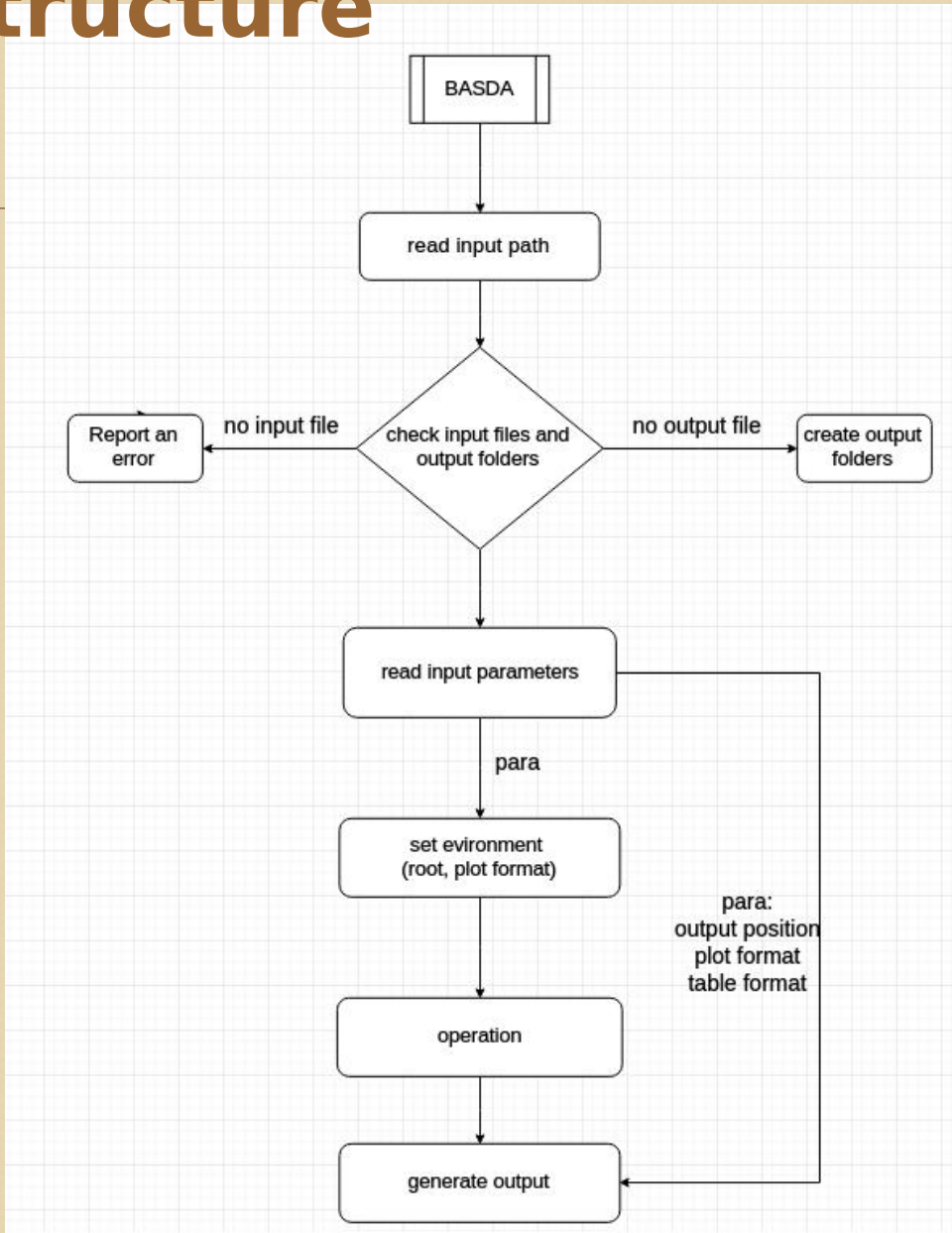
Analysis Tools

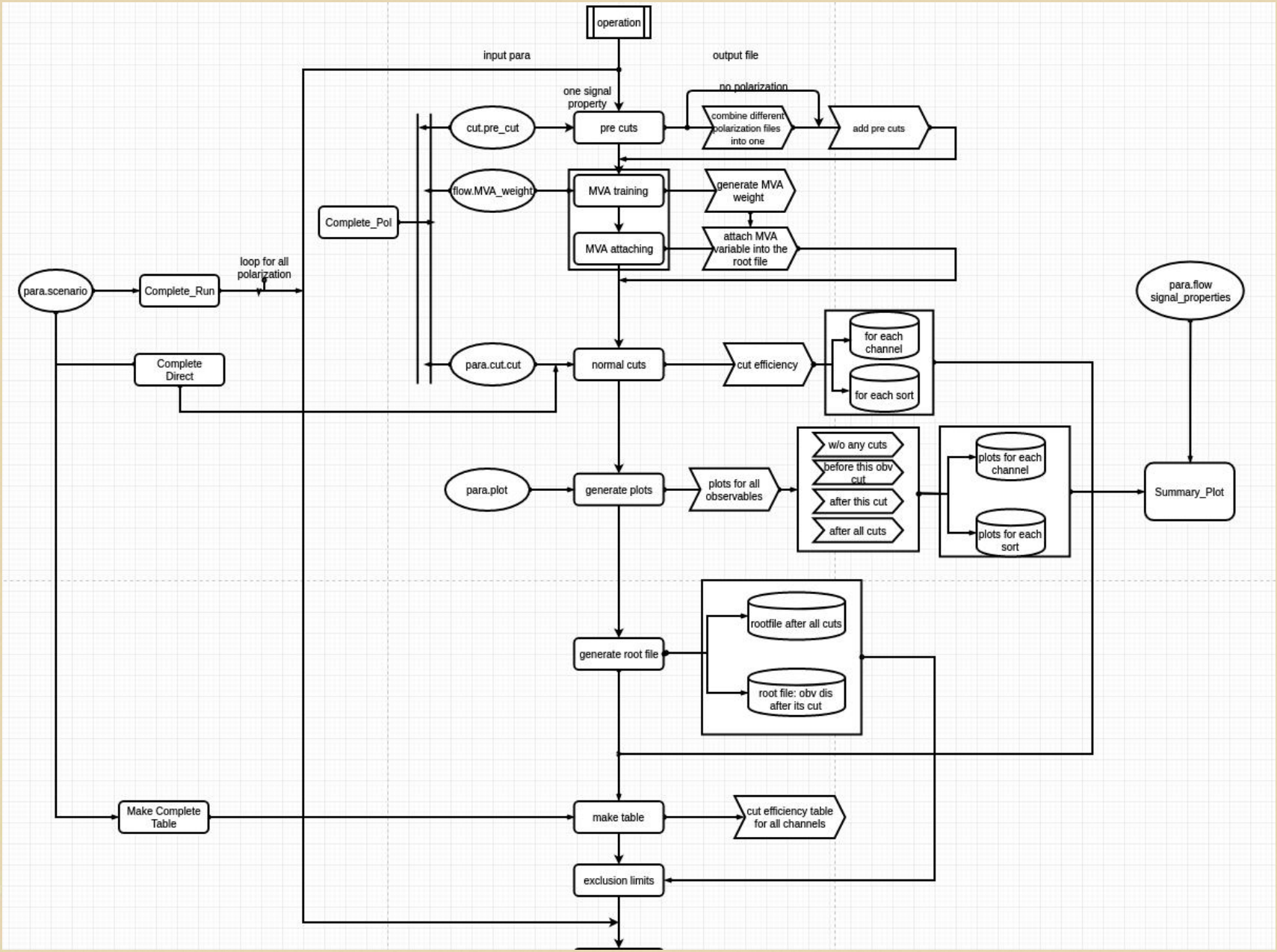
Beautiful_And_Simple_Drawing_Atificer (BASDA)

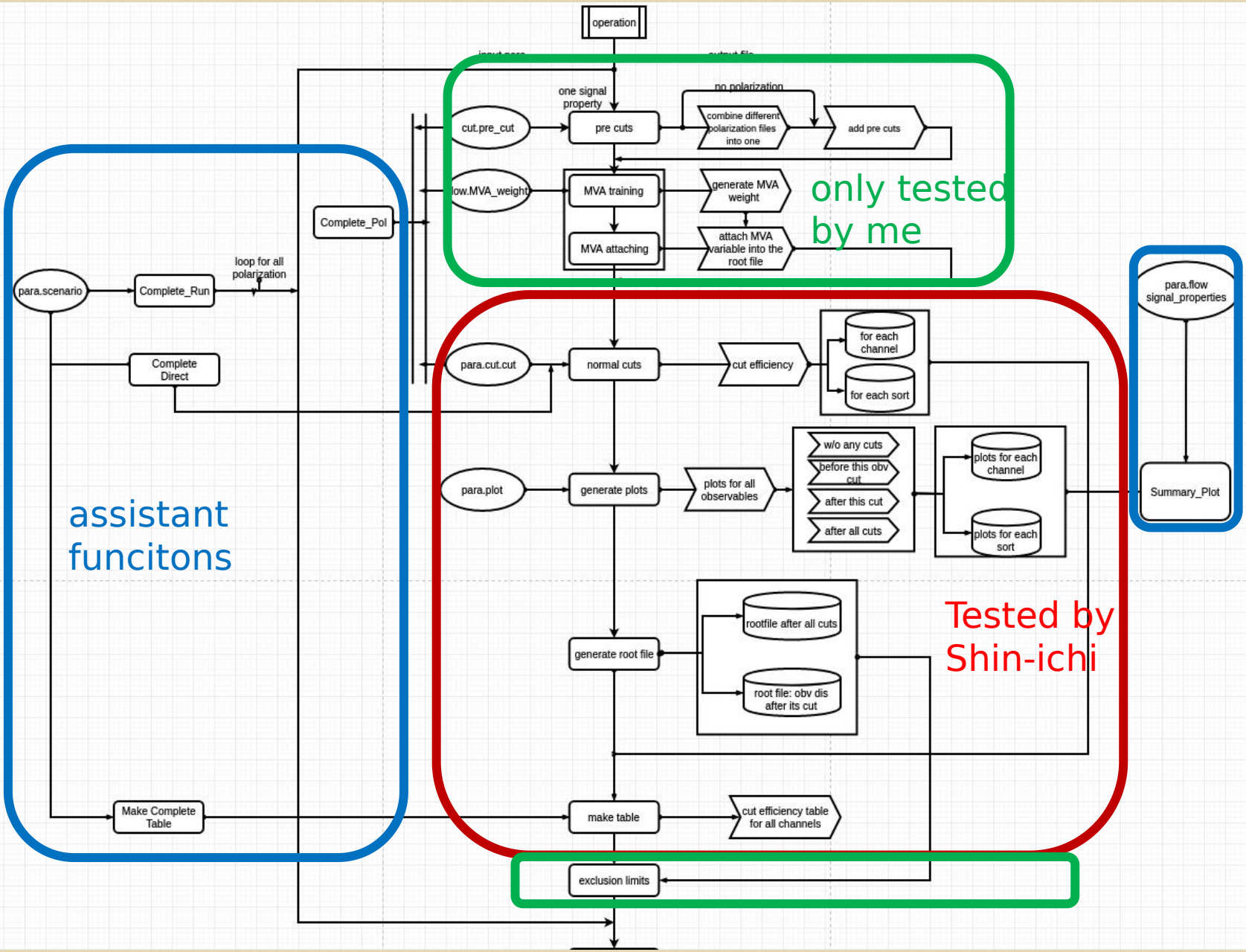
Yan Wang

09-04-2018

Code structure




operation
input para
output file
one signal property
no polarization
cut.pre_cut
pre cuts
combine different polarization files into one
add pre cuts
flow.MVA_weight
MVA training
generate MVA weight
Complete_Pol
MVA attaching
attach MVA variable into the root file
loop for all polarization
para.scenario
Complete_Run
para.flow signal_properties
Complete Direct
para.cut.cut
normal cuts
cut efficiency
for each channel
for each sort
para.plot
generate plots
plots for all observables
w/o any cuts
before this obv cut
after this cut
after all cuts
plots for each channel
plots for each sort
Summary_Plot
generate root file
rootfile after all cuts
root file: obv dis after its cut
Make Complete Table
make table
cut efficiency table for all channels
exclusion limits


operation
one signal property
no polarization
cut.pre_cut
pre cuts
combine different polarization files into one
add pre cuts
low.MVA_weight
MVA training
generate MVA weight
MVA attaching
attach MVA variable into the root file
only tested by me
Complete_Pol
loop for all polarization
para.scenario
Complete_Run
Complete Direct
para.cut.cut
normal cuts
cut efficiency
for each channel
for each sort
para.plot
generate plots
plots for all observables
w/o any cuts
before this obv cut
after this cut
after all cuts
plots for each channel
plots for each sort
para.flow signal_properties
Summary_Plot
assistant funcitons
generate root file
rootfile after all cuts
root file: obv dis after its cut
Tested by Shin-ichi
Make Complete Table
make table
cut efficiency table for all channels
exclusion limits

Improvement since last meeting

- code structure
- support "integer", "float" and "double"
  - force them into float
- fix some bugs

Improvement since last meeting

- improve offriendly interface
  - basic/advanced configs
  - input for a sequence
- generalize more settings into config files (MVA, plot ...)
- improve safety (report errors, tell where/how it happens)
- add more for introductions/README/Usage

past

vim Bkg_Sort.dat

```
sort_num    : 3

sub_sort_num_0 : 1
sub_sort_num_1 : 2
sub_sort_num_2 : 2

sub_sort_name_0 : nh
sub_sort_name_1 : 4f
sub_sort_name_2 : 2f

nh_0: 115;higgs_{115}

4f_0 : l
4f_1 : sl

2f_0 : l
2f_1 : bha
```

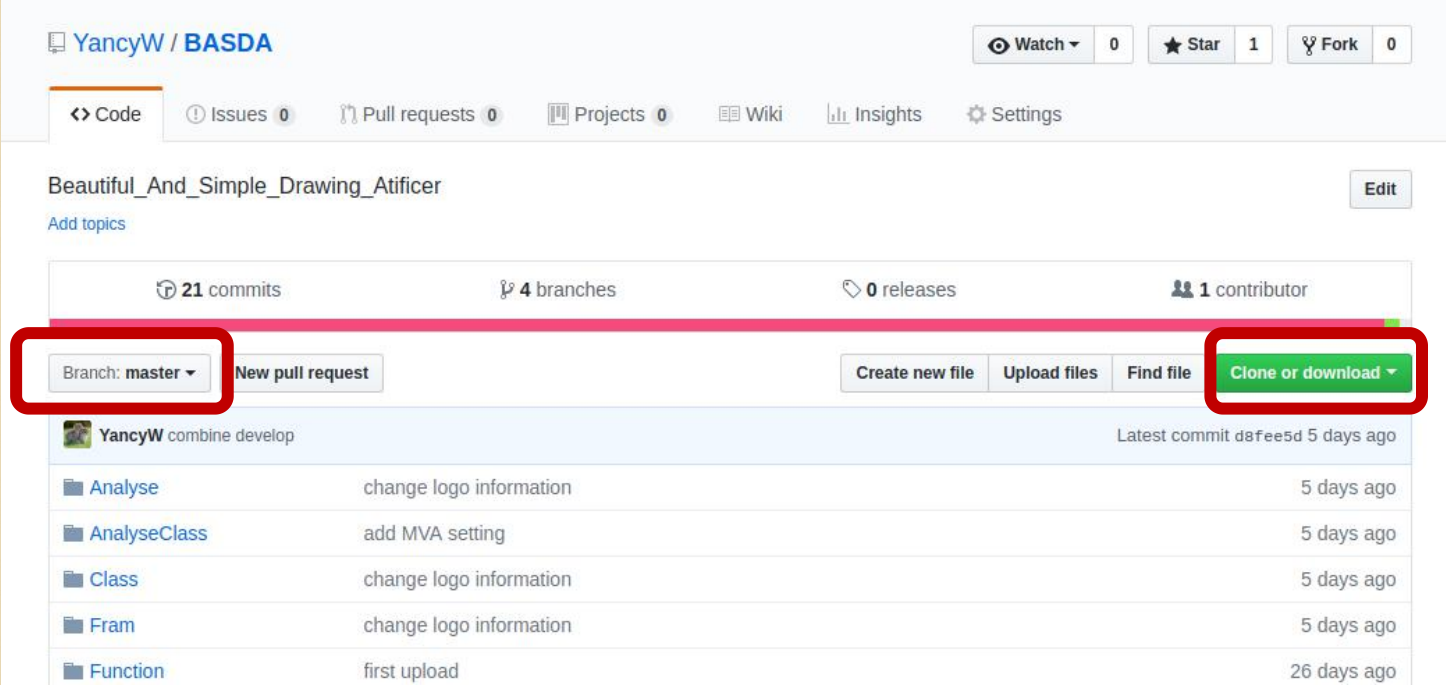
vim Bkg_Sort.dat

```
# do not change the key name
sort_num    : 4

sub_sort_name : [sig, wtata, tt, txll]
sub_sort_num  : [1,   1,     2,  1   ]

# change below based on sub_sort names
sig   : [HA]
wtata : [l]
tt    : [l, sl]
txll  : [l]
```

current

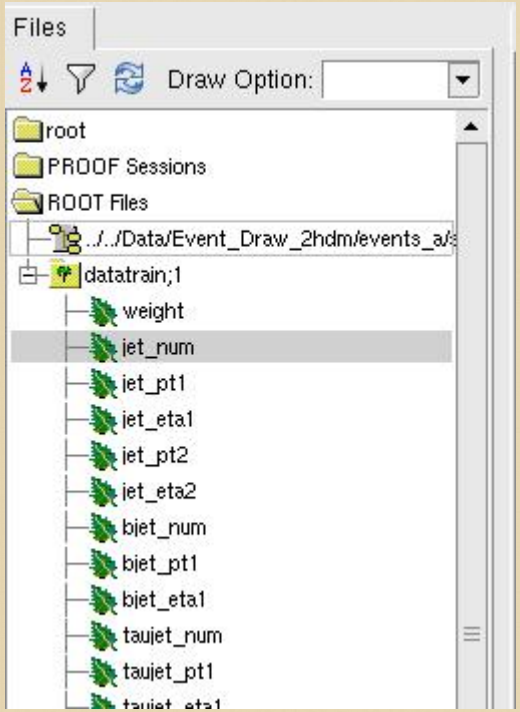
Installation

- https://github.com/YancyW/BASDA
  - four branches: master, release, develop, feature.



installation

- c++14 ⊕ root6 ⊕ boost_filesystem
- yaml-cpp
  - https://github.com/jbeder/yaml-cpp
- input root file structure
  - weight
  - observables



Installation (on NAF)

- initialize ilcsoft for gcc ≧ 4.8, root 6
- install YAML, change path for makefile_server
- in BASDA, cover makefile with makefile_server
  - mv makefile_server makefile

51 lines (40 sloc) | 1.92 KB Raw Blame History

```
TARGET=BASDA

CXX=g++
FLAGS=-std=c++14  -Wextra  -Wall ${shell root-config --cflags}

RootPath=/cvmfs/ilc.desy.de/sw/x86_64_gcc48_sl6/v01-17-11/root/6.08.00
YAMLPath=/afs/desy.de/user/y/ywang/Software/Lib/yaml-cpp-master/

INCLUDE = -I. -I..  -I$(RootPath)/include  \
                   -I${YAMLPath}/include

LIBS =  -L${YAMLPath} -lyaml-cpp -lboost_filesystem  -lboost_system -lTMVA -lTMVAGui
ROOTLIBS := $(shell root-config --libs) -lXMLIO

SRCS = \
$(TARGET).cpp \
Class/Path.cpp\
Fram/Fram.cpp Fram/Fsection.cpp \
Function/Fbasic.cpp Function/Fvector.cpp Function/Ffile.cpp Function/FCalc.cpp Function/FString.cpp \
Latex/LTable.cpp Latex/LItem.cpp\
Lib/MessageFormat.cpp Lib/PlotSetting.cpp Lib/Record.cpp Lib/YAML_Assistant.cpp\
RWpara/RWFile.cpp RWpara/RWFlow.cpp RWpara/RWPath.cpp RWpara/RWDebug.cpp RWpara/RWDraw.cpp RWpara/RWbasic.cpp RWpara/RWEvent.cpp
RWpara/RWSensitivity.cpp RWpara/RWCut.cpp RWpara/RWPlot.cpp RWpara/RWMass_Scan.cpp RWpara/RWBKG_Sort.cpp RWpara/RWScenario.cpp R
Analyse/Analyse_Find_Cut.cpp  Analyse/Make_Table.cpp \
Analyse/Make_Complete_Table.cpp \
Analyse/Analyse_Pre_Cut.cpp \
```

change YAML path

YAML format

- YAML---YAML Ain't Markup Language
  - download/manuals https://github.com/jbeder/yaml-cpp
- a human-readable data serialization language
- in BASDA, only use simplest command

```
key :  value
```

or,

```
key :
    key :  value
    key :
            key :  value
```

or list,

```
key: [value1,value2, value3...]
```

YAML format

space, NOT tab

note: DO NOT use tab in YAML.
DO NOT try to modify everything.
comments with #

vim Var_2hdm_a.dat

```
weight:
    Exist                    : False
    Weight_Type              : "F"
variable:
   v0:
     |  title_name           : "jet_num"
     |  cut_min              : 2.9
     |  cut_max              : 100
     |  latex_name           : "N^{j}"
     |  Canvas_name          : "c0"
     |  Canvas_width         : 1000
     |  Canvas_height        : 700
     |  leg_left             : 0.6
     |  leg_up               : 0.7
     |  leg_right            : 0.8
```

vim Bkg_Sort.dat

```
# do not change the key name
sort_num    : 4

sub_sort_name : [sig, wtata, tt, txll]
sub_sort_num  : [1,   1,     2,  1   ]

# change below based on sub_sort names
sig   : [HA]
wtata : [l]
tt    : [l, sl]
txll  : [l]
```

```
<marlin>
  <execute>
    <processor name="MyAdd4MomCovMatrixCharged"/>
    <!--processor name="MyThrustReconstruction"/-->
    <processor name="MyIsolatedLeptonTaggingProcessor"/>
    <processor name="MyHiggsProcessor"/>
  </execute>

  <global>
    <parameter name="LCIOInputFiles">
/pnfs/desy.de/ilc/prod/ilc/mc-dbd/ild/dst-merged/250-TDR_ws/np-light-higgs/ILD_o1_v05/v01-16-p10_250
    </parameter>
    <parameter name="GearXMLFile"> GearOutput.xml </parameter>
    <parameter name="MaxRecordNumber" value="0" />
    <parameter name="SkipNEvents" value="0" />
    <parameter name="SupressCheck" value="false" />
    <parameter name="Verbosity" options="DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT">WARNING<
    <parameter name="AllowToModifyEvent" value="true" />
  </global>

  <processor name="MyAdd4MomCovMatrixCharged" type="Add4MomCovMatrixCharged">
    <parameter name="InputPandoraPFOsCollection" type="string"> PandoraPFOs </parameter>
    <parameter name="Verbosity" options="DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT"> MESSAGE
  </processor>
                                                                    4,1           Top
```

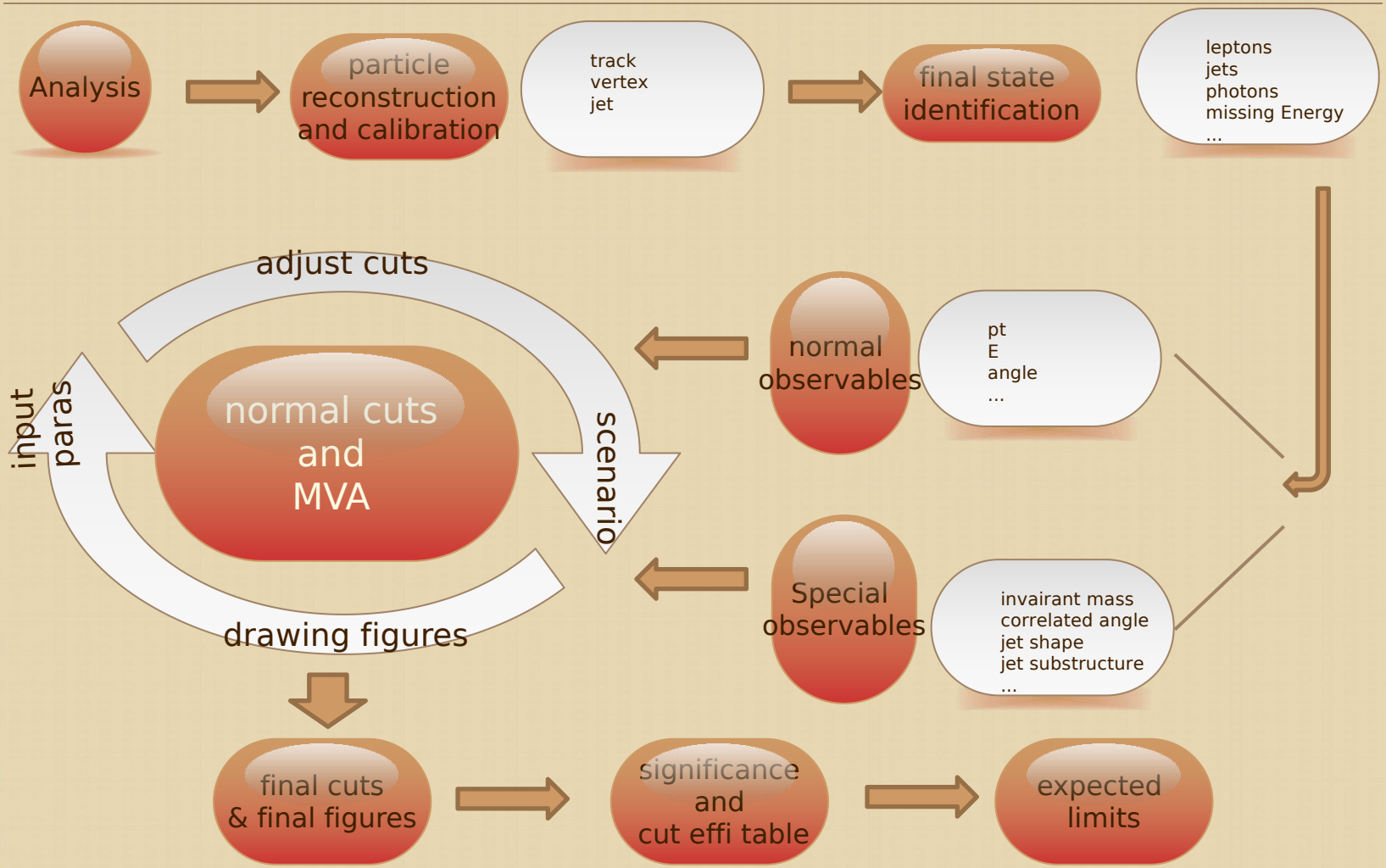
Motivation

Motivation: a standard analysis code will be helpful!

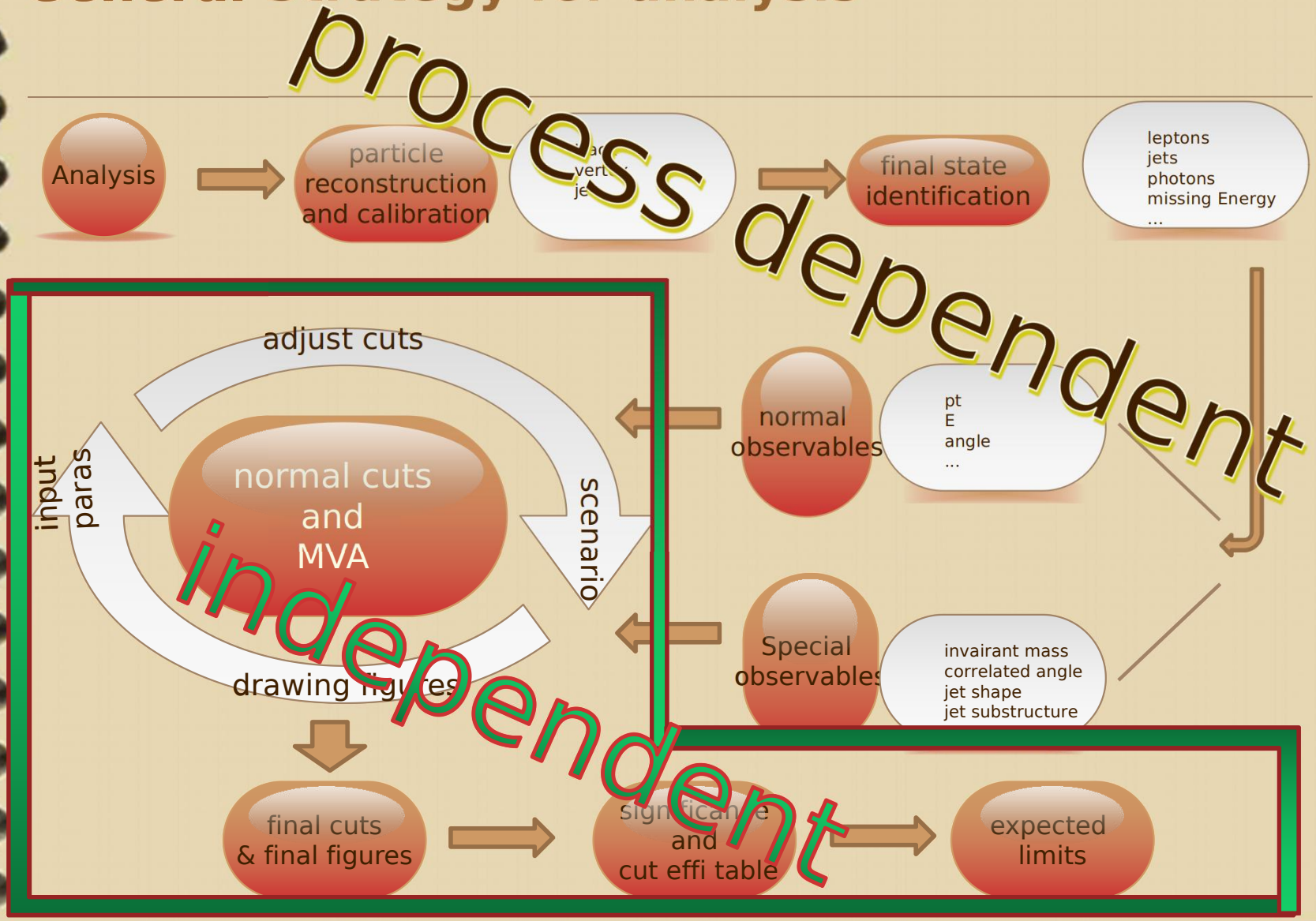
Physics Benchmarks

WG	Process	Physics	Detector	ECM	Who
Higgs & EW	**H->bb/cc/gg**	**BR**	**c-tag, b-tag, JER**	**500 GeV**	**NN + NN**
	H->bb	mass	JER, JES	500 GeV	Ali Ebrahimi (10%) + Junping Tian
	ee->tautau	A_FB, tau-pol, A_LR	tau-reco	500 GeV	Daniel Jeans + **NN**
	H->mumu	BR	momentum resolution	500 GeV	Shin-ichi Kawada + **NN**
	H->invisible	BR limit	JER, hermeticity	500 GeV	Yu Kato + **NN**
	WW->qqlv	MW, TGCs, beam pol.	JES, JER, electron, mu	500 GeV	Kostiantyn Shpak + **NN**
	vvqqqqq	**QGCs**	**JES / JER**	**1 TeV**	**Jakob Beyer + NN**
	gamma Z	**A_LR, sigma_tot, JES**	**photon, JER/JES, e, mu**	**500 GeV**	**NN + NN**
Top, Bottom & QCD	tt->bbqqqq	x-section, AFB	b-tag, vertex charge, PID	500 GeV	Amjad + **NN**
BSM	low deltaM Higgsinos	natural SUSY	low-p tracking, PID, hermeticity	500 GeV	Swathi Sasikumar + **NN**
	mono-photons	**WIMPs / WISPs**	**photon reco, BeamCal**	**500 GeV**	**NN + NN**
	Zh, mh < 125 GeV	limit on ZZh coupling	p res, e reco, JER, hermeticity	500 GeV	Yan Wang + **NN**

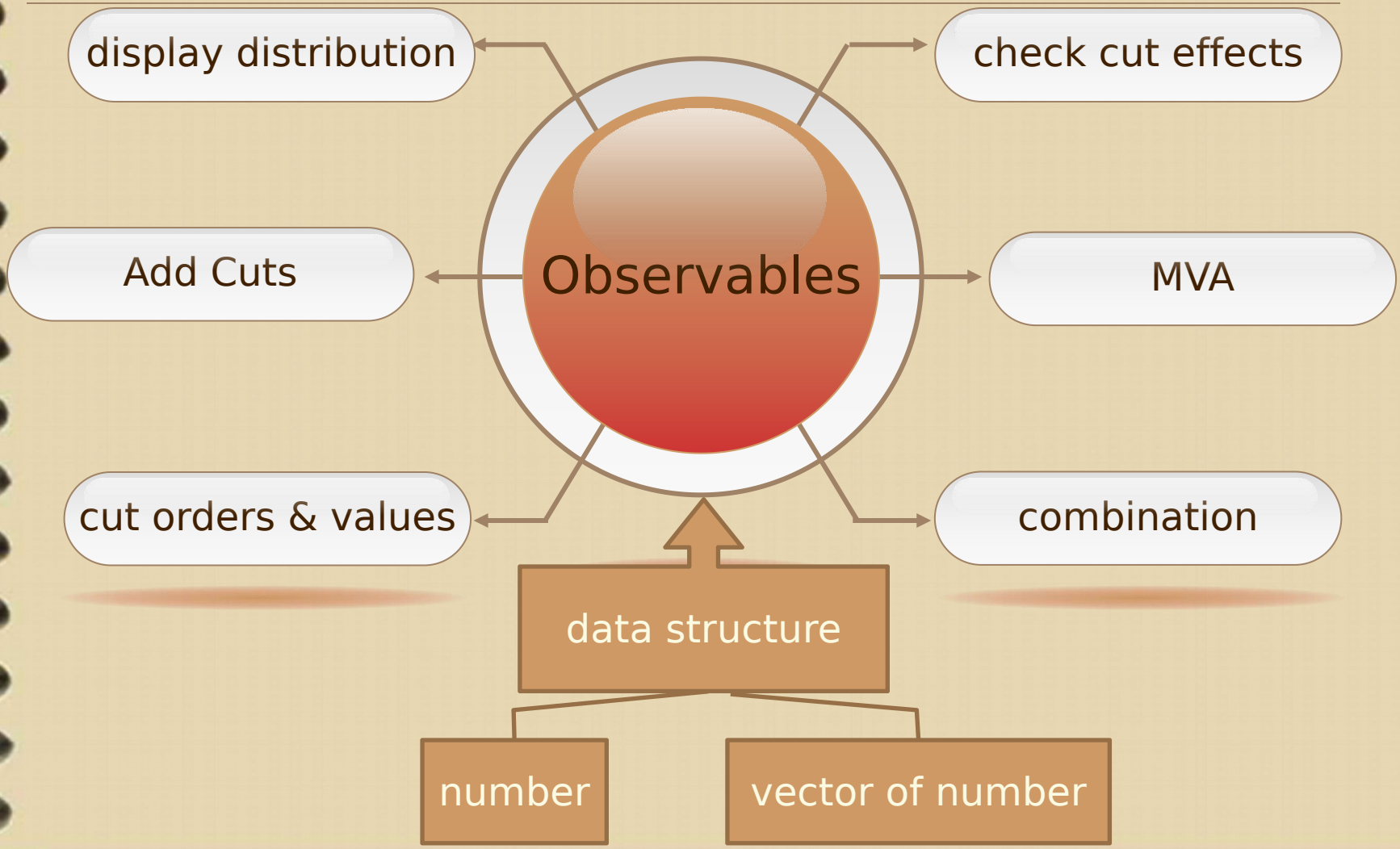
General Strategy for analysis



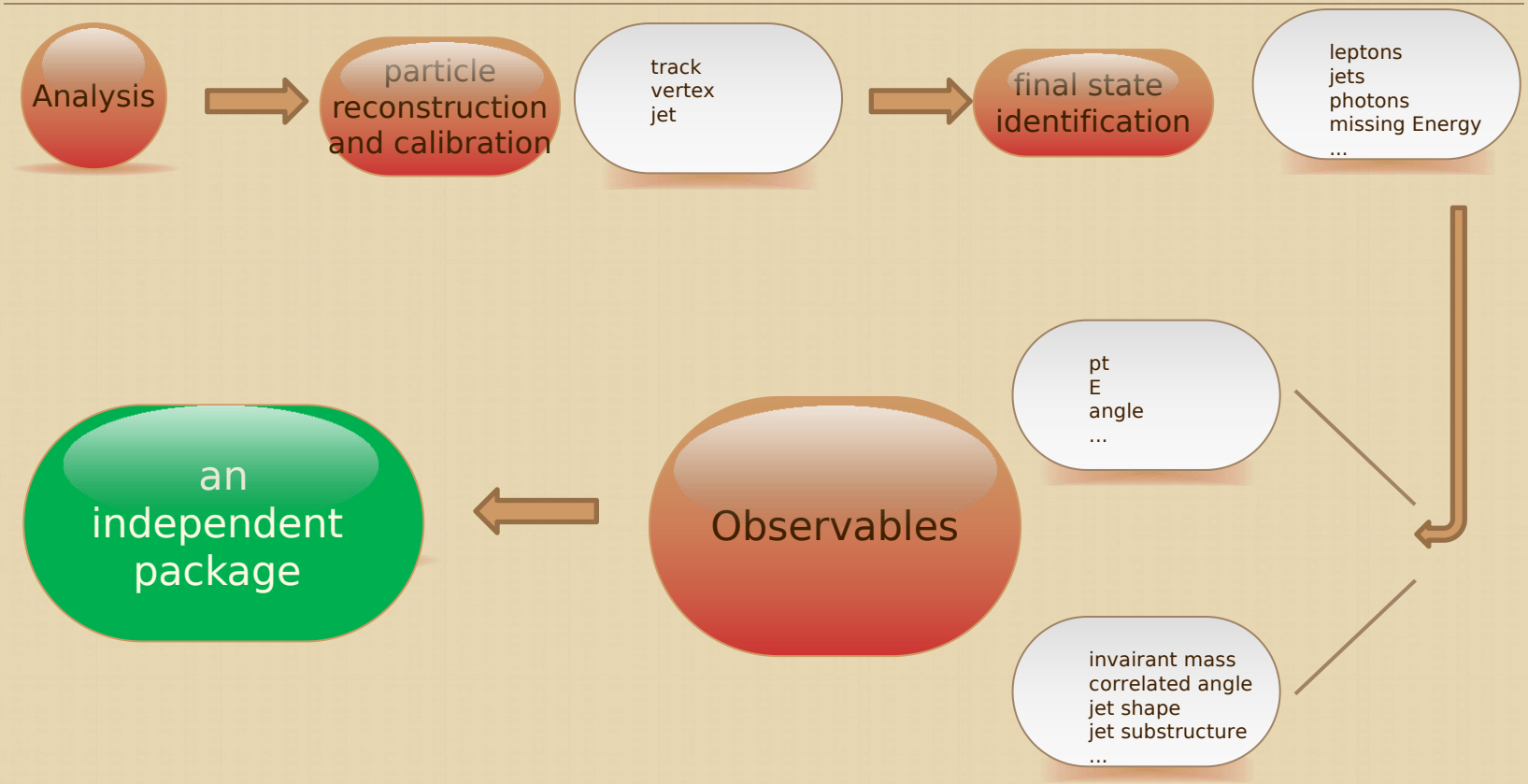
General Strategy for analysis



The key role


display distribution
check cut effects
Add Cuts
Observables
MVA
cut orders & values
combination
data structure
number
vector of number

General Strategy for analysis



Beautiful_And_Simple_Drawing_Atificer
(BASDA)

(BASDA)

The beautiful figure

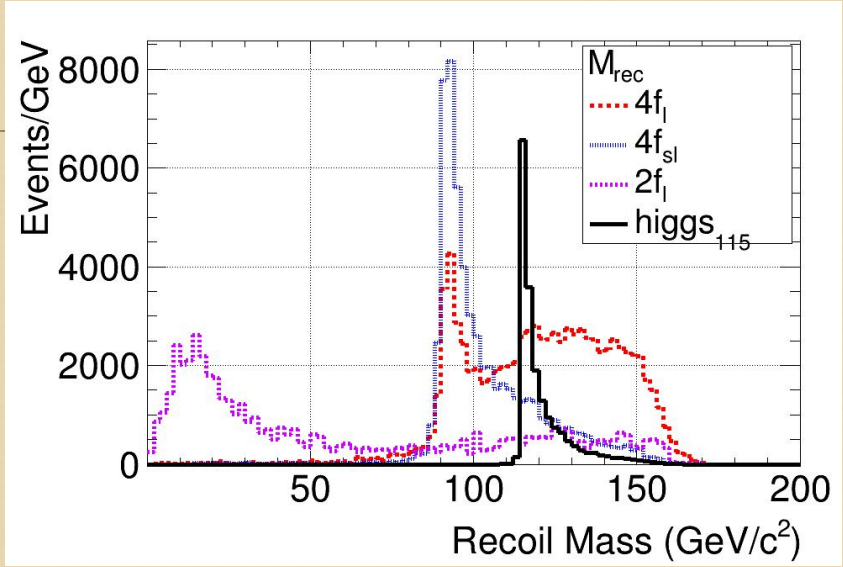
- plot templates
- easy to adjust
- once fixed, used forever

The simple operation

all with control files

Easy to repeat

only need to
preserve control file.



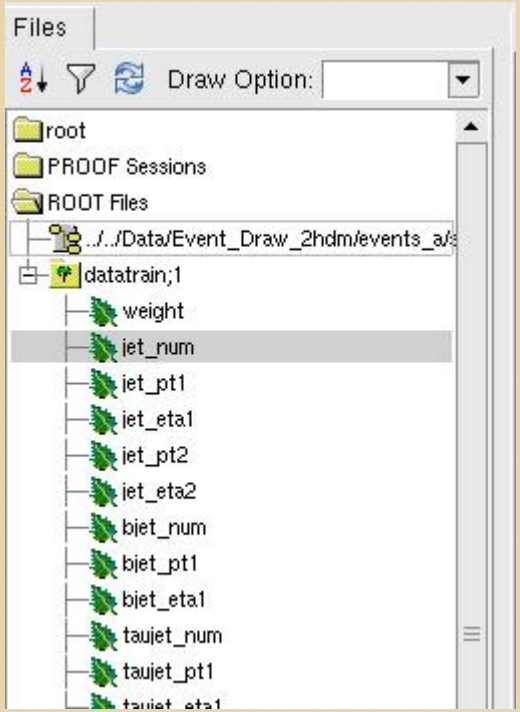
Release time from programing.

Focus on **Physics**!

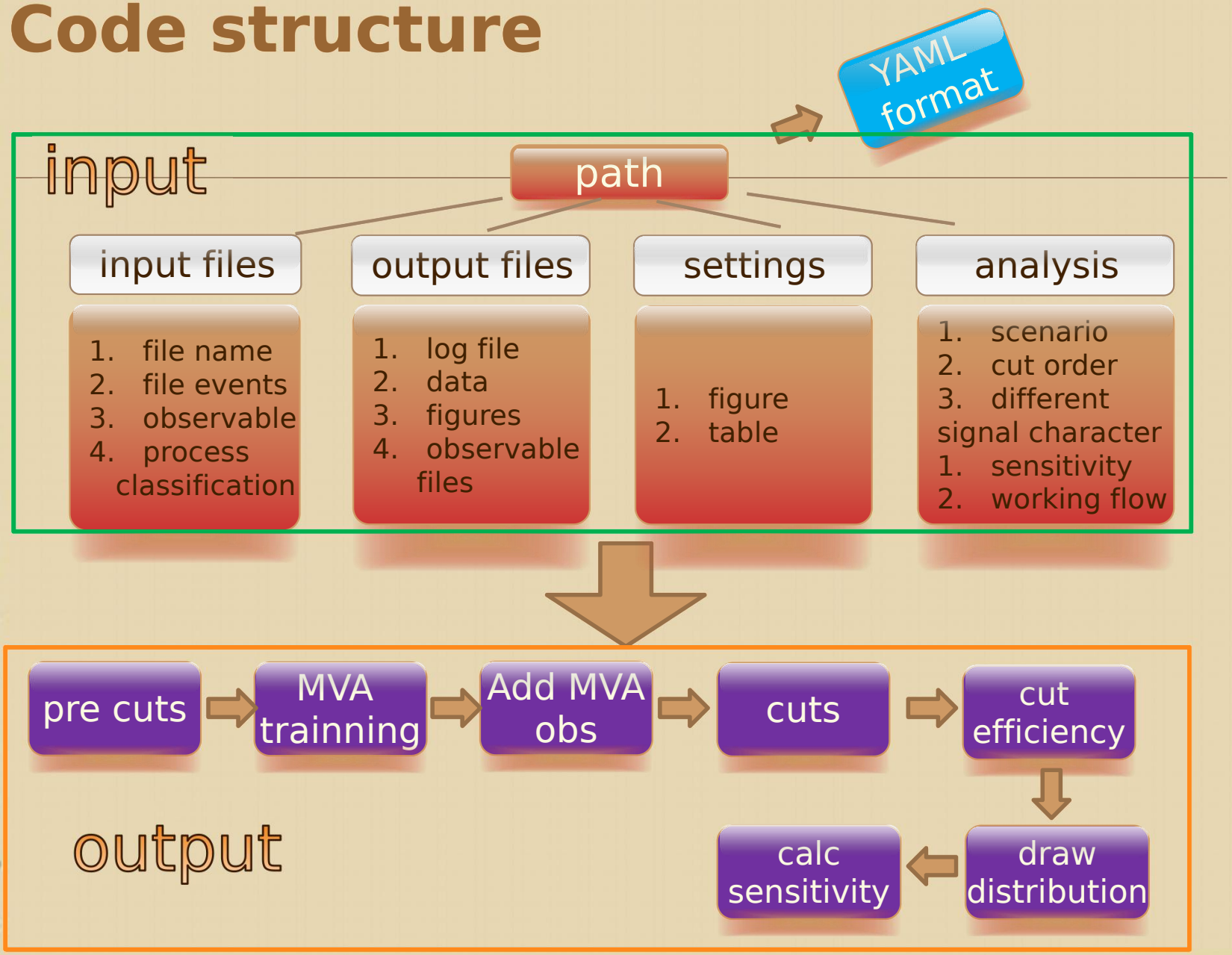
Cross check

installation

- c++14 ⊕ root6 ⊕ boost_filesystem
  - ilcsoft environment
  - for gcc ≧ 4.8, root 6
- yaml-cpp
  - https://github.com/jbeder/yaml-cpp
- input root file structure
  - weight
  - observables



Code structure



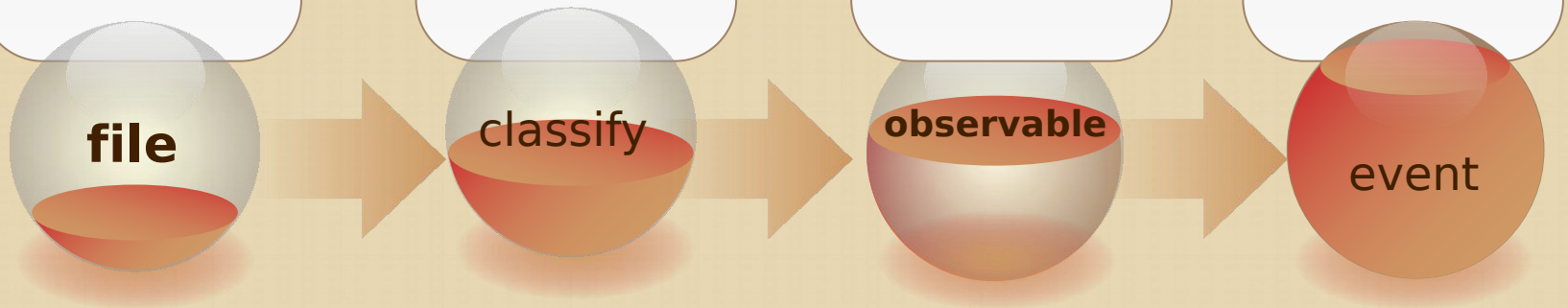
input data

which file for analysis

sort A: file 1,2
sortB: file 3,
sortC: file 4,5,6

name,
figure setting for this obs,
cut,

which event for running,
all,
the first 100,
101-303...



input example

Path

process name

main output folder,

- **sub folders are under this**
- **if doesn't exist, code will create one**

sub output folder

input folder

- **if doesn't exist, report error**

advanced input folder

vim path.dat

```
PROCESS           : zh

# output folder
OUTPUT_FOLDER     : '../../Data/Event_Analyse'

RECORD_FILE       : 'tmp/nh_115.dat'
OUTPUT_FILE       : 'data'
EVENT_FILE        : 'results'
SINGLE_PLOT       : 'plot'
COMBINE_PLOT      : 'plot_compare'
HORIZONTAL_SUMMARY_PLOT : 'Event_Analyse/plot_summary'


# input folder
FLOW_FILE         : './control/flow.dat'
INPUT_FILE        : './control/file.dat'
BKG_SORT_FILE     : './control/Bkg_Sort.dat'

CUT_FILE          : './control/Cut/Cut_nh.dat'
VAR_FILE          : './control/Var/Var_nh_115.dat'

MASS_SCAN_FILE    : './control/Mass_Scan.dat'


#Advanced input setting
PLOT_FILE         : './control/advanced/Plot.dat'
DEBUG_FILE        : './control/advanced/debug.dat'
ANALYSIS_FILE     : './control/advanced/Event.dat'
SENSITIVITY_FILE  : './control/advanced/Sensitivity.dat'
SCENARIO_FILE     : './control/advanced/Scenario.dat'
```

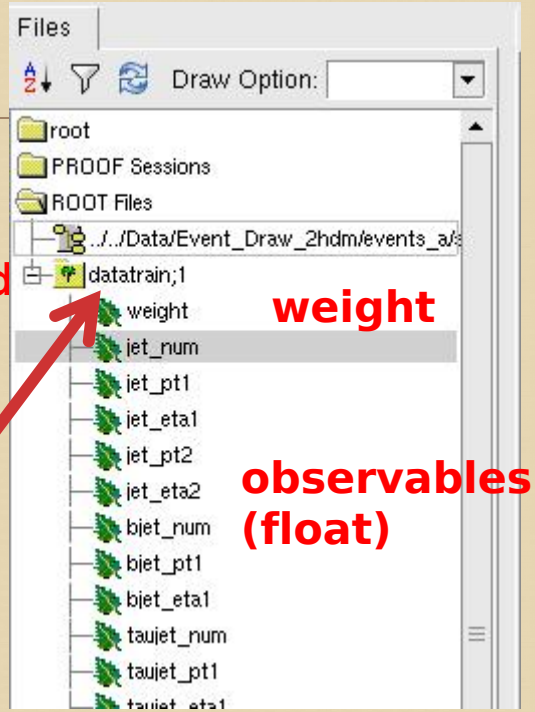
input file

```
FILE_NUM            : 1
FILE_0              :  ./control/xection_aa.dat

CHANNEL_NUM         : 1
FILE_DESCRIP_0      : all
#FILE_DESCRIP_0      : nh
#FILE_DESCRIP_1      : 4f_{zz}^{l}

Root_Head_Name      : datatest
Root_Head_BDT_Name  : datatrain
```

channel used in this time

root tree name

input root file structure



weight

observables (float)

xection file

```
vim xection_aa.dat
root file 10:
    nh_{10}^{10}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/nh_10_eL.pR.root  :  79.026852
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/nh_10_eR.pL.root  :  50.415099
    4f_{ww}^{l}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_ww_l_eL.pR.root  :  1564.2091
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_ww_l_eR.pL.root  :  14.691728
    4f_{ww}^{sl}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_ww_sl_eL.pR.root  :  18780.976
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_ww_sl_eR.pL.root  :  172.73264
    4f_{zz}^{l}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zz_l_eL.pR.root  :  157.96041
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zz_l_eR.pL.root  :  99.506102
    4f_{zz}^{sl}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zz_sl_eL.pR.root  :  1422.1429
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zz_sl_eR.pL.root  :  713.52633
    4f_{zzww}^{l}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zzww_l_eL.pR.root  :  1636.0362
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zzww_l_eR.pL.root  :  53.955513
    4f_{zzww}^{sl}:
    |   no such class:  0
    4f_{szee}^{l}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_szee_l_eL.pR.root  :  1084.0873
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_szee_l_eR.pL.root  :  1019.5228
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_szee_l_eL.pL.root  :  1009.6032
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_szee_l_eR.pR.root  :  1008.4063
```

xection file

signal property:

- **seperate different analysis processes**
- **every mh can different input files**

channel name: a_{b}^{c}

vim ection_aa.dat

```
root file 10:
    nh_{10}^{10}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/nh_10_eL.pR.root  :  79.026852
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/nh_10_eR.pL.root  :  50.415099
    4f_{ww}^{l}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_ww_l_eL.pR.root  :  1564.2091
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_ww_l_eR.pL.root  :  14.691728
    4f_{ww}^{sl}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_ww_sl_eL.pR.root  :  18780.976
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_ww_sl_eR.pL.root  :  172.73264
    4f_{zz}^{l}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zz_l_eL.pR.root  :  157.96041
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zz_l_eR.pL.root  :  99.506102
    4f_{zz}^{sl}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zz_sl_eL.pR.root  :  1422.1429
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zz_sl_eR.pL.root  :  713.52633
    4f_{zzww}^{l}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zzww_l_eL.pR.root  :  1636.0362
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_zzww_l_eR.pL.root  :  53.955513
    4f_{zzww}^{sl}:
    |   no such class:  0
    4f_{szee}^{l}:
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_szee_l_eL.pR.root  :  1084.0173
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_szee_l_eR.pL.root  :  1019.228
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_szee_l_eL.pL.root  :  1009.6032
    |   /home/yancy/Code/Data/Event_Light_Higgs/origin_data/results_10/4f_szee_l_eR.pR.root  :  100.4063
```

input file name with different polarizations:

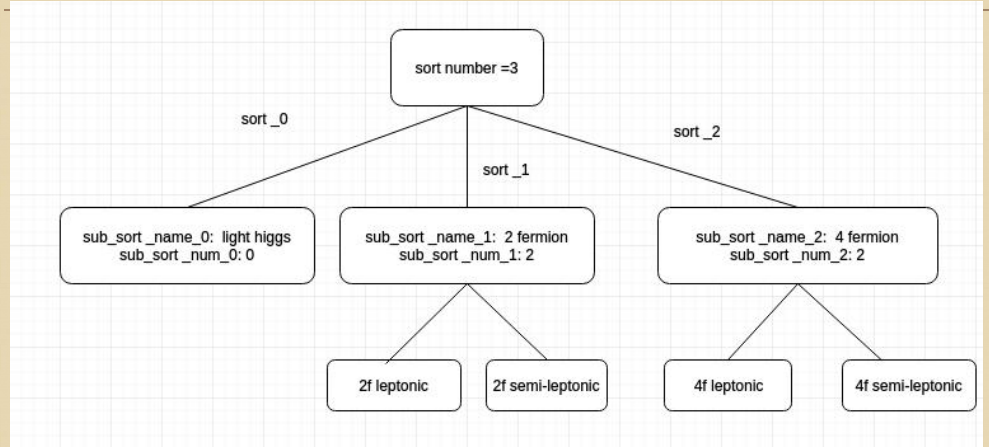
- **1 file : no polarization**
- **2 files: lr/rl**
- **4 files: lr/rl/ll/rr**

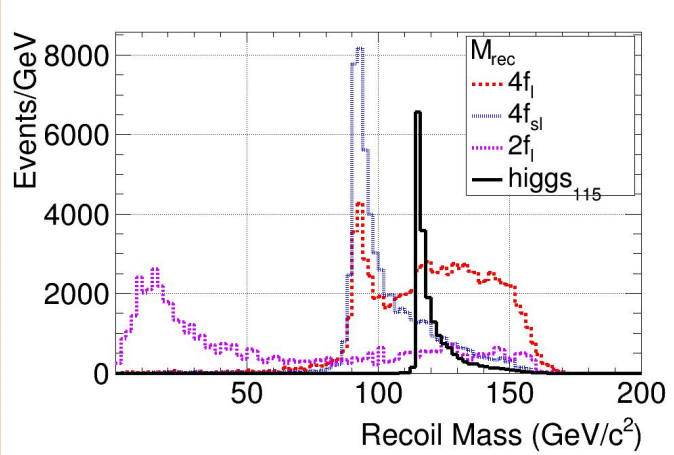
cross section:

- **works when choosing no "weight"**
- **when really don't have "weight" in input files**
- **when hoping to reweight events**

classify

channel name: a_{b}^{c}
a : sort
c : sub_sort
b : some special label






```
vim Bkg_Sort.dat
1  # do not change the key name
2  sort_num    : 4
3
4  sub_sort_name : [sig, wtata, tt, txll]
5  sub_sort_num  : [1,   1,     2,  1   ]
6
7  # change below based on sub_sort names
8  sig   : [HA]
9  wtata : [l]
10 tt    : [l, sl]
11 txll  : [l]
12
```

observables

obs name in code

obs name in input file

cut for this obs

plot para

whether use this obs as MVA input

vim Var_nh_30.dat

```
variable:
  v1:
    title_name           : "po_muon_kcut_zmass"
    cut_switch           : true
    cut_min              : 73.0
    cut_max              : 120.0
    plot_switch          : true
    latex_name           : "M_{l^{+}l^{-}}"
    Canvas_name          : "c1"
    Canvas_width         : 1000
    Canvas_height        : 700
    leg_left             : 0.75
    leg_up               : 0.7
    leg_right            : 0.9
    leg_down             : 0.9
    leg_header           : "e^{+}e^{-}->#mu^{+}#mu^{-} + X @ 250 GeV"
    xaxis_name           : "M_{Z} (GeV/c^{2})"
    xaxis_bin            : 100
    xaxis_min            : 51
    xaxis_max            : 131
    yaxis_name           : "Events"
    yaxis_bin            : 0
    yaxis_min            : 0
    yaxis_max            : 0
    log_yaxis            : true
    normalization_switch : false
    BDT                  : true
  v2:
    title_name           : "po_muon_kcut_zpt"
    cut_switch           : true
    cut_min              :   10.0
    cut_max              :   120.0
    plot_switch          : true
    latex_name           : "P_{T}^{l^{+}l^{-}}"
    Canvas_name          : "c2"
    Canvas_width         : 1000
    Canvas_height        : 700
    leg_left             : 0.80
    leg_up               : 0.7
    leg_right            : 0.95
    leg_down             : 0.9
    leg_header           : "e^{+}e^{-}->#mu^{+}#mu^{-} + X @ 250 GeV"
    xaxis_name           : "P_{T}^{Z} (GeV/c^{2})"
    xaxis_bin            : 100
    xaxis_min            : 0
    xaxis_max            : 150
    yaxis_name           : "Events"
    yaxis_bin            : 0
    yaxis_min            : 0
    yaxis_max            : 0
    log_yaxis            : true
    normalization_switch : false
    BDT                  : false
  v3:
    title_name           : "po_muon_kcut_invis_costheta"
    cut_switch           : true
```

Cut

pre-cut order

no pre cuts for this case, after Pre-CUT, event should be applied with MVA

Cut_test.dat

```
pre_cut_num : 0

pre_cut_order : []

cut_num : 17

cut_order : [0, 1, 2, 8, 9, 10, 11, 12, 13, 14, 15, 16, 19, 22, 28, 31, 29]
```

for normal cut: 17 cuts

cut order

Operation Flow

will do which operation with this code

use cut

record everything in log file

draw plots

only works for Summarize_Plot

generate events after all cuts

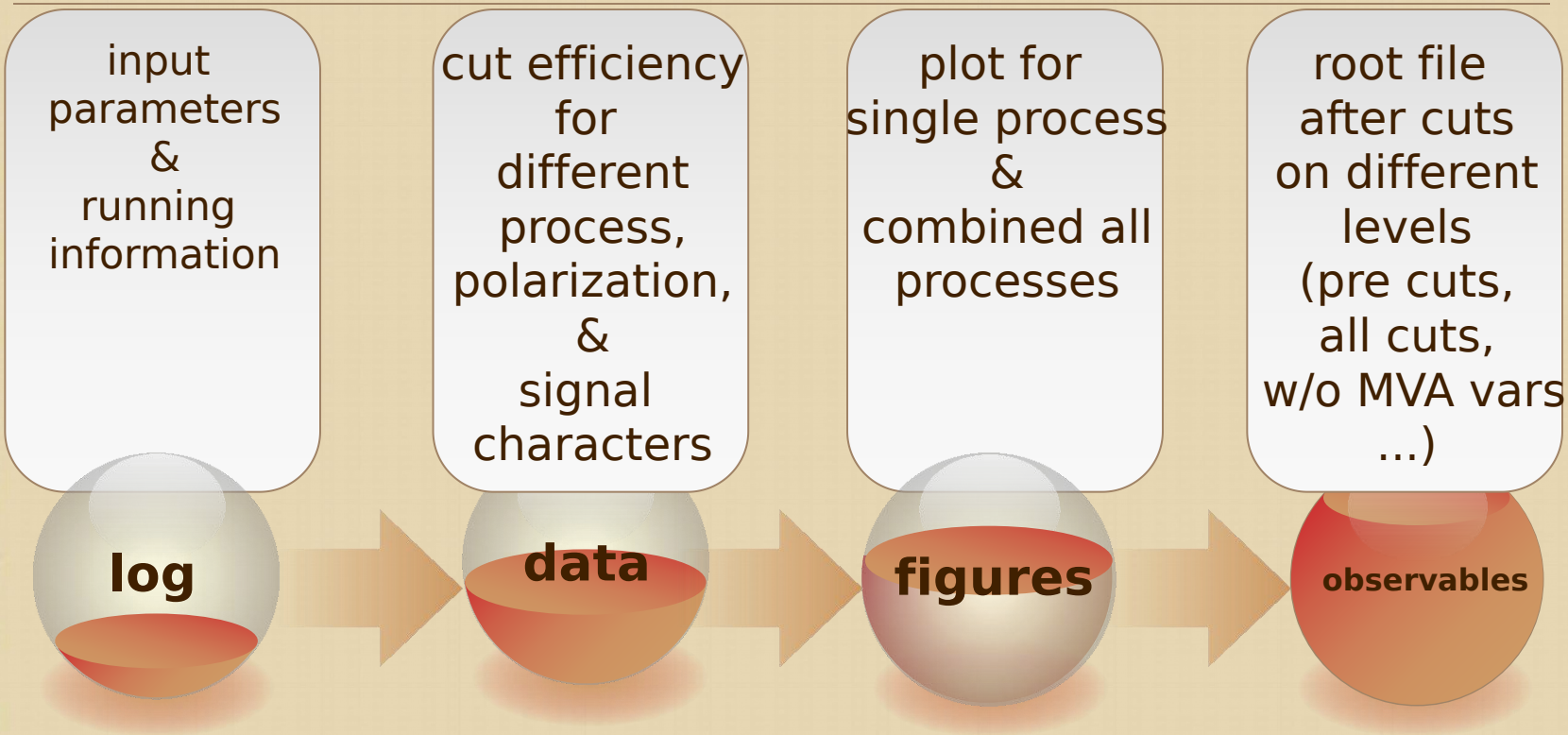
MVA setting

vim flow.dat

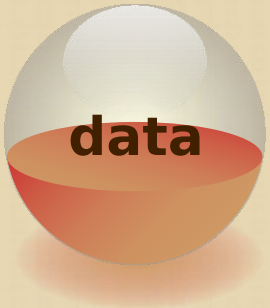
```
#operation

#BeginObject      : Complete_Run
#BeginObject      : Complete_Pol
#BeginObject      : Complete_Direct_Cut
#BeginObject      : Make_Complete_Table
#BeginObject      : Sensitivity
#BeginObject      : Pre_Cut
#BeginObject      : BDT_Train
#BeginObject      : BDT_Attach
#BeginObject      : Summarize_Plot
BeginObject       : Direct_Cut_ReWeight
#BeginObject      : Direct_Cut
#BeginObject      : Make_Table


# choose signal process and work scenario
signal_property   : 125
working_scenario  : running


# choose special setting

cut               : true
record_output     : false
plot              : true
plot_object       : before_cut
record_event      : true
MVA_training      : true
MVA_level         : 2
MVA_weight        : 1
```

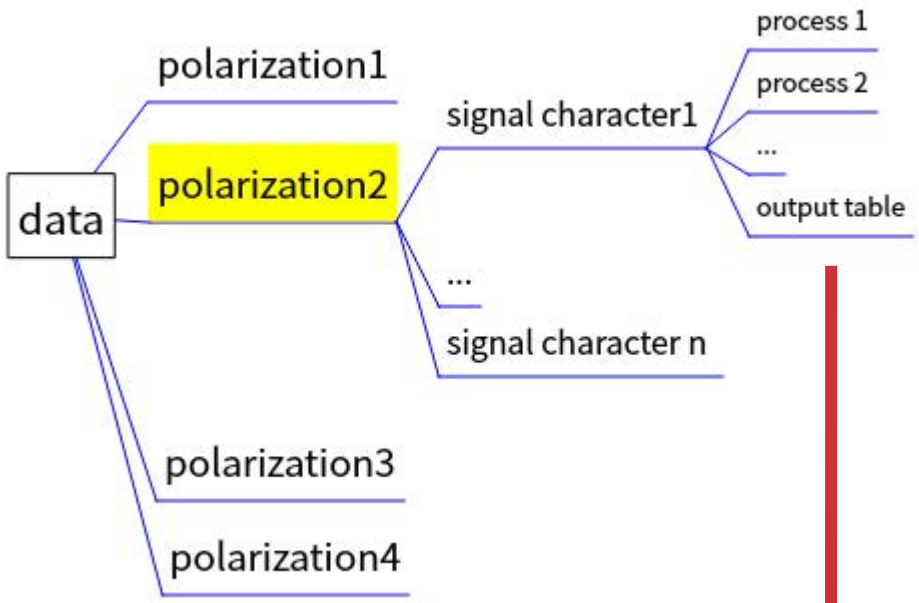
output

output data


input parameters & running information
cut efficiency for different process, polarization, & signal characters
plot for single process & combined all processes
root file after cuts on different levels (pre cuts, all cuts, w/o MVA vars ...)
log
data
figures
observables

output data structure



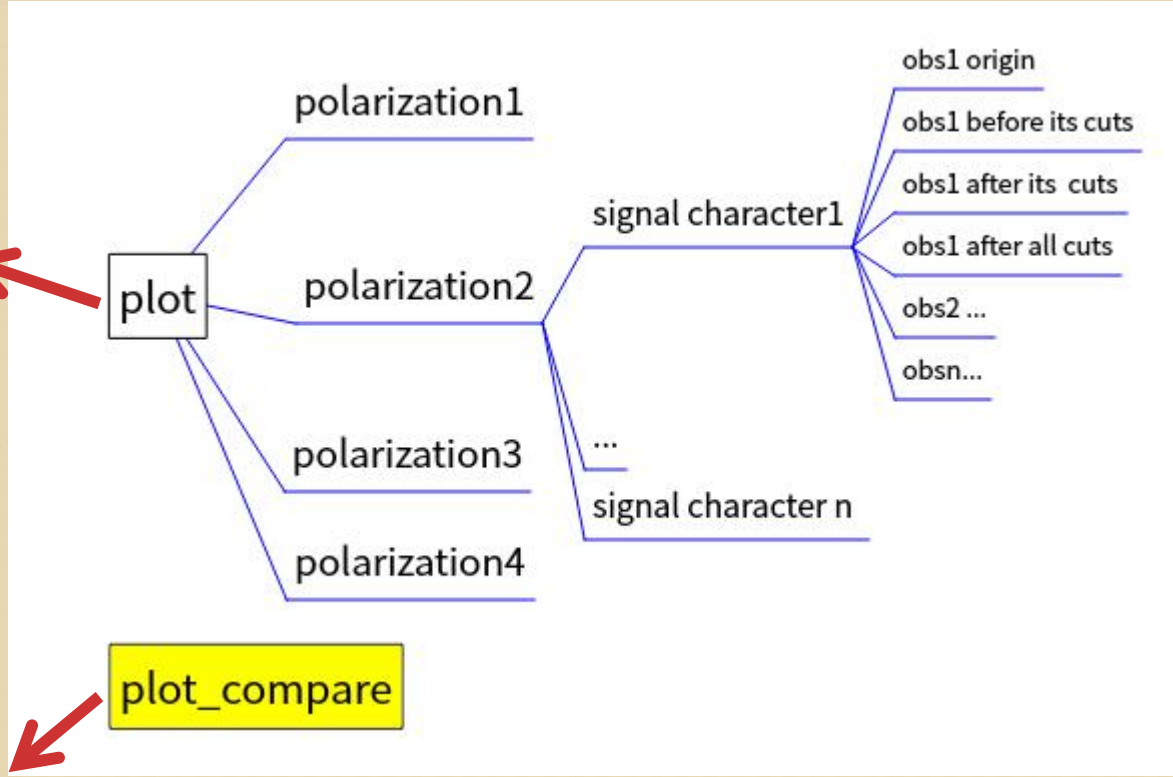


$\int Ldt = 1fb^{-1}$	sig_{ij}^{HA}	$wtata_i^j$	tt_{sd}^{i}	efficienty	significance
no cut	15.669	409.931	61989.6	1	0.0627186
$N^j \in [2.9, 100]$	12.2375	273.805	52783.2	0.781001	0.0531216
$P_T^{j1} \in [20, 1000]$	12.2375	273.805	52775.7	0.781001	0.0531254
$\eta^{j1} \in [-5, 5]$	12.2375	273.805	52775.7	0.781001	0.0531254
$P_T^{j2} \in [20, 1000]$	12.2375	273.805	52775.7	0.781001	0.0531254
$\eta^{j2} \in [-5, 5]$	12.2375	273.805	52775.7	0.781001	0.0531254
$N^\tau \in [1.9, 100]$	0.91043	17.334	1504.45	0.0581039	0.0233314
$P_T^{\tau 1} \in [20, 1000]$	0.91043	17.334	1504.45	0.0581039	0.0233314
$\eta^{\tau 1} \in [-2.5, 2.5]$	0.91043	17.334	1471.69	0.0581039	0.0235865
$P_T^{\tau 2} \in [20, 1000]$	0.91043	17.334	1471.69	0.0581039	0.0235865
$\eta^{\tau 2} \in [-2.5, 2.5]$	0.91043	17.334	1426.33	0.0581039	0.0239539
$N^{lep} \in [0.9, 100]$	0.239751	4.65097	451.083	0.015301	0.0112277
$P_T^{l1} \in [20, 1000]$	0.222514	4.20652	441.003	0.0142009	0.0105431
$\eta^{l1} \in [-2.5, 2.5]$	0.222514	4.20652	441.003	0.0142009	0.0105431
$M^W \in [70, 110]$	0.20371	2.63503	302.402	0.0130008	0.0116598
$M^A \in [0, 90]$	0.181772	2.0477	105.841	0.0116007	0.0174853
$M^{H^\pm} \in [0, 180]$	0.104989	0.603198	5.04004	0.00670043	0.0437902
$M^t \in [1, 500]$	0.103422	0.603198	5.04004	0.00660042	0.0431425
$M^t \in [0, 4000]$	0.092453	0.444462	2.52002	0.00590038	0.0528784
$\theta_*^l \in [0.8, 1]$	0.034474	0.158736	2.52002	0.00220014	0.020929
$P_T^{\not{e}} \in [0, 30]$	0.029773	0.142863	0	0.00190012	0.0716567
all cut	0.029773	0.142863	0	0.00190012	0.0716567

the output table contains cut efficiency table for all processes with "Tex format", can be directly used in latex file

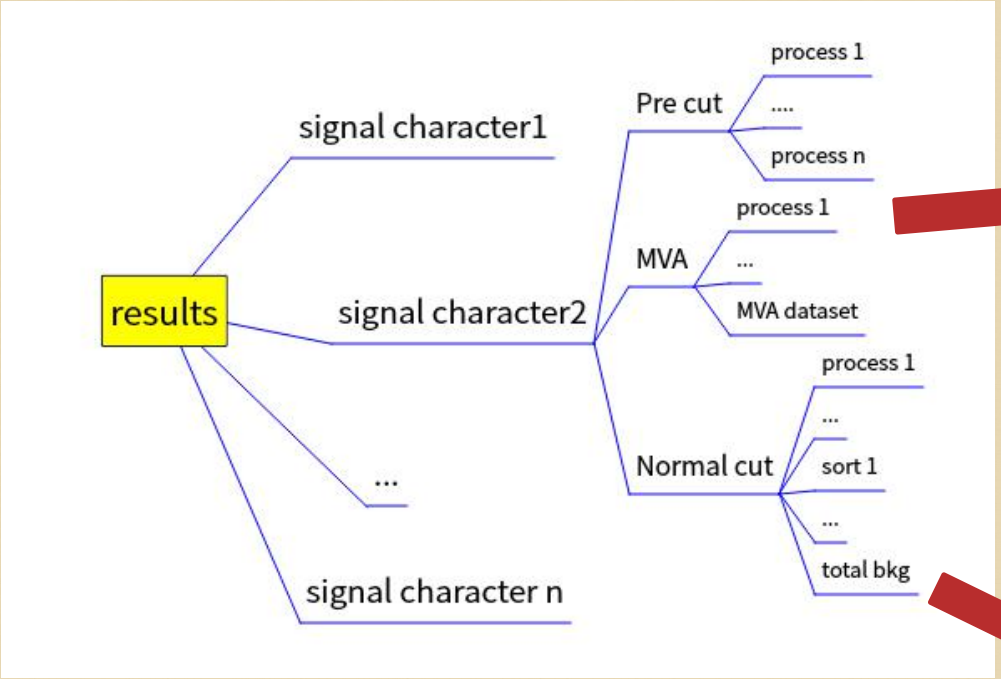
Diagram (png format)

for each process



compare all processes in one figure

Root results



1. root format
2. after cuts in different level
3. can be used independently

1. events after all cuts
2. all distributions
3. adjust diagrams directly

output example

Pre cuts

sp is "signel property" set in the flow.dat
xxx is channel names set in the xection.dat

- input: cut.pre_cut
- output:
  - data/sp/xxx_sp_Unpol.dat
    - cut efficiency for pre cuts
  - results/sp_Unpol/xxx_sp.dat
    - root file after applying pre cuts
  - results/sp_MVA/xxx_sp.dat
    - file for attaching MVA values
    - file for applying normal cuts

MVA trainning

- input: flow.MVA
- output:
  - data/sp/dataset
    - MVA weight dataset

MVA value attaching

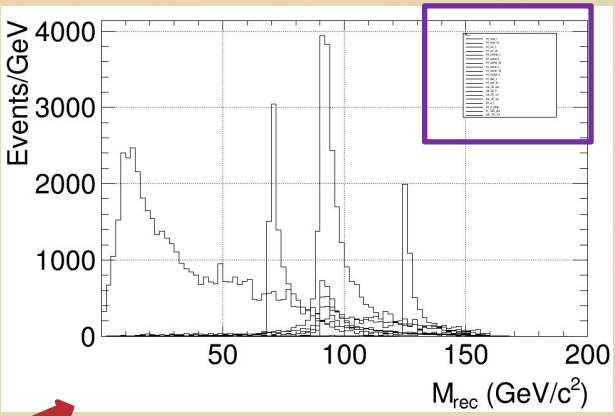
- input: MVA weight file
- output:
  - results/sp_MVA/xxx_sp.dat
    - add MVA value for each channel

normal cuts

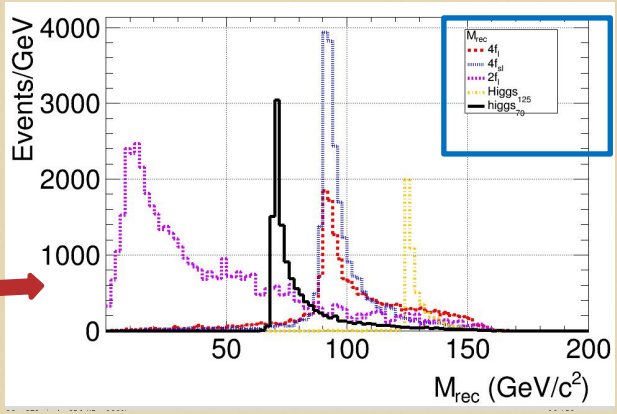
- input: cut.pre_cut
- output:
  - data/sp/xxx_sp.dat
    - cut efficiency for pre cuts
  - results/sp_CUT/xxx_sp.dat

    file for applying normal cuts
  - plot/sp/obv.png
    - for each channel
  - plot_compare/sp/obv.png
    - for each sort

all channel



combined into sorts

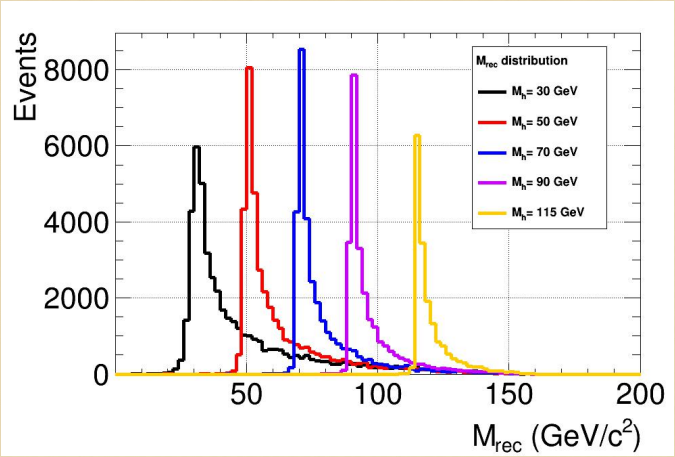


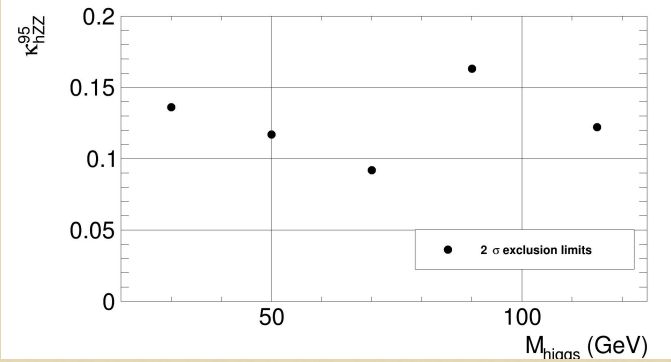
compare between different signal characters

after generate results for different signal characters

compare them by control/scan.dat

- signal distribution
- cut efficiency table
- sensitivity





advanced input example

Event

vim Event.dat

```
#show many first events want to be analyses
first_events_num    : -1

#which events want to analysse
special_events_num : -1

```

1. when first =-1, special=-1 => use all events
2. when first = n, special=-1 => use first n events
3. when first =-1, special= n => use the n-th event
4. when first =n1, special=n2 => use events from n1 to n2

where if n > total events, n = total events, and n1 < n2

Sensitivity

1. control/Sensitivity.dat
2. get data from root results
3. calculate sensitivities according to the control file.

rescale signal cross section

vim Sensitivity.dat

```
ExpCL              : 0.9545
sig_scale_switch   : true
sig_scale_factor   : 1000
sig_scale_pow      : 2
```

new scenario

1. control/scenario.dat
2. re-use results in Cuts_1 when changing the polarization. (MVA variable need to be recalculated.)
3. re-use results in Cuts_2 when changing the luminosity.

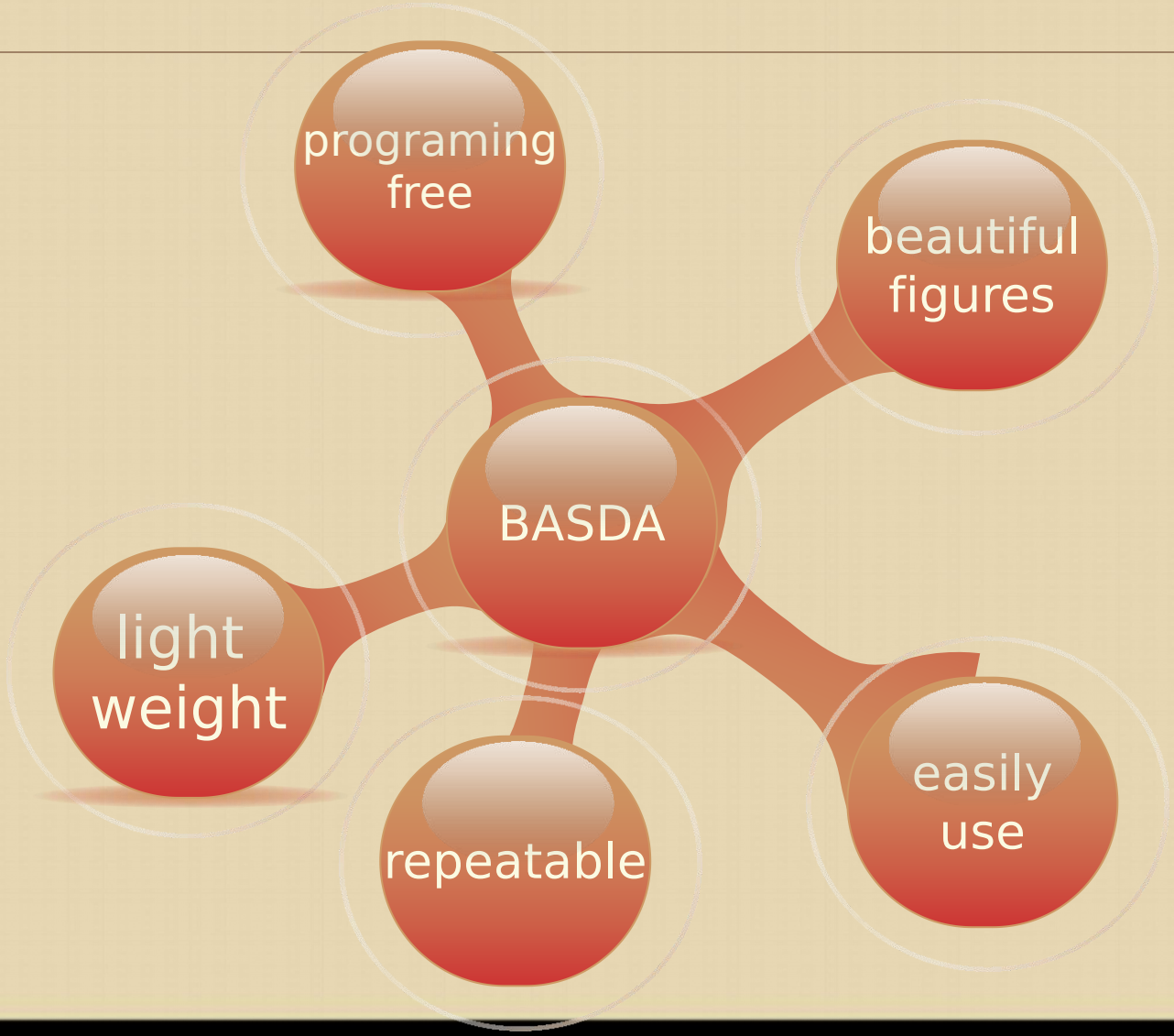
polarization value

luminosity
running polarization

vim Scenario.dat

```
elec_pol        : 0.8
posi_pol        : 0.3

Scenario        :
    scenario1:
        |  name          : running
        |  energy        : 250
        |  luminosity    : 2000
        |  (-,+)         : 0.45
        |  (+,-)         : 0.45
        |  (-,-)         : 0.05
        |  (+,+)         : 0.05
```

Conclusion


programing free
beautiful figures
BASDA
light weight
repeatable
easily use

Summary & Future

- https://github.com/YancyW/BASDA
  - four branches: master, release, develop, feature.
- provide more functions,
  - 2-D plots
  - statistic (like wsmaker,nplot ... in ATLAS group).
- GUI for basic users

Backup

Thank you!