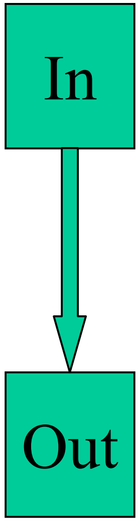
Confidence Levels with ROOT

Tlimit is a ROOT add-on.

It computes limits using the Likelihood ratio method using the method originaly implemented by Tom Junk in fortran77.

The classes are



In

Out

- TLimitDataSource

It takes the signal, background and data histograms to form a channel. More channels can be added using AddChannel(), as well as different systematic sources.

- TLimit

It is the actual algorithm. Is takes a TLimitDataSource as input and run a set of MC experiments in order to compute the limits. If needed, the inputs (s_i and b_i) are fluctuated within their systematics. The output is a TConfidenceLevel

- TConfidenceLevel

It is the final result of the algorithm. It is created just after the time-consuming part and can be stored in a TFile for further processing. It contains light methods to return CLs, CLb and other interesting quantities.

TLimitDataSource class

Main methods are:

- TLimitDataSource();

  default constructor
- TLimitDataSource(TH1F* s,TH1F* b,TH1F* d);

  usefull constructor, creates an analysis channel with signal, background and data histograms
- virtual void AddChannel(TH1F*,TH1F*,TH1F*);

  adds a channel
- virtual void AddChannel(TH1F*,TH1F*,TH1F*,TH1F*, TH1F*, TObjArray*);

  adds a channel with systematics. The 3 last arguments are:

  TH1F* error on the signal

  TH1F* error on the background

  1 bin = 1 error source for that channel (relative error)

  TObjArray* name of the error sources

  errors with the same name are 100% correlated.
- virtual void SetOwner(bool swtch=kTRUE);

  sets the TLimitDataSource owner of the histograms.

  They will be deleted with the TLimitDataSource

Tlimit class

```
TConfidenceLevel* TLimit::ComputeLimit
                    (TLimitDataSource* data,
                    Int_t nmc, TRandom* generator,
                    Double_t (*statistic)(Double_t, Double_t, Double_t) stat)
```

- data is the input TLimitDataSource.
- nmc is the number of MC experiments to produce
- generator is the MC generator used. Default is TRandom3 (Mersenne Twister)
- stat is the function used as statistic. Default is TLimit::LogLikelihood

TConfidenceLevel class

Object returned by Tlimit::Compute().
Contains all the information to provide limits.
Interesting public methods are:

```
Double_t GetStatistic() const ;
Double_t GetExpectedStatistic_b(Int_t sigma = 0) const;
Double_t GetExpectedStatistic_sb(Int_t sigma = 0) const;
Double_t CLb(bool use_sMC = kFALSE) const;
Double_t CLsb(bool use_sMC = kFALSE) const;
Double_t CLs(bool use_sMC = kFALSE) const;
Double_t GetExpectedCLb_sb(Int_t sigma = 0) const;
Double_t GetExpectedCLb_b(Int_t sigma = 0) const;
Double_t GetExpectedCLsb_b(Int_t sigma = 0) const;
Double_t GetExpectedCLs_b(Int_t sigma = 0) const ;
Double_t GetAverageCLs() const;
Double_t GetAverageCLsb() const;
Double_t Get3sProbability() const;
Double_t Get5sProbability() const;
Int_t    GetDtot() const ;
Double_t GetStot() const ;
Double_t GetBtot() const ;
```

It can be stored as is in a ROOT file.

Installing and using it.

As for all ROOT add-ons:

The simplest way to use those classes in an interactive ROOTCINT session is to include the headers in your files and to just call:

```
.L TLimitDataSource.cpp+
.L TConfidenceLevel.cpp+
.L TLimit.cpp+
```

at the beginning of the session.

Then, supposing that there is a plotfile.root file containing 3 histograms (signal, background and data), you can imagine doing things like:

```
TFile* infile=new TFile("plotfile.root","READ");
infile->cd();
TH1F* sh=(TH1F*)infile->Get("signal");
TH1F* bh=(TH1F*)infile->Get("background");
TH1F* dh=(TH1F*)infile->Get("data");
TLimitDataSource* mydatasource = new TLimitDataSource(sh,bh,dh);
TConfidenceLevel *myconfidence = TLimit::ComputeLimit(mydatasource,50000);
cout << "CLs    : " << CLs() << endl;
cout << "CLsb   : " << CLsb() << endl;
cout << "CLb    : " << CLb() << endl;
cout << "< CLs >  : " << GetExpectedCLs_b() << endl;
cout << "< CLsb > : " << GetExpectedCLsb_b() << endl;
cout << "< CLb >  : " << GetExpectedCLb_b() << endl;
delete myconfidence;
delete mydatasource;
infile->Close();
```

Documentation

Over the method:

- HEP-EX/9902006
- Tom Junk 's page :

  http://thomasj.home.cern.ch/thomasj/searchlimits/ecl.html

Over the implementation:

- this presentation
- the ALPHA++ web-site « tools »link:

  http://cern.ch/aleph-proj-alphapp/doc/tlimit.html