Shit just GoAT serious...

Cristina Collicott

March 2014

The following manual describes analysis using the GoAT (Generation of Analysis Trees) framework. This software was designed to provide a fast, tree-based analysis framework for the A2-Collaboration. The code uses AcquRoot to produce generic analysis trees (using a physics class named TA2GoAT). These generic trees are then processed using GoAT software where particle reconstruction, data checks, and data sorting is completed.

1 Where can I get the software?

The most up-to-date software is available via the A2-Collabotation-dev github repository.

Additions to Acqu: https://github.com/A2-Collaboration-dev/acqu
Branch name: A2GoAT

GoAT software: https://github.com/A2-Collaboration-dev/a2GoAT
Branch name: development

As a disclaimer, the code is in development stage. If you find errors (or improvements), please share this with others (pull request via git).

2 Analysis Procedures

2.1 AcquRoot - generate raw data trees

Run AcquRoot using TA2GoAT as the physics class. By default, a generic tree file will be produced with the name ”Acqu_rawfilename.dat”. As an example, a raw file named CB_100.dat will produce a tree file named ”Acqu_CB_100.root”.

The physics class TA2GoAT inherits from an SQL access class (designed to link to CaLib database). If you need to specify different apparatus classes or detector classes, you change change them inside this class. However, the code is currently designed to take advantage of the best available apparatus classes (so I would recommend using the defaults).

2.2 GoAT - generate analysis trees

GoAT provides the framework to analyse these generic raw data trees from AcquRoot. A set of tree managers allows the user to sidestep most of the difficulties of working with trees. A single global config-file is used to setup all the reconstruction and sorting choices of the user. Specification of input and output files is best done via command line flags.

Running from the command line
A set of command line flags is available

- **-f** Input file name
- **-d** Input directory (runs all files within a directory - default name is ”Acqu_XXX.root”)
- **-p** Change default input prefix (-d flag will search for files ”prefix_XXX.root”)
- **-F** Output file name (if unspecified, default name is ”GoAT_XXX.root”)
- **-d** Output directory for GoAT trees
- **-p** Change default output prefix (default output name will be ”prefix_XXX.root”)

When a flag is given on command line, the next input is associated with this flag (ex. -f inputfile.root). The config-file should be given on the command line too. No flag needs to be used, just specify it at some point in the command line.

Example command:

```
./build/bin/goat -d $acqu/output/ -D /home/cristina/GoATTrees configfiles/GoAT-example.dat
```

This command would run *goat* on all root files within the directory $acqu/output/ which match ”Acqu_XXX.root”. The analysed files will be output to the folder /home/cristina/GoATTrees and will be named ”GoAT_XXX.root”. The sorting conditions and particle reconstruction choices will be taken from the config-file ”configfiles/GoAT-example.dat”.

Another Example command:

```
./build/bin/goat -d $acqu/output/ -p RawTrees -P GoATTrees configfiles/GoAT-example.dat
```

This command would run *goat* on all root files within the directory $acqu/output/ which match ”RawTrees_XXX.root”. The analysed files will be output to the folder $acqu/output/ (same as input) and will be named ”GoATTrees_XXX.root”. The sorting conditions and particle reconstruction choices will be taken from the config-file ”configfiles/GoAT-example.dat”.

Controlling the global config-file
The analysis using GoAT can be controlled using a config-file.

- **Period-Macro**
  Update to terminal after N events analysed

- **CheckCBStability**
  Turn on/off data checks for missing CB sections in data

- **Do-Particle-Reconstruction**
  Turn on particle reconstruction? If turned off, raw particle tree is saved.

- **Do-Charged-Particle-Reconstruction**
  Turn on charged particle reconstruction?

- **Cut-dE-E-CB-Proton**
  Provide cutfile and cutname (similar cut available for π^+ and e$^-$, TAPS)

- **Do-Meson-Reconstruction**
  Turn on meson reconstruction?

- **Cut-IM-Width-Pi0:**
  Set an IM width (acts as weighting factor and IM cut) for π^0
  Similar cut available for η and η'.

- **SortRaw-NParticles**
  Sort on raw # of particles (total, CB, and TAPS).
  Use the flags +, -, and = to specify "or more", "or less", or "exactly".

- **SortRaw-CBEnergySum**
  Sort on CB energy sum, use +,-,= flag

- **Sort-NParticles**
  Sort on reconstruction # of particles (total), use +,-,= flag

- **Sort-Particle**
  Turn on this sort multiple times. Specify a particle (by name or PDG code, or charged/neutral), the number of particles (use +,-,= flag) and within which θ region.

  ex. Sort-Particle: pi0 1= 20 180 (require exactly 1 π^0 within $\theta = (20°,180°)$
  ex. Sort-Particle: 2212 1+ 0 180 (require 1 or more proton within $\theta = (0°,180°)$
  ex. Sort-Particle: neutral 3- 0 90 (require 3 or less neutral particles within $\theta = (0°,90°)$