh script seperately for each run numbers or runType.

```
./rungrid.sh [runType or runnumber] [channel] (example: nohup ./rungrid.sh siu4b &)
```

B1) Nomencalature
a job: the smallest unit which we run, usually d3pds of 5k events.
a runnumber: run number if for a MC or Data set. consists of many job files. ttbar has ~1400 job files.
a dataset: dbx definition of a MC or Data set, e.g. single top MC files are singleT. singleT has many run numbers: 108340, 108341 108342 108343 108344 108345 108346 .

C) how to get results from the grid:
These files will be HUGE. I usually work in /tmp/unel/ .

```
cd /tmp/unel
ln -s ~/public/4grid/getres.sh .
ln -s ~/public/4grid/runlist.txt .
ln -s ~/public/4grid/checkjobs.sh .
```

update the local repository before you work:

```
pbook -c sync\(\)
```

Then edit the file getres.sh (example in ~unel/public/4grid/getres.sh) to set your username and your job version number (e.g rv=65). Usage is

```
./getres.sh dataset_name action
```

where action is one of these:

```
add check checkdisp checkget checkmerge checkresub get remove
```

Main usage examples are below:

```
# to check the status of the jobs:
./getres.sh singleT check
# the output will be something like this:
we have 0 root files for run 108340  ;    Expecting 30 jobs.

# if you want more details on the jobs status:
./getres.sh singleT checkdisp
# the output will be something like this:
we have 25 root files for run 108342  ;    Expecting 30 jobs. show command
received for JID 10775 status : running finished : 30 running : 3

# to resubmit failed jobs
./getres.sh Wjets checkresub

# to get from the grid, all the root and lhco files for all w+jet samples:
./getres.sh Wjets get

# to check if the get command finished or not (lcg-cp is SLOW!!!) AND will
merge automatically
./getres.sh singleT checkget
# the output will be like
./getres.sh singleT checkget
get the number from : singleT
lcg-cp for  108340  is  Finished
we have 30 root files for run 108340  ;    Expecting 30 jobs.
ready, will add automatically.

# to add all the root files _manually_ for all w+jet dataset and to make
run_#.root files
./getres.sh Wjets add

# to merge all sub-jobs under a single root file IF all jobs finished
```

correctly we do this only for DATA.

```
./getres.sh dataPeriodB_muons checkmerge
```

+++you can also use a run number instead of the dataset name to handle 1 run only
+++e.g. ./getres.sh 108342 checkdisp

***if you can not remember dataset_name or the action command
***just put some nonsense text like abc, the program will remind you
***the list of valid names and commands.

if you really must look under the hood, the is also this script

```
checkjobs.sh -[command] [run number] [analysis number]
```

commands are:

```
-k : kill  ***** you use this very very cautiously *****
-r : retry
-d : do nothing
-u : update local repository
-s : show
-f : show and retry
```

=====> an example case:

```
./getres.sh singleT check
get the number from : singleT
-rw-r--r-- 1 unel zp 310K Dec  9 15:41 108340.root
-rw-r--r-- 1 unel zp 453K Dec  9 15:41 108341.root
we have 27 root files for run 108342  ;    Expecting 30 jobs.
-rw-r--r-- 1 unel zp 317K Dec  9 15:44 108343.root
-rw-r--r-- 1 unel zp 475K Dec  9 15:44 108344.root
we have 7 root files for run 108345  ;    Expecting 30 jobs.
-rw-r--r-- 1 unel zp 899K Dec  9 15:45 108346.root
```

When all get and add commands finish on all datasets,
tar and gz all the #.root files to jobnumber.tgz (e.g. 165.tgz)
download it to your analysis machine (e.g. your laptop)

C1) How to get the luminosity in data files

- setup ATLAS release
- get data analysis results from grid and add
- run the script calclumi.sh as below and look at the output

```
./calclumi.sh dataPeriodB.root dataPeriodD.root ....
```

D) How to setup the analysis part

edit ANA_DEFS file. (ALL? analysis parameters are there)
put the MC files into the sub-directory with the same number as the job version
put the data file in the main directory ==> the file you produce when you analyse the LHC data

```
root -q -x QCDout.C
./makesoftlinks
```

E) you are ready to play
(for the first time users, execute the following 2 commands in toplu directory

```
touch Limit.txt
ln -s Limit.txt OutputLimit.txt
```

)

```
root -x compare.C
./getlimits.sh
```

E1) Which Limit Setter to use?
RLimit and RSLimit are available with CLs method.
To select one or the other edit the file analysis_core/CalculateLimits.C to find the variable below:

```
int Limitter=0; // 0: TLimit, 1:RSLimit 2:fastPCL, 3:mclimit
```

Before use these things, make sure that your system has correct setup to use them e.g:
-if roofit is installed or not, to enable this package ./configure --enable-roofit
-go to mclimit directory and compile it etc.

Dumping d3pds

To be able to analyze simple versions of the d3pds, dump them first and then analyze the output file.

```
--> ./root_analysisd3pd NTUP_TOPEL.602075._000009.root -D 1
```

Output file of this command is "lvl0.root" by default.
To analyze that file in your machine by using prooflite, run this command;

```
--> root.exe -x runSelector.C
```

If you do not want to use all of the cores that your machine has, change runSelector.C file according to your wish with the help of this command

```
-->proof->SetParallel(n); n=number of cores you want to use
```

FILES INCLUDED IN THIS DISTRIBUTION

Description of Included files:

- LeptonicWReconstructor
  to make a W out of a lepton & MET info
- PrintEfficiencies
  to print the cut efficiencies on the screen
- ReadCard
  to read in cut values into an analysis
- compare.H
  to plot any relevant histogram
- doFinalFit
  various plots and significance extractions
- draw
  to prepare final plots
- TwoVarSample
  to keep track of means, variances and correlations of two variables
- WIndices
  to keep the indices of the W

Description of Analysis files:

- dbx_particle
  a particle has 4vector +charge +flavor +pt/et_cone
- dbx_a
  generic class for an analysis
- ff_a
  derived class inheriting from dbx_a
- ntrad3pd
  class to access data or MC files & do multiple analyses
- root_analysisd3pd
  function to drive an analysis
- ExclusionPlot.C
  Plooter for Exclusion Graph

HOW TO START YOUR ANALYSIS AFTER INSTALLATION

Compile and run your analysis code in the corresponding folder as
Ex:

```
> cd FF
> ./root_analysisd3pd ../../data/ttbar/NTUP_TOP.366473._000115.root.1 -FF 1
```

This will produce a set of .root files. You can get histograms simply writing;

```
> root
> TFile *_file0 = TFile::Open("<filename>.root")
> <histogramname>->Draw()
```

IMPORTANT !!

- Use -m64 option if you have 64-bit g++ compiler installed. Beware that runme.sh script also uses -m32.

HOW TO RUN YOUR ANALYSIS CODE WITH GRID

Analysis code includes a Makefile that can produce files to run on the GRID.
Keep in mind that you should have a valid certificate and PANDA account in order to successfully submit GRID jobs.

```
> cd FF (or go into your analysis directory)
> make grid
> sftp <username>@lxplus.cern.ch
> put 4grid.tgz
> exit
> ssh <username>@lxplus.cern.ch
> tar -xvf 4grid.tgz
> asetup 16.6.5
> source /afs/cern.ch/atlas/offline/external/GRID/DA/panda-client/latest/
etc/panda/panda_setup.sh
> export PATHENA_GRID_SETUP_SH=/afs/cern.ch/project/gd/LCG-share/current/
etc/profile.d/grid_env.sh
> source /afs/cern.ch/atlas/offline/external/GRID/ddm/DQ2Clients/setup.sh
> voms-proxy-init -voms atlas
> ./root_analysisd3pdmake.sh
> make ( before submitting to grid - to be sure 4grid dir compiles fine)
```

Now you are ready to submit your jobs if there are no errors about above process.

```
> cd 4grid
> ./rungrid.sh [runType] [channel]
```

where runType is singleT or ttbar or Wjets or dataI or dataF... and channel is electrons or muons... read the code for details
or

```
> ./rungrid.sh [runnumber] [channel] (e.g where runnumber is 105507..)
```

Before sourcing rungrid.sh make sure that you specified correct D3DPs and PANDA username in the rungrid.sh file.

After submitting your jobs, you will get some job ids.
Use Panda monitor with the job ids to get the final results.

LICENSE