**aj\_rtools.cpp – macro for analysis of X-Array data**

This text aims to provide a simplified explanation for aj\_rtools.cpp. This is very much a building process. If you find any errors, please contact the author (aj.mitchell@anu.edu.au).

* Using aj\_rtools.cpp

aj\_rtools is written for use in ROOT. Preamble includes defining:

* Loading aj\_rtools.cpp

In a terminal, open root → root -l

***Load the program → .L /location of file/aj\_rtools.cpp***

Storing the file in the root/macros directory will make this process easier.

* Loading a root file

**void** dload(**const** Char\_t \*fn, **const** Char\_t \*options="READ")

Root files will have been generated from the sorting process.

***dload(“/location of root file/name.root”)***

.ls will show the histograms within the root file.

* **Plotting a histograms**

**void** d2d(**const** Char\_t \*matn, Int\_t n=1)

*HistogramName->Draw() e.g. CEhiBeta->Draw()*

e.g. CEhiBeta->Draw(), or d2d(“CEhiBeta”)

* **Projections:**

HistogramName->ProjectionX() or ProjectionY()

HistogramName\_px->Draw() or HistogramName\_py->Draw()

e.g. CEhiBeta->ProjectionX(); CEhiBeta\_px->Draw() will graw the b-gamma energy spectrum

* **Drawing a decay curve**

e.g. EvsTape->Draw(); EvsTape->ProjectionY()

EvsTape is a 2D matrix. Decay curve obtained from projection onto timing axis (Y-axis).

* **Gating on a gamma ray within a gamma-gamma (GG)/beta-gg (BGG) matrix**

Float\_t GX(**const** Char\_t \*matn,Float\_t peak,Float\_t width=0.0,Int\_t cy=0)

GG matrix → GGE

BGG matrix → GGEB

It is possible to gate on a single gamma ray, e.g. in GG matrix, by using the following command:

GX(“GGE”, Egam, 3, 1)

182.0 → The energy of the gated gamma ray

3.0 → FWHM of the peak

1.0 → Unsure but if set to zero, aj\_rtools crashes.

This command performs the following:

Generates both x- and y-projection spectra of the 2D matrix (gg\_px and gg\_py)

Determines an appropriate background of these spectra (gg\_bgx and gg\_bgy)

Gates on the gamma ray

Subtracts the background

Draws the resulting spectrum

* **Reading a matrix**

**void** readmat(**const** Char\_t \*filename)

A matrix can be read into aj\_rtools using the following command:

readmat(“name”)

* **Writing a matrix (for export to Radware for example)**

**void** writemat(**const** Char\_t \*filename)

A matrix can be written into a different format using the following command:

writemat(“name”)

The extension of the written matrix is changeable within the code. It is currently defaulted to .m4b.

The dimensions of the matrix are set by NCHX and NCHY.

* **Writing a spectrum (for export to Radware for example)**

**void** writespe(**const** Char\_t \*hisname, Char\_t \*spename)\

1D spectra may be written to .spe format using the writespe command:

writespe(“name to be written”,”name of written file”)

The default extension is .spe but this is changeable.

The dimensions of the spectrum are set by NCHX or NCHY.

* **Reading a spectrum**

**void readspe(Char\_t \*histname,Char\_t \*filename,Char\_t \*xy="X")**

A matrix can be read into aj\_rtools using the following command:

readspe(“name”)

* **Taking a projection of a 2D GG or BGG matrix**

**void** dx(**const** Char\_t \*matn, Int\_t w)This command will project the GG matrix onto the x-axis. It then finds an appropriate background, before subtracting this and plotting the remaining total projection.

*dx(“GGE”)*

A similar function exists for a Y-Projection dy.

* **Take projection of a 2D matrix**

**Int\_t pjx(Char\_t \*matn, Int\_t w=45)**

Takes a projection \_px and determines a background \_bgx. Does not plot anything.

*pjx(“GGE”)*

A similar function exists for a Y-Projection pjy.