

Performing Simple Binned Fits

The following is a set of java examples that illustrate how to perform relatively simple fits. These examples can be compiled and run as standalone programs using JAIDA (see the JAIDA [Release Notes](#)) or they can be loaded, compiled and run within [JAS3](#). Just copy and paste the desired example in a file that has the same name as the class file.

Gaussian Fit To An Histogram

Chi2GaussianFit.java

```
import hep.aida.*;
import java.util.Random;

public class Chi2GaussianFit {

    public static void main(String[] argv) {

        // Create the AIDA factories
        IAnalysisFactory analysisFactory = IAnalysisFactory.create();
        ITree tree = analysisFactory.createTreeFactory().create();
        IHistogramFactory histogramFactory = analysisFactory.createHistogramFactory(tree);
        IFunctionFactory functionFactory = analysisFactory.createFunctionFactory(tree);
        IFitFactory fitFactory = analysisFactory.createFitFactory();

        // Create a random number generator
        Random r = new Random();

        // Create a 1D histogram. Fill it with a set of random gaussian distributed data
        IHistogram1D hl = histogramFactory.createHistogram1D("Histogram 1D",50,-6,6);

        for (int i=0; i<100000; i++)
            hl.fill(r.nextGaussian());

        // Create a gaussian and set its parameters
        IFunction gauss = functionFactory.createFunctionByName("gauss","g");
        gauss.setParameter("amplitude",hl.maxBinHeight());
        gauss.setParameter("mean",hl.mean());
        gauss.setParameter("sigma",hl.rms());

        // Create the fitter, with chi squared as fit method and Minuit as fit engine
        IFitter fitter = fitFactory.createFitter("chi2","minuit");

        // Fit the histogram with the gaussian function
        IFitResult result = fitter.fit(hl,gauss);

        // Create a plotter and display the result of the fit
        IPlotter plotter = analysisFactory.createPlotterFactory().create("Plotter");
        plotter.region(0).plot(hl);
        plotter.region(0).plot(result.fittedFunction());
        plotter.show();
    }
}
```

Fit To A Data Point Set