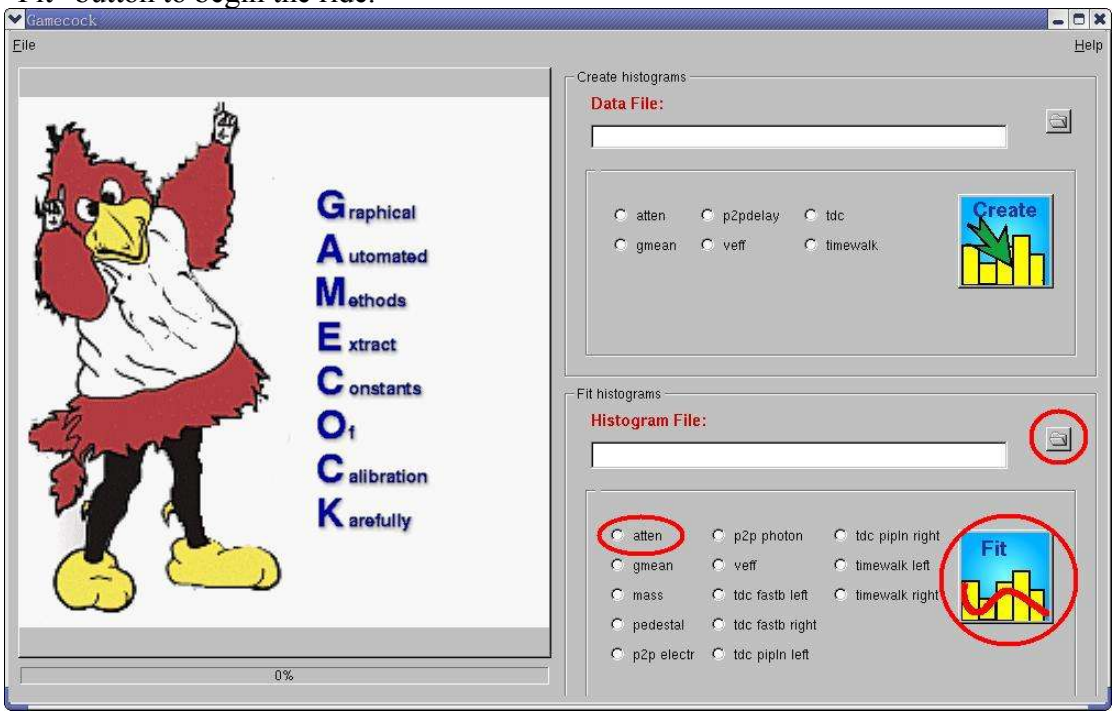
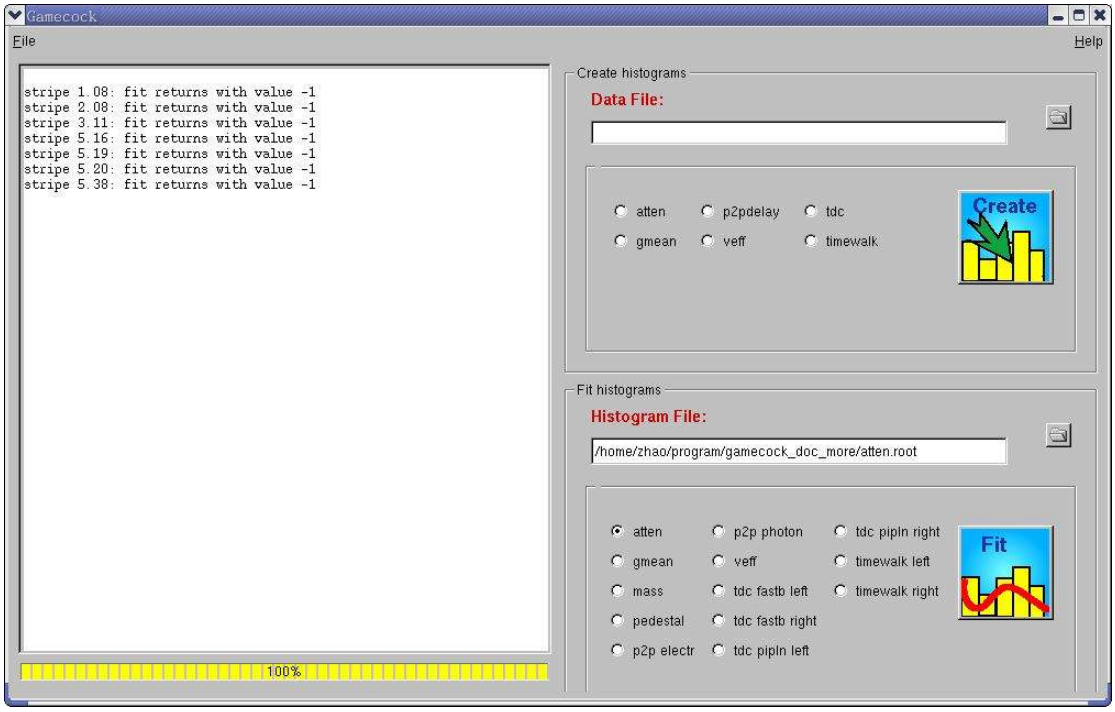
Tutorial: Atten

After open the main window of Gamecock. we can choose the data file by clicking the open file picture button in the “Fit histograms” section or inputing the file name directly. Leaving the “Histogram File” entry blank will make Gamecock use the default file “./root/atten.root”. (If there's no such file, it will report an error in a pop-up window). The next step is choose the radio button "atten”. Then we can click the big “Fit” button to begin the ride.

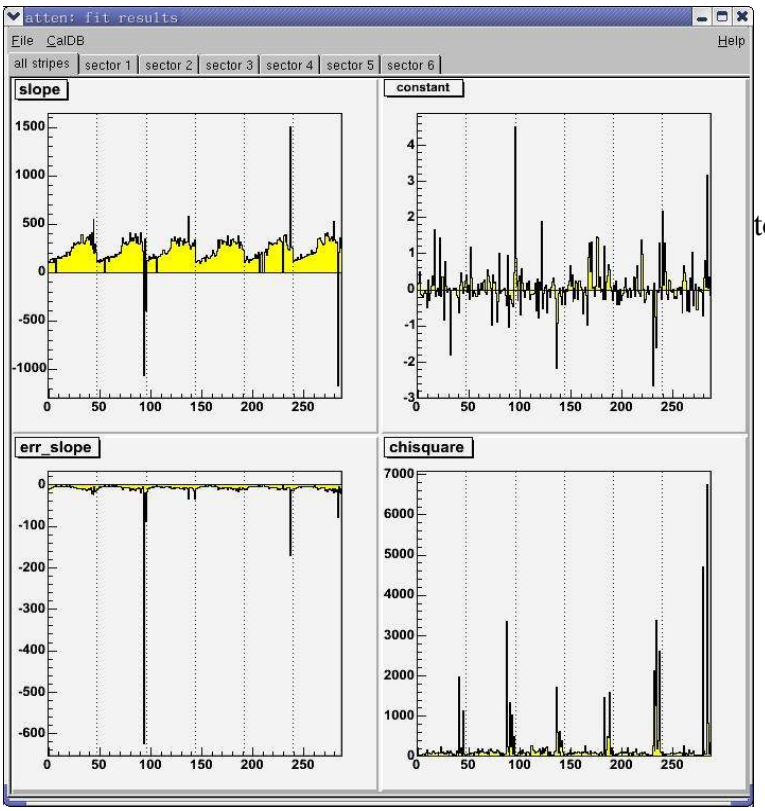


When you see the processing bar moving, you should also see the problematic channels showing in the log window. It reads like sector 1, channel 08 having no data. Here value “-1” means no data (or a dead channel) and value “4” means no successful



fit which can be generally caused by too few data entries (here we don't see it, but we will saw some those cases in gmean calibration tutorial).

However, there are more channels than those automatically detected by Gamecock who may not have good fit with the default fitting parameters and require us to identify them and tweak the parameters to make them better.

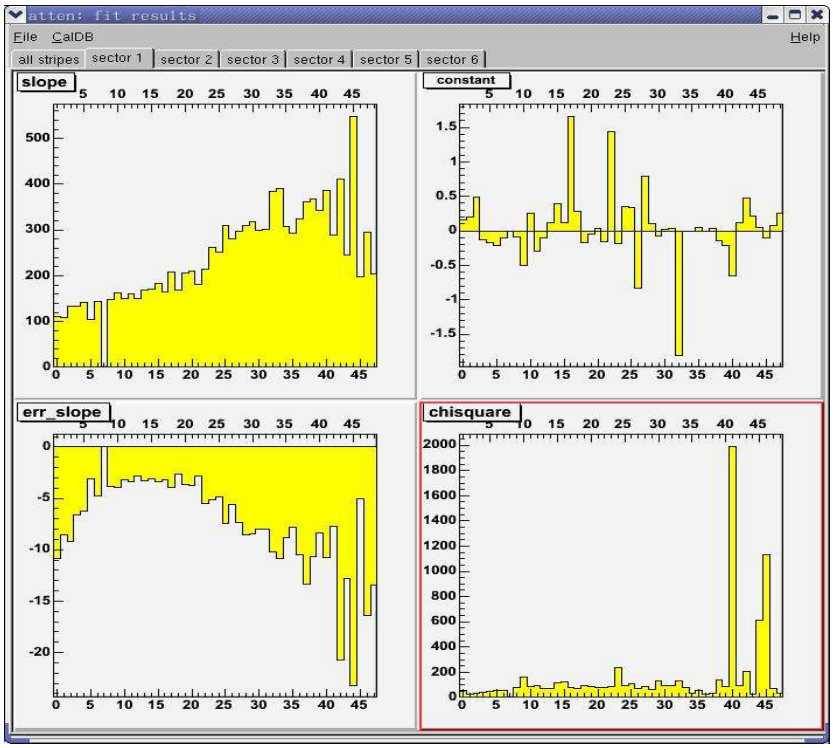


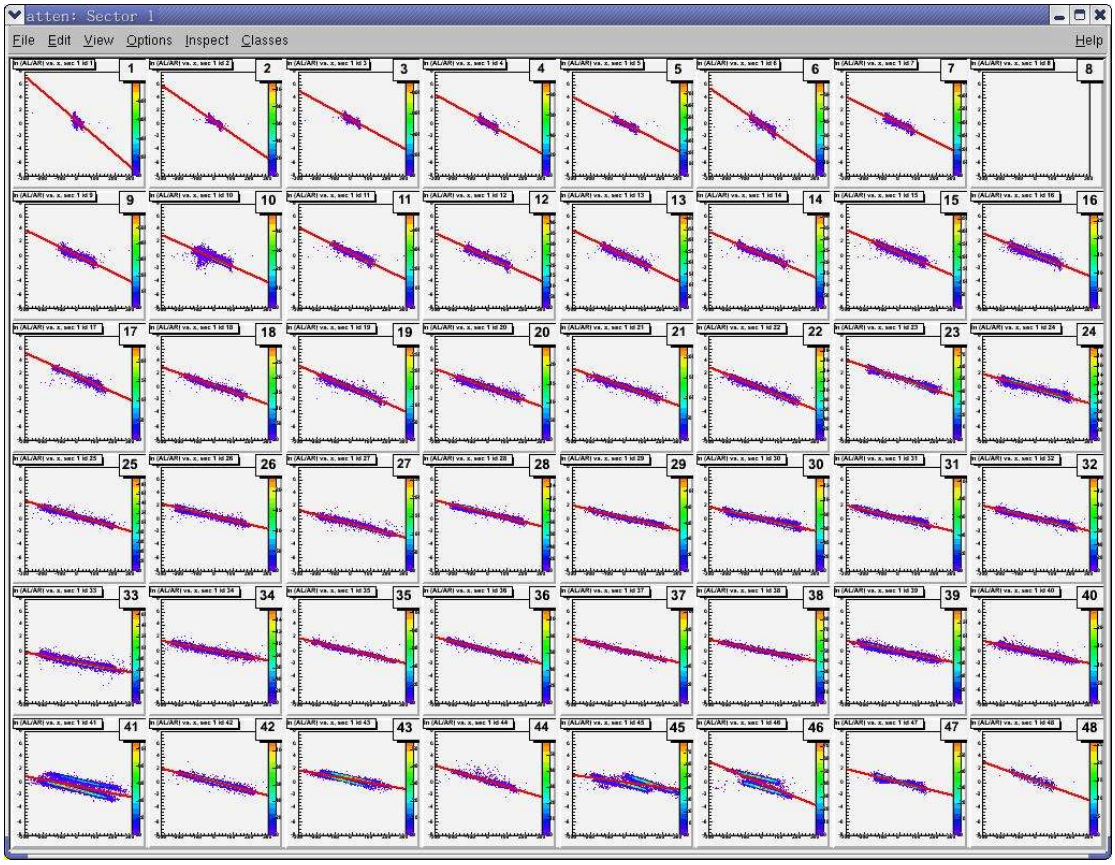
Therefore, after the auto-fitting process done, two new windows will pop up.

one called “fit results” shows the fit results (for “atten” calibration, they are slope, constant, error of slope, chisquare).

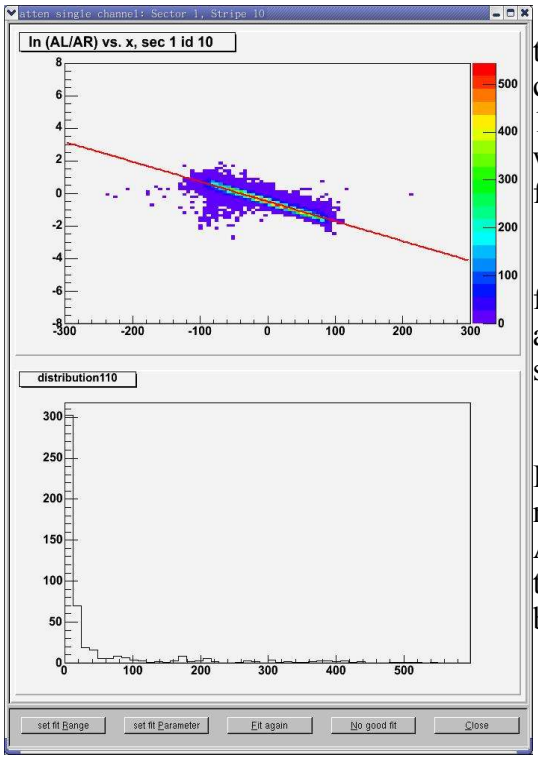
The other one with thumbnails shows the histograms and their fits for one sector (in next page).

By clicking on the tabs, we can check the fitting for all channels in different sectors and the thumbnail window will switch to the current sector. Let's go to Sector 1 first.





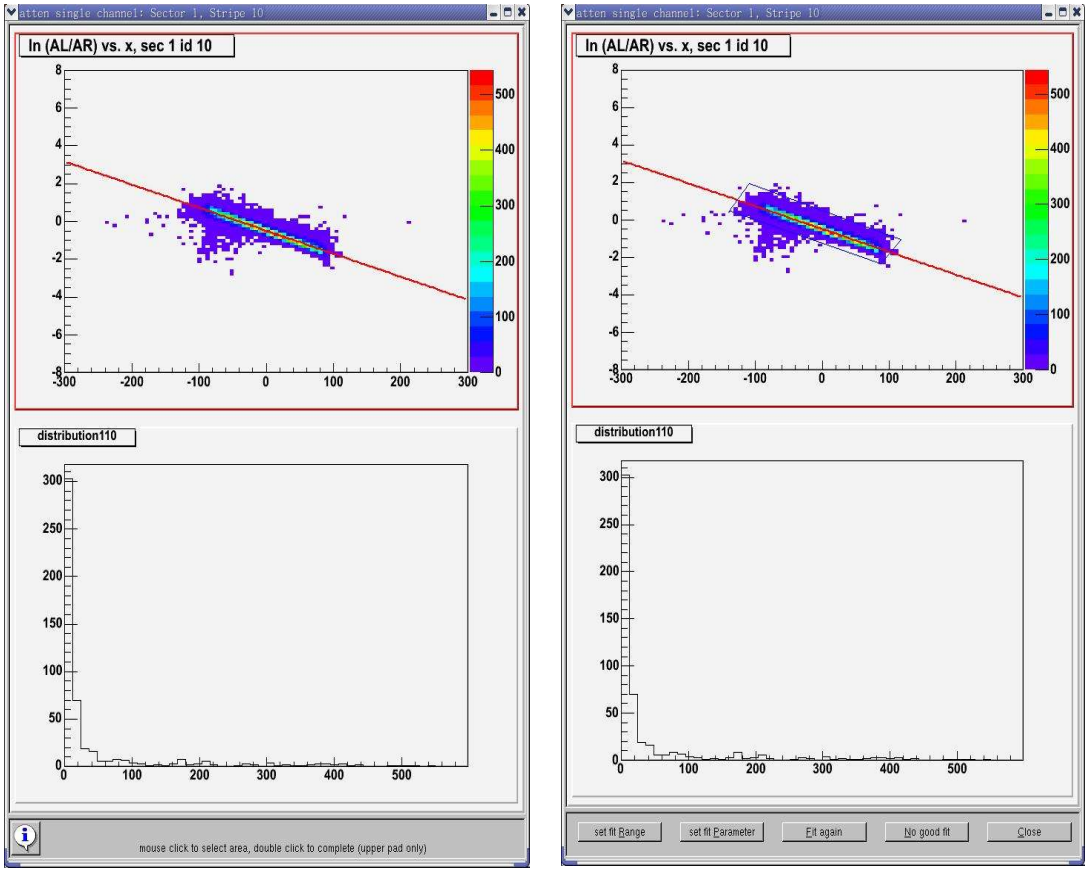
The “atten” histograms are 2D and we try to fit them by straight lines. For some stripes, the widespread background may have bad influences on the fitting. We know that Stripe 40 – 48 have coupled PMTs so that they always give trouble (sometimes double peaks or double lines). So we pay more attention to other channels.



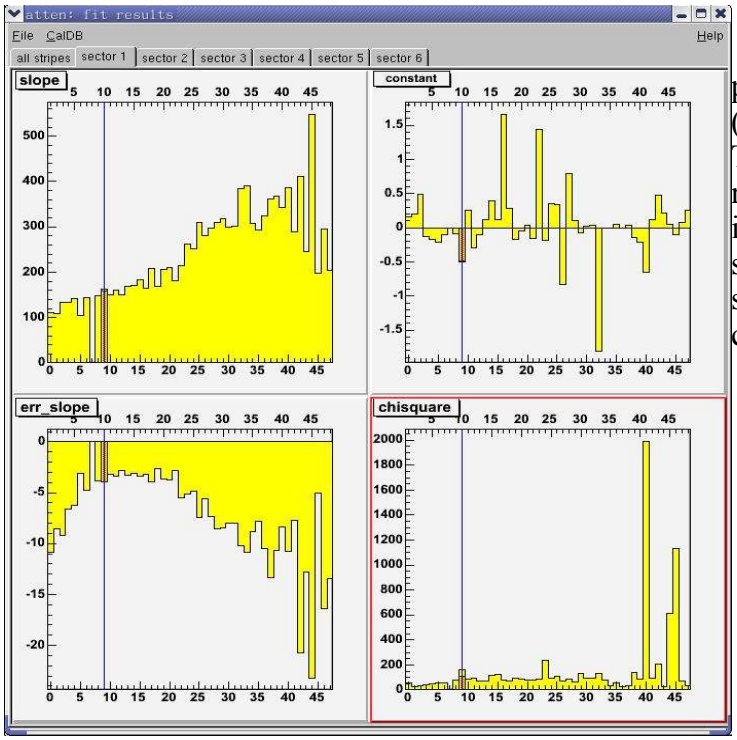
From the “fit results” window, we can see the Stripe 10 and 24 are having a little bigger chisqure. As an example, Let's look at Stripe 10. By clicking on its thumbnail, a new window will pop up to show the bigger figure for this channel.

In the “single channel” window, the upper figure shows the 2D histogram and fitting and the lower one shows the data at the cross section y=0.

The background of Stripe 10 is spread. It should be helpful that limiting the fit in a new range instead of all the data points. After clicking on the “set fit Range” button, the window will change like the left one below.



Just follow the hint at the bottom, *n* left mouse single clicks and finish with left mouse double click in the upper figure will set up a box to include a new fit range. In our case, we use 4 clicks. After the final doucle click, the window will look like the right one showing the new fit range in a blue box. Then click on “Fit again” we will get a new fit. Even though the difference between the old and new fits can't be easily seen from the fitting histogram, It shows clearly in the “fit results” window.



The current Stripe 10 is pointed out by a blue line (referred to the upper index). The new fit result shows in red. We can see its chisqure improved by setting a smaller fit range, while the slope and constant don't change much.

The other way to improve fitting is to reset the fit parameters. For “atten” calibration, it's not very useful because it's simply a linear fit. We will show an example in “gmean” calibration.

Sometimes, there's simply no way to have a good fit. Then we can click the “No good fit” and follow the instruction in “Tutorial: How to deal with channel without a reasonable data distribution”.

By repeating these steps to look through all channels in all sectors, we are ready to push the constants into database. Ref: “Tutorial: Check calibration constants into the Caldb (MySQL) database”.