粒子物理与核物理实验中的数据分析

陈少敏
清华大学

第六讲：PAW与ROOT在数据分析中的应用

本讲要点

- PAW 与 ROOT 简介
- PAW 与 ROOT 的数据结构
- PAW 与 ROOT 的图形运算
- PAW 与 ROOT 上的随机抽样

物理分析工作平台(PAW)

PAW 是英文 “Physics Analysis Workstation” 的缩写。

诞生于1986年，最后版本更新于2002年。

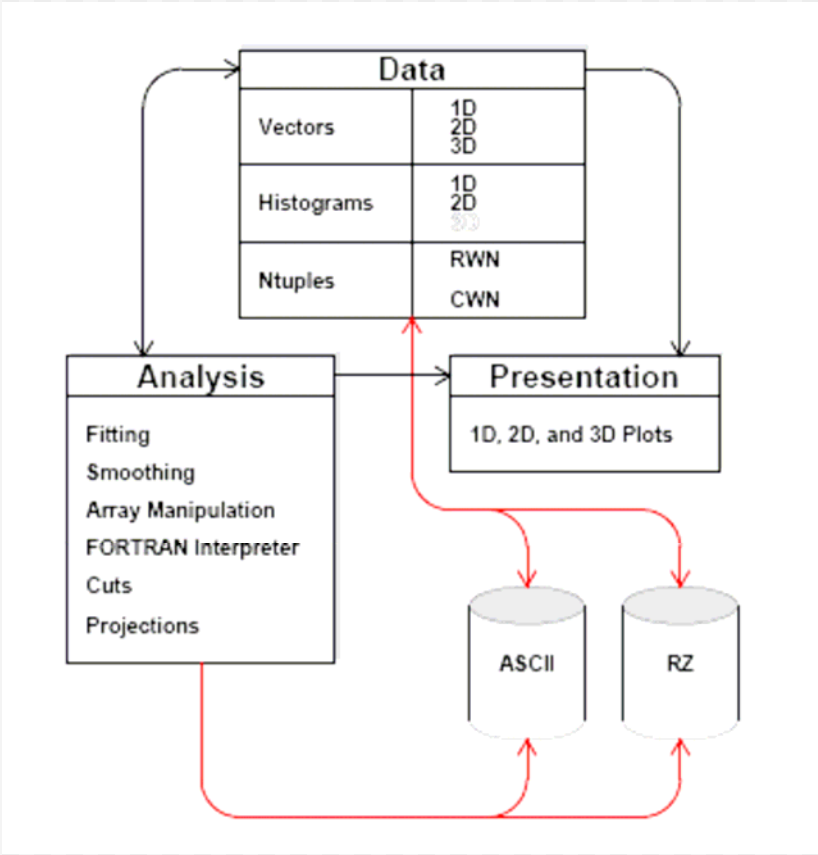
版权归西欧核子研究中心(CERN)所有，免费提供给与CERN有关的科学实验或与CERN有科学合作的协议方免费使用。

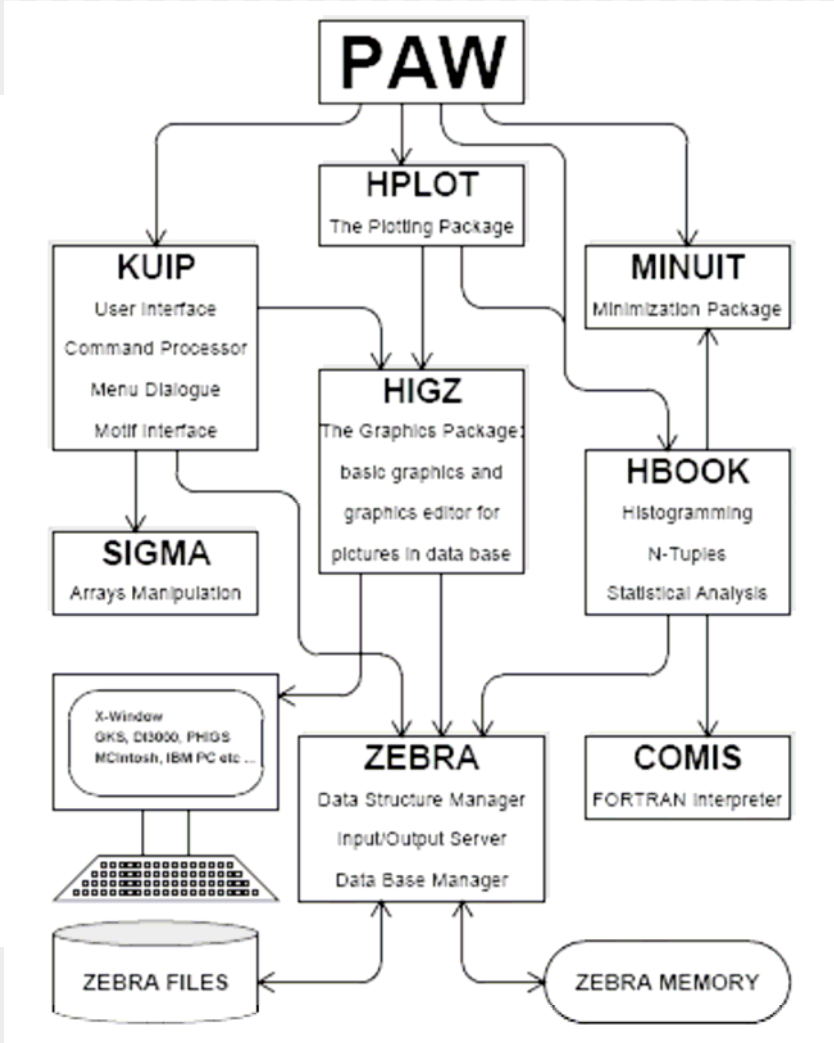
广泛应用于2002年以前进行的实验。2002年以后新建的实验项目大都采用基于C++的ROOT软件包。

在PAW与ROOT软件包之间有PAW++。所有三个软件包均由CERN计算机中心开发与维护。

CERN提供了在不同操作系统运行的免费下载网址：
http://cernlib.web.cern.ch/cernlib/version.html

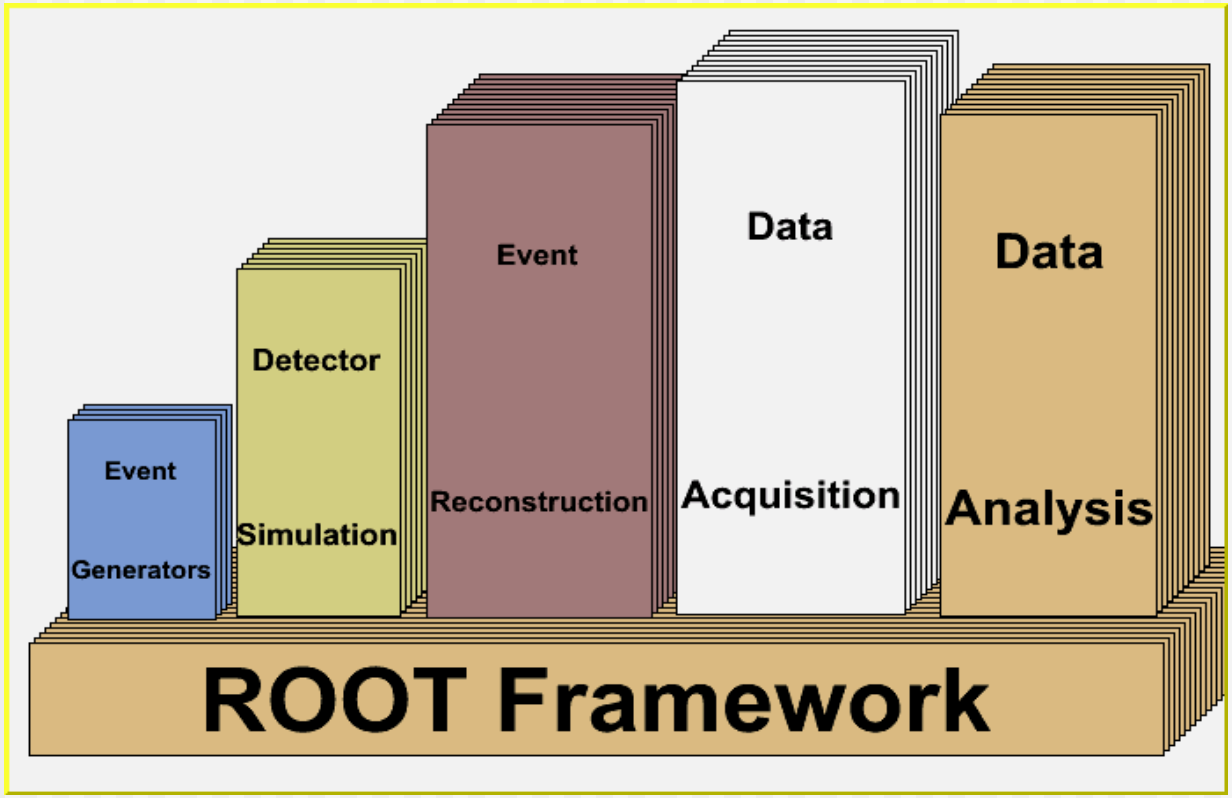
PAW的主要结构


Data
Vectors
1D
2D
3D
Histograms
1D
2D
Ntuples
RWN
CWN
Analysis
Fitting
Smoothing
Array Manipulation
FORTRAN Interpreter
Cuts
Projections
Presentation
1D, 2D, and 3D Plots
ASCII
RZ


PAW
HPLOT
The Plotting Package
KUIP
User Interface
Command Processor
Menu Dialogue
Motif Interface
MINUIT
Minimization Package
HIGZ
The Graphics Package:
basic graphics and
graphics editor for
pictures in data base
HBOOK
Histogramming
N-Tuples
Statistical Analysis
SIGMA
Arrays Manipulation
X-Window
GKS, DI3000, PHIGS
MCIntosh, IBM PC etc ...
ZEBRA
Data Structure Manager
Input/Output Server
Data Base Manager
COMIS
FORTRAN Interpreter
ZEBRA FILES
ZEBRA MEMORY

新一代的工作平台ROOT

ROOT充分利用了计算机技术的最新发展，并且能适应现代粒子物理实验对极大数量的数据分析以及模拟要求。在这一点上PAW已经达到了极限。



CERN提供了在不同操作系统运行的免费下载网址：http://root.cern.ch/

PAW 与 ROOT

用户界面：PAW ≈ ROOT

图形功能：PAW ≈ ROOT

同等数据处理时间：PAW 快于 ROOT

同等数据存储空间：PAW 大于 ROOT

2414133 run_615026.dat	58197330 run_615531.dat
1581056 run_615026.ntuple	内存不够
901766 run_615026.root	17552250 run_615531.root

我个人使用PAW与ROOT的感觉: 对于单纯的数据处理, **PAW**使用起来相对容易一些。对于非常复杂的数据结构和多次计算，**ROOT**有它的优点，尤其是在处理数据量在10^{12}以上的应用。在很多指令操作上，两者之间有不少一一对应的关系。

数据库Ntuple

“Ntuple”： 是一个加有多种功能的N-维数据库。数据在 Ntuple 中按行或列排列，而各行或列有 N 个数据块。例如，在分析练习中说给的数据

…

事例号 光电倍增管号码 时间测量值 电荷测量值

…

就可以形成这样能被 PAW 或者 ROOT 识别的数据库。包含在这两种物理分析工作平台的各种统计分析工具就可以得到充分的利用。

为了优化数据结构，可以在

PAW: CWN, RWN,BLOCK…

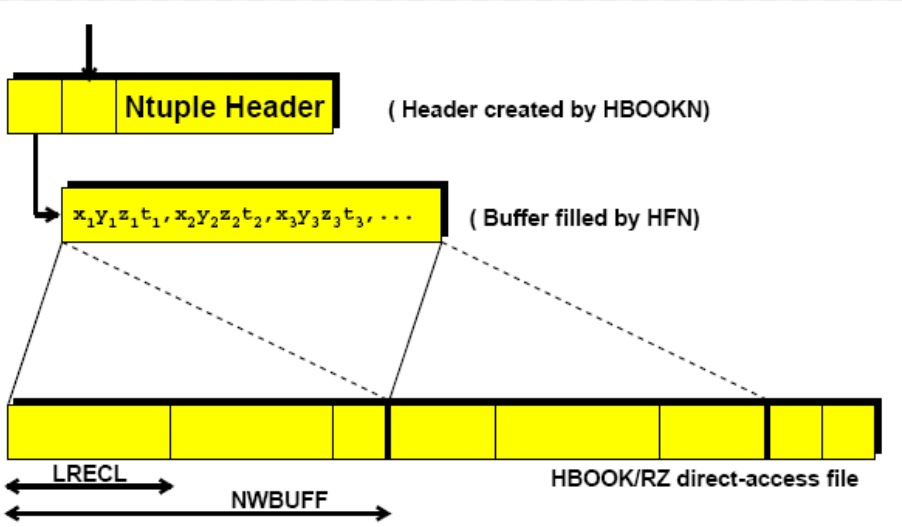
ROOT: TREE, BRANCH…

以加速数据处理的时间。

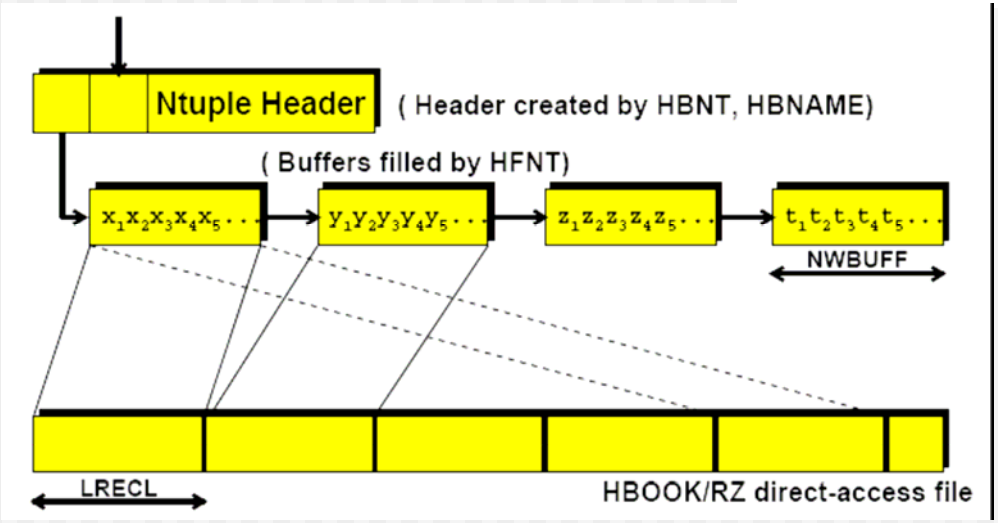
PAW数据中的行排列与列排列

假设每个事例有观测量(*x*,*y*,*z*,*t*)

按行排列



用于观测量较少的情况。如果内存大，数据处理时间较短。



按列排列

用于观测量较多的情况。该排列方式不适用于本课程的练习。

ROOT数据中的TREE结构

步骤:

1. 建立一个 TFile

   TFile *hfile = new TFile("AFile.root","RECREATE");

2. 建立一个 Ttree

   TTree *tree = new TTree("myTree","A ROOT tree");

   TTree aliTree("aliTree", "/aliroot");

3. 把 TBranch 加到 Ttree

   Event *event = new Event();
   myTree->Branch("EventBranch","Event",&event);

4. 填入 tree

   myTree->Fill();

5. 写到一个文件上

   hfile->Write();

Tree = H1F, H2F, Ntuple 等等。

如何生成Ntuple数据库

```
      subroutine readin
      integer nevent,pmtid
      real time,charge,vect(4)
      parameter (nwpawc=50000,ndim=4)
      character*15 ntit(ndim)
      data ntit/'nevent','pmtid','time','charge'/
      common/pawc/h(nwpawc)
      call hbookn(100,'Test',ndim,' ',1024,ntit)
      open(61,file='run_615026.dat')
1     read(61,*,end=10)nevent,pmtid,time,charge
      vect(1)=nevent
      vect(2)=pmtid
      vect(3)=time
      vect(4)=charge
      call hfn(100,vect)
      go to 1
10    close(61)
      return
      end
```

在ROOT平台

```
{
  gROOT->Reset();
#include "Riostream.h"
  ifstream in;
  in.open("run_615026.dat");
  Float_t nevent,pmtid,time,charge;
  TFile *f =new TFile("run_615026.root","RECREATE");
  TNtuple *ntuple = new TNtuple("ntuple",
"data from ascii file","nevent:pmtid:time:charge");
  while (1) {
    in >> nevent >> pmtid >> time >> charge;
    if (!in.good()) break;
    ntuple->Fill(nevent,pmtid,time,charge);
  }
  in.close();
  f->Write();
}
```

root –b –q readin.C+

PAW>call readin.f

PAW>h/fil 1 run_615026.ntuple ! N;cd //lun1;hrout 0;close 1

从PAW到ROOT: **h2root run_615026.ntuple run_615026.root**

PAW数据文件中的内容

PAW>h/fil 1 run_615026.ntuple 0
PAW>scan 100

Event	nevent	pmtid	time	charge
1	1.	13.	17.2407	0.984996
2	1.	80.	26.8364	1.97757
3	1.	101.	20.2386	1.38074
4	1.	142.	28.0259	1.43446
5	1.	256.	18.9224	0.53552
6	1.	294.	22.6816	1.38957
7	1.	485.	-169.929	0.967207
8	1.	543.	44.1801	1.26147
9	1.	587.	14.3102	1.01928

More...? (<CR>/N/G)

ROOT数据文件中的内容

```
root [0] TFile f("run_615026.root");
root [1] ntuple.Scan();
```

```
*************************************************
*Row   *  nevent *   pmtid *       time *     charge *
*************************************************
*      0 *       1 *      13 * 17.240699 * 0.9849960 *
*      1 *       1 *      80 * 26.836399 * 1.9775700 *
*      2 *       1 *     101 * 20.238599 * 1.3807400 *
*      3 *       1 *     142 * 28.025899 * 1.4344600 *
*      4 *       1 *     256 * 18.922399 * 0.5355200 *
*      5 *       1 *     294 * 22.681600 *   1.38957 *
*      6 *       1 *     485 * -169.9290 *  0.9672070 *
*      7 *       1 *     543 * 44.180099 * 1.2614699 *
*      8 *       1 *     587 * 14.310199 * 1.0192799 *
*      9 *       1 *     607 * 4.7850999 * 1.5807800 *
Type <CR> to continue or q to quit ==>
```

分析数据中的基本PAW指令

PAW>h/fil 1 run_615026.ntuple 0 读入数据文件
PAW>zone 2 2; 在终端显示 2x2 幅图
PAW>n/pl 100.nevent; n/pl 100.pmtid; n/pl 100.time; n/pl 100.charge 画图
PAW>n/pl 100.charge%time pmtid.eq.1 第一根管 t vs. q 的二维散点图
PAW>n/pl 100.charge%time%pmtid 画三维散点图
PAW>h/cr 10 'time (ns)' 100 –50 50 开出一维直方图框架
PAW>n/proj 10 100.time pmtid.eq.1 填第一根管的时间分布直方图
PAW>h/cr 20 't vs. q' 100 –50 50 100 –10 90 开出二维图框架
PAW>n/proj 20 100.charge%time pmtid.eq.1 填第一根管t vs. q的二维图
PAW>zone 1 2; h/pl 10; h/pl 20 画图
PAW>slix 20 1; h/proj 20; h/pl 20.slix.1 画出在x轴的边缘分布图
PAW>ve/cr ipmt(680); ve/cr x(680); ve/cr y(680); ve/cr z(680) 开设数组
PAW>ve/read ipmt,x,y,z /home/chensm/geom/geom.dat 读入几何数据
PAW>ve/pl y%x 画出 x vs. y 的二维散点图
PAW>picture/print file_name.gif 把当前图形生成 gif 文件

分析数据中的基本ROOT指令

```
root[0]TFile f("run_615026.root")
root[1]TCanvas *c=new Tcanvas("c","plots",0,0,600,600);c.Divide(2,2)
root[2]c.cd(1);ntuple.Draw("pmtid");c.cd(2);ntuple.Draw("time");
root[3]c.Clear();c.cd(1);ntuple.Draw("charge:time", "pmtid==1");
root[4]c.Clear();c.cd(1);ntuple.Draw("charge%time%pmtid");
root[5]TH1F *h10=new TH1F("h10","time (ns)",100,-50,50);
root[6]ntuple.Draw("time>>h10", "pmtid==1");
root[7]TH2F *h20=new TH2F("h20","t vs. q",100 –50,50,100,–10,90);
root[8]ntuple.Draw("charge:time>>h20","pmtid==1");
root[9]c.Divide(1,2);c.cd(1);h10.Draw();c.cd(2);h20.Draw();
root[10]h20.ProjectionX("Xproj");Xproj.Draw();
root[11]h10.Draw("e");
root[12]h20.Draw("box");
root[12]h20.Draw("lego2");
root[14]c.SaveAs("file_name.gif");
```

画出误差$\sqrt{n}$

画出灰度

将二维散点图表示成三维图

把当前图形生成 gif 文件

Macros 文件

Macro 文件包含了一串PAW或ROOT的指令行

例如，在PAW平台中，一个叫“check.kumac”的Macro文件可以是

```
h/file 1 run_615026.ntuple 0
n/pl 100.charge
picture/print charge.gif
```

显然，它把如何做出一张图的PAW指令保留下来，以便日后可以重复你的结果。

通过“PAW>exec check.kumac”，来执行Macro文件中的指令。

Macro文件中还可以包含子Macro文件，

在ROOT平台中，也有一个叫“check.C”的Macro文件，如

```
{
TFile f(“run_615026.root”);
ntuple.Draw(“charge”);
c1.SaveAs(“charge.gif”);
}
```

把如何做出一张图的ROOT指令保留下来，以便日后可以重复你的结果。

一维直方图归一化

在PAW环境下

```
PAW>norm id1 1
PAW>norm id2 1
PAW>…
PAW>h/pl id1; h/pl id2 s;…
```

常用于比较两种分布，找出区别。

在ROOT环境下

```
root[0]id1.Scale(1);
root[1]id2.Scale(1);
root[2]…
root[?]id1.Draw(); id2.Draw("same") ;…
```

一维直方图之间的运算

相加

```
PAW>h/op/add id1 id2 id3 a b
root>TH1F *id3=new TH1F(*id1);
root>id3.Add(id1,id2,a,b);
```

常用于相同实验的数据叠加，增加统计量。

相减

```
PAW>h/op/sub id1 id2 id3 a b e
root>TH1F *id3=new TH1F(*id1);
root>id3.Sumw2();
root>id3.Add(id1,id2,a,-b);
```

常用于从实验测量的分布中，扣除本底得到纯信号的分布。

误差： $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2} = \sqrt{n_1 + n_2}$

一维直方图之间的运算(续)

相除

```
PAW>h/op/div id1 id2 id3 a b E
root>TH1F *id3=new TH1F(*id1);
root>id3.Sumw2();
root>id3.Divide(id1,id2,a,b);
root>id3.Divide(id1,id2,a,b,"B");
```

常用于效率的计算。

$$\sigma = \frac{n_1}{n_2}\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}$$

$$\sigma = \sqrt{\frac{\frac{n_1}{n_2}\left(1-\frac{n_1}{n_2}\right)}{n_2}}$$

相乘

```
PAW>h/op/multi id1 id2 id3 a b
root>TH1F *id3=new TH1F(*id1);
root>id3.Sumw2();
root>id3.Multiply(id1,id2,a,b);
```

常用于对分布进行诸如效率等的修正。

$$\sigma = n_1 n_2\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}$$

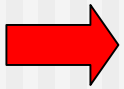
一维直方图之间运算的误差

虽然PAW与ROOT都提供了一维直方图的运算功能，但对结果的误差一定要仔细检查。很多情况下，用户需要从图中读出频数值与误差并确认运算无误。

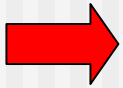
二维散点图的应用

- 用于寻找两个随机变量之间的关联

```
PAW>n/pl 100.time%charge
Root[0]ntuple.Draw("rime:charge");
```



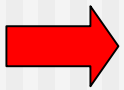
找出在时间测量或残差与电荷大小是否有关



大量应用于各种定性分析

- 用来表示关联矩阵

```
call hfill(100,xmean,ymean,E[(xmean-x)*(ymean-y)])
h100->Fill(xmean,ymean,E[(xmean-x)*(ymean-y)]);
```



每个格子的灰度大小表明了关联的强弱

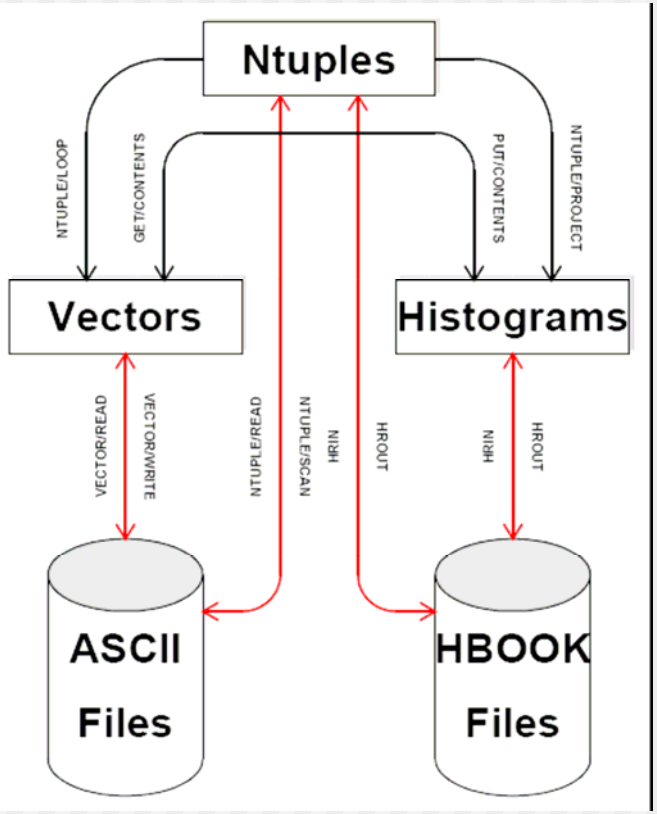
PAW中数据的输入输出

在PAW的环境下有两种。对于数据量较少(<5000)，例如可以采用

PAW>ve/cr nevent(5000); ve/cr pmtid(5000);ve/cr t(5000); ve/cr q(5000)
PAW>ve/read nevent,pmtid,t,q /home/chensm/data/run_615026.dat
PAW>h/cr/1d 100 'PMT ID" 680 0.5 680.5
PAW>ve/put_vect/content 100 pmtid
…

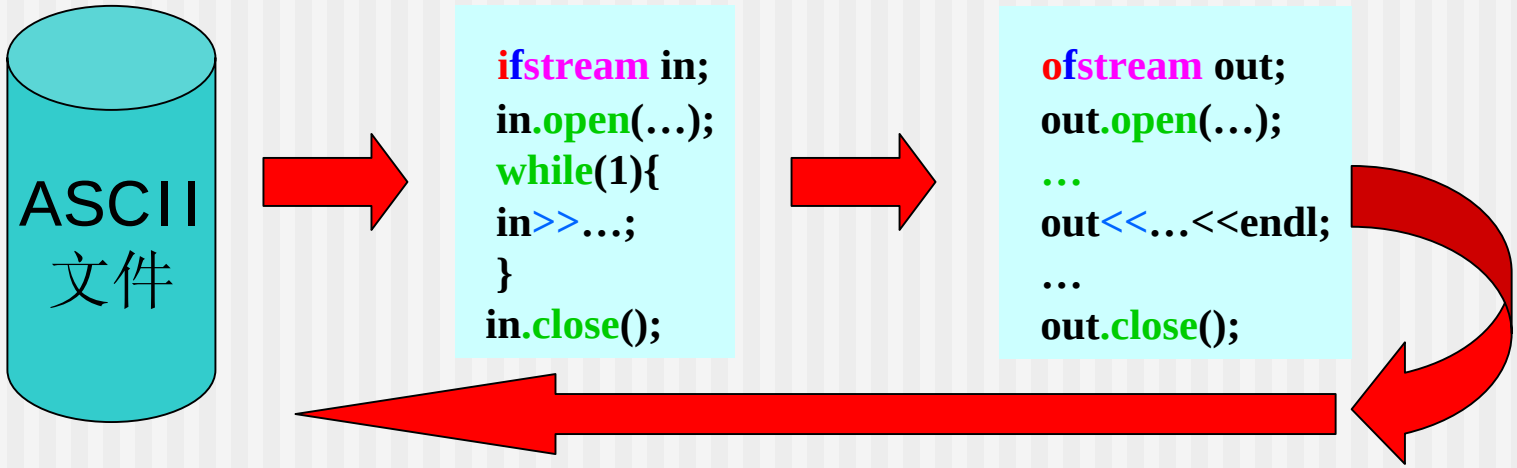
可以用"PAW>help vector"
查看有关vector的指令

对于输入数据量较大的情况，
可采用调用FORTRAN子程序
的形式，生成PAW环境下默
认的数据格式，如"ntuple"格式。



ROOT中数据的输入输出

在ROOT的环境下，可以采用



还可以采用输出到一个新的TREE或BRANCH的方法。

在PAW上的随机抽样

高斯(正态)分布

```
subroutine gauss
real rvec(1),sigma,mean
data sigma,mean/2.0,1.0/
call hbook1(10,'x',100,-10.,10.,0.)
do i=1,1000
   call rnorml(rvec,1)
   x=rvec(1)*sigma+mean
   call hfill(10,x,0,1.0)
end do
return
end
```

如果换为rndm(rvec,1)，将产生在[mean,sigma+mean]的均匀分布。如果需要产生其它分布，需要把MATHLIB相关函数连接上

产生平均值为mean标准偏差为sigma的高斯分布。

```
PAW>call gauss.f
PAW>h/pl 10
```

在ROOT上的随机抽样

高斯(正态)分布

```
{  gROOT->Reset();
  hx = new TH1F("hx","x dis.",100,-10,10)
  gRandom->SetSeed();
  Double_t x;
  const Double_t sigma=2.0;
  const Double_t mean=1.0;
  const Int_t kUPDATE = 1000;
  for ( Int_t i=0; i<1000; i++) {
     x=gRandom->Gaus(mean,sigma);
     hx->Fill(x); }
}
```

可以换为

```
x = gRandom->Rndm(i);
x = gRandom->Landau(mean,sigma);
x = gRandom->Binomial(ntot,prob);
x = gRandom->Poisson(mean);
x = gRandom->Exp(tau);
x = gRandom->BreitWigner (me,sig);
```

产生平均值为mean标准偏差为sigma的高斯分布。

对PAW与ROOT的一些评论

对于要参加一个已经运行多年的实验，比如超级神冈中微子实验，需要学习PAW的使用。对于邀参加即将进行的实验，比如LHCb，BES等，需要学习ROOT的使用。

对于只是完成本门课程的同学，建议只学习ROOT。

本次讲座仅涉及一些在数据分析中常见的操作，很多方面的应用，包括神经网络，拟合等。大家可以参见相关的使用手册。这里我只做一些抛砖引玉的介绍。

小结

- PAW 与 ROOT 简介

  PAW基于FORTRAN，ROOT基于C++

- PAW 与 ROOT 的数据结构

  PAW采用CWN与RWN，ROOT采用tree，branch

- PAW 与 ROOT 的图形运算

  一维图之间的加减乘除，注意误差的计算；二维图的定性分析

- PAW 与 ROOT 上的随机抽样

  可以对各种分布的随机抽样

习题

习题6.1:考虑一随机变量 x 具有期待值 μ 和方差 σ^2，并假设我们得到一样本，它包含有 n 次测量 $x_1,\ldots,x_n$。为了证明样本的平均值 $\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i$ 是一个符合期待值μ 的估计量。

(a) 首先证明Chebyshev不等式 $P(|x-\mu|\geq a)\leq\frac{\sigma^2}{a^2}$

只要 x 的方差存在，对任何正定的 a 均成立。这一点可利用方差的定义

$$\sigma^2=\int_{-\infty}^{+\infty}(x-\mu)^2 f(x)dx$$

这里 $f(x)$是x的p.d.f.。在证明中，可以利用下列事实：如果积分被限制在$|x-\mu|\geq a$ 区间时，上述积分值会变小，而且在该区间内将$(x-\mu)^2$代之以a^2时，积分值会变得更小。

习题(续一)

(b) 利用Chebyshev不等式证明大数弱定理，即对于任何给定

$\varepsilon > 0$ 有 $\lim_{n\to\infty} P\left(\left|\frac{1}{n}\sum_{i=1}^{n} x_i - \mu\right| \geq \varepsilon\right) = 0$

这等价于只要x 的方差存在的，$\bar{x}$ 是一个符合 μ 期待值估计量的结论。

习题7.2:考虑一随机变量 x 具有期待值 μ 和方差 σ^2,对应于一个样本的测量 $x_1,\ldots,x_n$.

(a) 假设利用样本平均值$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ 已经估计出平均值μ。证明

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{n}{n-1}(\overline{x^2} - \bar{x}^2)$$

是方差σ^2是一个无偏估计量。(提示,利用当i≠j时 $E[x_i x_j]=\mu^2$和对所有的 i ， $E[x_i^2]=\mu^2+\sigma^2$)

习题(续二)

(b) 假设平均值μ已知，证明

$$S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 = \overline{x^2} - \mu^2$$

是σ^2的一个无偏估计量。

习题7.3: (a) 证明上题中 s^2 的方差是

$$V[s^2] = E[s^4] - (E[s^2])^2 = \frac{1}{n}(\mu_4 - \frac{n-3}{n-1}\mu_2^2)$$

这里 $\mu_k = E[(x-\mu)^k]$ 是x的第k阶中心矩。为此，可以首先证明 s^2 可写为

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n} x_i^2 - \frac{1}{n(n-1)}\sum_{i,j=1}^{n} x_i x_j$$

然后,证明 s^4的期待值为

习题(续三)

$$E[s^4]=\frac{1}{(n-1)^2}\sum_{i,j=1}^{n}E[x_i^2x_j^2]-\frac{2}{n(n-1)^2}\sum_{i,j,k=1}^{n}E[x_ix_jx_k^2]$$
$$+\frac{1}{n^2(n-1)^2}\sum_{i,j,k,l=1}^{n}E[x_ix_jx_kx_l]$$

计算在每个求和中有多少项给出代数矩μ'_4或$\mu_2{}'^2$。注意其它项均包含至少μ的一次幂。将包含μ的项设为零，把结果以中心矩μ_2和μ_4表示出来。扣除上题中给出的$(E[s^2])^2$，就可以得到s^2方差的最后结果。

(b) 找出 x 服从高斯分布情况的 s^2 方差(可以利用高斯分布的第4阶中心的矩为 $\mu_4=3\sigma^4$这一关系)。