# Table of Contents

- Data Preparation .............................................................................................................. 1
- Datasets (prefix /castor/cern.ch/user removed) ................................................................. 2
- L0 ......................................................................................................................................... 4
- Hlt1 ................................................................................................................................. 5
- TMVA ............................................................................................................................. 7
division

This page is to collect information regarding the bandwidth division for 2015 data taking. TOC:

- Data Preparation
- Datasets
- L0
- Hlt1
- TMVA
- Location of the samples at the Farm

Data Preparation

We started from a selection of the samples prepared for the physics WGs by Conor. The min bias sample was a separate production with Conditions: Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8 (500k mag up, 500k mag down). Applying the L0 to this data goes as follows.

1. All scripts are run on slc6, after SetupProject Moore v22r0, with gaudirun.py. To recreate my environment follow the steps in /afs/cern.ch/user/e/evh/w0/forAlvaro/README.

   ```python
   lhcb-proxy-init
   SetupProject LHCBDirac
   lhcb_bkk
   
   Find the LFN of the dataset, e.g.
   /lhcb/MC/Dev/MCFILTER.DST/00033990/0000/00033990_00000001_1.mcfilter.dst. Copy it to some afs workspace by using Thomas’ myDiracCopy.py:
   
   # SetupProject LHCBDirac
   # lhcb-proxy-init
   from DIRAC.Core.Base import Script
   Script.parseCommandLine(ignoreErrors = True )
   from DIRAC.Interfaces.API.Dirac import Dirac
   dirac = Dirac()
   import os
   if len(os.sys.argv) > 1 :
     lfn = os.sys.argv[1]
   else:
     # 2010 binc1 MC
     lfn = 'LFN:/lhcb/MC/2010/DST/00006385/0000/00006385_00000533_1.dst'
     lfn = 'LFN:/lhcb/data/2010/DIMUON.DST/00007045/0000/00007045_000000054_1.dimuon.dst'
     l_local_file = lfn.rfind('/')+1
     local_file = lfn[l_local_file:]
   
   result = dirac.getReplicas(lfn,printOutput=True)
   print result
   if result['OK'] :
     txt = result['Value']['Successful']
     dict = txt[lfn.replace('LFN:','')]
     dest1 = dict.keys()[0]
     srm = dict.values()[0]
     sc = os.system('lcg-cp '+srm+' '+local_file)
   
   print sc
   
   3. The datasets were reduced to 1k events (see /afs/cern.ch/user/e/evh/w0/bw/makesignalfiles.py):
   ```
#!/user/bin/env python
import GaudiPython
from GaudiConfig import *
from Gaudi.Configuration import *
from Configurables import LHCbApp
from LHCbConfig import *
from Configurables import InputCopyStream
InputCopyStream().Output = "DATAPATH='PFN:/afs/cern.ch/user/e/evh/updateL0/bw/21263002.dst' TYP='POOL_ROOTTREE' OPT='REC'"
LHCbApp.DataType = "2012"
LHCbApp.DDBBtag = 'head-20120413'
LHCbApp.CondDBtag = 'head-20120420'
from Configurables import EventClockSvc, CondDB
EventClockSvc().EventTimeDecoder = "OdinTimeDecoder"
CondDB(IgnoreHeartBeat = True)
appConf = ApplicationMgr (OutputLevel = INFO, AppName = 'readtest',OutStream=[InputCopyStream()])
appMgr = GaudiPython.AppMgr()

sel=appMgr.evtsel()

sel.open(['root://castorlhcb.cern.ch//castor/cern.ch/user/c/chaen/bw/21263002/00033552_00000004_1.mcfilter.dst'])
appMgr.algorithm('InputCopyStream').Enable=False

evt = appMgr.evtsvc()

while nwrite<1000 :
    appMgr.run(1)
    if evt['/Event/DAQ']==None :
        print 'No Event/DAQ found'
        continue
    appMgr.algorithm('InputCopyStream').execute()

3. Run the raw event juggler to move the raw event around for processing (see /afs/cern.ch/user/e/evh/w0/bw/runJuggler.py):

    from Gaudi.Configuration import *
    from Configurables import LHCbApp
    LHCbApp()
    from GaudiConf import IOHelper
    IOHelper().inputFiles([ 'rfio://castorlhcb.cern.ch//castor/cern.ch/user/e/evh/bwdivision/11102003.dst' ])
    from Configurables import GaudiSequencer, RawEventJuggler
    ApplicationMgr().TopAlg+=[GaudiSequencer("Spam")]
    Writer=InputCopyStream("MyStream")
    IOHelper().outStream("11102003-juggled.dst", writer=Writer)
    import RawEventCompat
    RawEventJuggler().Input=2.0
    RawEventJuggler().Output=0.0
    RawEventJuggler().Sequencer=GaudiSequencer("Spam")
    RawEventJuggler().WriterOptItemList=Writer
    RawEventJuggler().KillExtraNodes=True
    RawEventJuggler().KillExtraBanks=True
    RawEventJuggler().KillExtraDirectories=True

4. Run the L0App to apply the L0 to the data (see /afs/cern.ch/user/e/evh/w0/bw/runL0.py):

    #!/user/bin/env python
    from Gaudi.Configuration import *
    from Configurables import L0App
    L0App()
    L0App().Simulation=True
    L0App().TCK = '0x1810'
    L0App().outputFile='13774002-L0.dst'
    from GaudiConf import IOHelper
    IOHelper().inputFiles([ 'rfio://castorlhcb.cern.ch//castor/cern.ch/user/e/evh/bwdivision/11102003-juggled.dst' ])

Datasets (prefix /castor/cern.ch/user removed)

• BWDivisionDatasetsForDec14LHCbWeek
## Larger samples:

<table>
<thead>
<tr>
<th>Event type</th>
<th>Channel</th>
<th>Nb of events</th>
<th>Path L0 applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>11102003</td>
<td>B02Kpi</td>
<td>28169</td>
<td>e/evh/bwdivision/11102003/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-11102003-L0-28169evts.dst</td>
</tr>
<tr>
<td>11114001</td>
<td>B2KstMuMu</td>
<td>24149</td>
<td>e/evh/bwdivision/11114001/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-11114001-L0-24149evts.dst</td>
</tr>
<tr>
<td>11124001</td>
<td>Bd2Kstaree</td>
<td>25245</td>
<td>e/evh/bwdivision/11124001/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-11124001-L0-25245evts-L0.dst</td>
</tr>
<tr>
<td>13774002</td>
<td>Bs2Dsmuantinu</td>
<td>24723</td>
<td>e/evh/bwdivision/13774002/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-13774002-L0-24723evts.dst</td>
</tr>
<tr>
<td>12103035</td>
<td>Bplus2KKPi</td>
<td>24273</td>
<td>e/evh/bwdivision/12103035/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-12103035-L0-24273evts-L0.dst</td>
</tr>
<tr>
<td>12165106</td>
<td>Bplus2DK</td>
<td>25101</td>
<td>e/evh/bwdivision/12165106/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-12165106-L0-25101evts-L0.dst</td>
</tr>
<tr>
<td>13102201</td>
<td>Bsphigamma</td>
<td>24768</td>
<td>e/evh/bwdivision/13102201/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-13102201-L0-24768evts-L0.dst</td>
</tr>
<tr>
<td>13144001</td>
<td>Bs2JpsPhi</td>
<td>21207</td>
<td>e/evh/bwdivision/13144001/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-13144001-L0-21207evts-L0.dst</td>
</tr>
<tr>
<td>13264021</td>
<td>Bs2Dspi</td>
<td>21899</td>
<td>e/evh/bwdivision/13264021/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-13264021-L0-21899evts-L0.dst</td>
</tr>
<tr>
<td>11874004</td>
<td>B2Dmuantinu</td>
<td>22471</td>
<td>e/evh/bwdivision/11874004/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-11874004-L0-22471evts-L0.dst</td>
</tr>
<tr>
<td>21263002</td>
<td>D2KKpi</td>
<td>23568</td>
<td>e/evh/bwdivision/21263002/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-21263002-L0-23568evts-L0.dst</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>11497</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-11497evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>1247</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-1247evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>17806</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-17806evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>2019</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-2019evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>22121</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-22121evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>27043</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-27043evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>27181</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-27181evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>27319</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-27319evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>27360</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-27360evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>27360</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-27360evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>27360</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-27360evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>27360</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-27360evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>27360</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-27360evts.xdigi</td>
</tr>
<tr>
<td>30000000</td>
<td>min bias</td>
<td>27360</td>
<td>e/evh/bwdivision/30000000/Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8-30000000-L0-27360evts.xdigi</td>
</tr>
</tbody>
</table>

Datasets (prefix /castor/cern.ch/user removed)
L0

* Minbias and rate:
  o In MC we only write xings with at least one pp-int to tape (checked with plotting Gen/Collisions/size()!)
  In real data NoBias events, we write also the empty xings.
  o MC: to convert number of events to rate:
    - (1.-exp(-nu))*nr-bunches * 11.245 kHz.
    - 25 ns: nr-bunches is 2330 (Massi thinks ~2400), hence nu=1.5 gives 20.355 MHz
    - 50 ns: nr-bunches is 1296, hence with nu=2.7 gives 13.594 MHz
  o In real data NoBias events: rate=nr-bunches*11.245 kHz

* Scripts
  1. The steering program.
  2. The Minuit FCN.

* Plots
  o for the channels in the above table, i.e. minbias nu=1.5,
    (Beam6500GeV-RunII-MagUp-Nu1.5-25ns-Pythia8)
    made some plots as a function of the cut applied.

* Maximum Efficiencies obtained for the signal channels for L0 only:

<table>
<thead>
<tr>
<th>Event type</th>
<th>Channel</th>
<th>Nb of events</th>
<th>Max Eff (L0 only)</th>
<th>Throttle</th>
</tr>
</thead>
<tbody>
<tr>
<td>11102003</td>
<td>B02Kpi</td>
<td>28169</td>
<td>0.38</td>
<td>0.097</td>
</tr>
<tr>
<td>11114001</td>
<td>B2KstMuMu</td>
<td>24149</td>
<td>0.86</td>
<td>0.036</td>
</tr>
<tr>
<td>11124001</td>
<td>Bd2Kstaree</td>
<td>25245</td>
<td>0.56</td>
<td>0.092</td>
</tr>
<tr>
<td>13774002</td>
<td>Bs2Dsmuantinu</td>
<td>24723</td>
<td>0.62</td>
<td>0.072</td>
</tr>
</tbody>
</table>
see attached slides for the optimised efficiencies and rates

Efficiencies' table for various scenarios.

<table>
<thead>
<tr>
<th>Event type</th>
<th>Channel</th>
<th>max eff</th>
<th>eff/(max eff)</th>
<th>eff/(max eff)</th>
<th>eff/(max eff)</th>
<th>eff/(max eff)</th>
<th>eff/(max eff)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no CPU limit</td>
<td>CPU limit</td>
<td>4X w(Kstee)</td>
<td>4X w(charm)</td>
<td>adding</td>
<td>phigamma</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13112001</td>
<td>Bs2MuMu</td>
<td>0.92</td>
<td>0.86</td>
<td>0.87</td>
<td>0.79</td>
<td>0.80</td>
<td>0.90</td>
</tr>
<tr>
<td>13144001</td>
<td>Bs2JpsPhi</td>
<td>0.87</td>
<td>0.79</td>
<td>0.80</td>
<td>0.67</td>
<td>0.64</td>
<td>0.83</td>
</tr>
<tr>
<td>11114001</td>
<td>B2KstMuMu</td>
<td>0.86</td>
<td>0.78</td>
<td>0.79</td>
<td>0.66</td>
<td>0.64</td>
<td>0.83</td>
</tr>
<tr>
<td>11874004</td>
<td>B2Dmuantinu</td>
<td>0.66</td>
<td>0.66</td>
<td>0.67</td>
<td>0.44</td>
<td>0.42</td>
<td>0.67</td>
</tr>
<tr>
<td>13774002</td>
<td>Bs2Dsmuantinu</td>
<td>0.62</td>
<td>0.61</td>
<td>0.61</td>
<td>0.37</td>
<td>0.35</td>
<td>0.61</td>
</tr>
<tr>
<td>11124001</td>
<td>Bd2Kstaree</td>
<td>0.56</td>
<td>0.34</td>
<td>0.43</td>
<td>0.80</td>
<td>0.25</td>
<td>0.50</td>
</tr>
<tr>
<td>11102003</td>
<td>B02Kpi</td>
<td>0.38</td>
<td>0.76</td>
<td>0.71</td>
<td>0.53</td>
<td>0.84</td>
<td>0.61</td>
</tr>
<tr>
<td>12103035</td>
<td>Bplus2KKPi</td>
<td>0.34</td>
<td>0.73</td>
<td>0.71</td>
<td>0.50</td>
<td>0.82</td>
<td>0.59</td>
</tr>
<tr>
<td>13264021</td>
<td>B2Dspsi</td>
<td>0.32</td>
<td>0.75</td>
<td>0.72</td>
<td>0.50</td>
<td>0.84</td>
<td>0.59</td>
</tr>
<tr>
<td>12165106</td>
<td>Bplus2DK</td>
<td>0.29</td>
<td>0.72</td>
<td>0.72</td>
<td>0.48</td>
<td>0.83</td>
<td>0.57</td>
</tr>
<tr>
<td>21263002</td>
<td>D2KKpi</td>
<td>0.18</td>
<td>0.67</td>
<td>0.61</td>
<td>0.33</td>
<td>0.72</td>
<td>0.44</td>
</tr>
<tr>
<td>13102201</td>
<td>Bsphigamma</td>
<td>0.60</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.48</td>
</tr>
</tbody>
</table>

- Thresholds for the above options:

<table>
<thead>
<tr>
<th>L0Channel</th>
<th>eff/(max eff)</th>
<th>eff/(max eff)</th>
<th>eff/(max eff)</th>
<th>eff/(max eff)</th>
<th>adding</th>
</tr>
</thead>
<tbody>
<tr>
<td>L0Electron</td>
<td>200</td>
<td>170</td>
<td>117</td>
<td>240</td>
<td>170</td>
</tr>
<tr>
<td>L0Photon</td>
<td>200</td>
<td>170</td>
<td>117</td>
<td>240</td>
<td>190</td>
</tr>
<tr>
<td>L0Hadron</td>
<td>210</td>
<td>210</td>
<td>250</td>
<td>208</td>
<td>238</td>
</tr>
<tr>
<td>L0Muon</td>
<td>60</td>
<td>60</td>
<td>112</td>
<td>116</td>
<td>67</td>
</tr>
<tr>
<td>L0DiMuon</td>
<td>950</td>
<td>940</td>
<td>1440</td>
<td>1970</td>
<td>665</td>
</tr>
<tr>
<td>SPD mu,had,ele</td>
<td>480</td>
<td>530</td>
<td>570</td>
<td>540</td>
<td></td>
</tr>
<tr>
<td>SPD dimu (fixed)</td>
<td>900</td>
<td>900</td>
<td>900</td>
<td>900</td>
<td></td>
</tr>
</tbody>
</table>

http://dijkstra.web.cern.ch/dijkstra/BW-15/compare.ps

**Hlt1**

- Modifying the thresholds of Hlt1 lines at runtime

Vanya has modified the HltUnit code to allow for changable parameters (in the svn head of Phys/LoKiTrigger) via a dictionary Params. For instance introduce a parameter for PtMin in the Hlt1TrackAllL0 line as follows:
def hlt1TrackNonMuon_Streamer( self, name, props ) :
    from Hlt1Lines.Hlt1GECs import Hlt1GECUnit
    from Configurables import LoKi__HltUnit as HltUnit
    props['name'] = name
    props['forward'] = 'LooseForward' if name.find('Photon') > -1 else 'TightForward'
    if props['ValidateTT'] :
        props['forward'] = "ValidateWithTT >>" + props['forward']
    Params= { 'PtMin': props.get('PT',0) }
    lineCode = ""
    VeloCandidates
    >> ( ( TrIDC('isVelo') > %(Velo_NHits)s ) &
        ( TrVELOMISS < %(Velo_Qcut)s ) &
        ( Tr_HLTMIP ( 'PV3D' ) > %(IP)s * mm )
    >> (monitor( TC_SIZE > 0, '# pass VeloQ/IP', LoKi.Monitoring.ContextSvc ) )
    >> tree( monitor( TC_SIZE , 'nVeloIP', LoKi.Monitoring.ContextSvc ) )
    >> %(forward)s
    >> ( (TrTNORMIDC > %(TrNTHits)s ) &
        ( TrPT > PARAM('Params[PtMin]') ) &
        ( TrP > %(P)s * MeV )
    >> tree( monitor( TC_SIZE > 0, '# pass P/PT', LoKi.Monitoring.ContextSvc ) )
    >> tree( monitor( TC_SIZE , 'nP', LoKi.Monitoring.ContextSvc ) )
    >> FitTrack
    >> tree( monitor( TC_SIZE > 0, '# pass TrackFit', LoKi.Monitoring.ContextSvc ) )
    >> tree( monitor( TC_SIZE , 'nFit', LoKi.Monitoring.ContextSvc ) )
    >> ( ( TrCHI2PDOF < %(TrChi2)s ) &
        ( Tr_HLTMIPCHI2 ( 'PV3D' ) > %(IPChi2)s )
    >> tree( monitor( TC_SIZE > 0, '# pass TrackChi2/IPChi2', LoKi.Monitoring.ContextSvc )
    >> tree( monitor( TC_SIZE , 'nChi2', LoKi.Monitoring.ContextSvc ) )
    >> SINK( 'Hlt1%(name)sDecision' )
    >> ~TC_EMPTY
    """ % props
    hlt1TrackNonMuon_Unit = HltUnit( 'Hlt1'+name+'Unit',
                                      Preambulo = self.hlt1Track_Preambulo( name ),
                                      Params= Params,
                                      Code = lineCode )
    from HltTracking.HltPVs import PV3D
    return [ Hlt1GECUnit( 'Loose' ), PV3D(), hlt1TrackNonMuon_Unit ]

and then change it at runtime as follows:

#!/user/bin/env python
from Gaudi.Configuration import *
from Configurables import Moore
from LHCbKernel.Configuration import *
from Gaudi.Configuration import *
from Configurables import EventClockSvc, CondDB
EventClockSvc().EventTimeDecoder = "OdinTimeDecoder"
CondDB(IgnoreHeartBeat = True)

Moore().UseTCK = True
Moore().InitialTCK = '0x007b0044'
Moore().CheckOdin = False
Moore().EvtMax = 10
Moore().DDDBtag = "dddb-20120831"
Moore().CondDBtag = "cond-20120831"
Moore().Simulation = False
Moore().DataType = '2012'
Moore().Split = "Hlt1"
Moore().WriterRequires = ["Hlt1"]
Moore().inputFiles = [ 'rfio://castorlhcb.cern.ch//castor/cern.ch/user/e/evh/117770/117770_0x0044_NB_L0Phys_00.raw' ]
print Moore()
from Configurables import EventSelector
EventSelector().PrintFreq = 1
import GaudiPython
gaudi = GaudiPython.AppMgr(outputlevel = 3)
gaudi.initialize()

def setupEvent(gaudi, Loop):
    from GaudiKernel.SystemOfUnits import MeV
    alg_list = gaudi.algorithms()
    for a in alg_list:
        if a == "Hlt1TrackAllL0Unit":
            line = gaudi.algorithm("Hlt1TrackAllL0Unit")
            props = line.properties()
            for p in props:
                value = props[p].value()
                print "Property Name/Value: 's'/s " % (p, value)
                line.OutputLevel = 2
                if Loop == 1:
                    line.Params = {'PtMin': 5000.0 * MeV}
                props = line.properties()
                for p in props:
                    value = props[p].value()
                    print "After changing. Property Name/Value: 's'/s " % (p, value)
                    if Loop == 2:
                        line.Params = {'PtMin': 500.0 * MeV}
                    props = line.properties()
                    for p in props:
                        value = props[p].value()
                        print "After changing. Property Name/Value: 's'/s " % (p, value)
                line.OutputLevel = 3

import gaudigadgets
nEvents = 10
Loop = 1
while (Loop < 3):
    if Loop == 2: gaudigadgets.panorewind()
    setupEvent(gaudi, Loop)
    for i in range(0, nEvents):
        gaudi.run(1)
        print "event number ", i, " Loop ", Loop
    Loop = Loop + 1

These scripts plus instructions on how to get the right environment can be found in
/afs/cern.ch/user/e/evh/w0/forAlvaro. #TMVA

TMVA

- Setting a discrete range of variables for the TMVA Genetic algorithm is done as follows:

    std::vector<TMVA::Interval*> parameterRanges;
    parameterRanges.push_back(new TMVA::Interval(.5, 1., 6));  //means: Interval(.5, 1., 6) = .5, .6, ...

-- EricvanHerwijnen - 31 Mar 2014

This topic: LHCb > BWDivision
Topic revision: r28 - 2015-08-25 - ConorFitzpatrick

Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback